

Evaluating Voxel Drawing for Virtual Reality Environments

by

Johna Latouf

Submitted in partial fulfilment of the requirements  
for the degree of Bachelor of Computer Science

at

Dalhousie University

Halifax, Nova Scotia

July 2018

© Copyright by Johna Latouf, 2018

# TABLE OF CONTENTS

TABLE OF CONTENTS.....	III
LIST OF TABLES .....	VI
LIST OF FIGURES.....	VII
ABSTRACT.....	IX
LIST OF ABBREVIATIONS USED.....	X
ACKNOWLEDGEMENTS.....	XI
CHAPTER 1 INTRODUCTION .....	1
CHAPTER 2 BACKGROUND AND RELATED WORK.....	3
2.1 VR DEVICES.....	3
2.2 ACCURACY AND EASE OF USE.....	3
2.3 PHYSICAL COMFORT.....	5
2.4 3D MODELING TECHNIQUES .....	6
CHAPTER 3 SYSTEM DESIGN .....	8
3.1 THE OCULUS RIFT AND LEAP MOTION .....	8
3.2 THE VR SYSTEM .....	9
3.2.1 <i>The VR Drawing Environment</i> .....	9
3.2.2 <i>The VR User Interface</i> .....	10
3.2.3 <i>The VR Drawing Tools</i> .....	11
3.2.4 <i>Saving, Loading, and Exporting</i> .....	13
3.3 THE DESKTOP-KEYBOARD SYSTEM.....	14
3.3.1 <i>The Desktop-Keyboard Drawing Environment</i> .....	14
3.3.2 <i>The Keyboard Drawing Tools</i> .....	15
CHAPTER 4 RESEARCH METHODOLOGY .....	16
4.1 GENERAL STUDY OVERVIEW .....	16
4.2 RESEARCH QUESTIONS .....	16

4.3 DESKTOP/KEYBOARD ACTIVITY .....	17
4.4 VR ACTIVITY .....	18
4.4.1 <i>Tutorial</i> .....	18
4.4.2 <i>Model Recreation</i> .....	20
4.5 ANALYTICAL SETUP .....	20
4.5.1 <i>3D Models</i> .....	20
4.5.2 <i>Recordings and Logs</i> .....	21
4.5.3 <i>Questionnaires</i> .....	22
CHAPTER 5 RESULTS AND ANALYSIS .....	24
5.1 TIMING ANALYSIS .....	24
5.2 DISCUSSION OF THE DESKTOP/KEYBOARD TASK .....	24
5.3 DISCUSSION OF THE VR TASK .....	26
5.3.1 <i>Tutorial</i> .....	26
5.3.2 <i>Platform and Drawing Environment</i> .....	27
5.3.3 <i>Comfort</i> .....	28
5.3.4 <i>Pinch Drawing</i> .....	28
5.3.5 <i>User Interface</i> .....	29
5.3.6 <i>Drawing Techniques</i> .....	30
5.4 MODEL ANALYSIS .....	31
5.5 QUESTIONNAIRE .....	34
5.5.1 <i>Version Comparison</i> .....	34
5.5.2 <i>VR Ratings</i> .....	36
5.5.3 <i>Comments and Feature Requests</i> .....	37
CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS .....	38
6.1 GENERAL CONCLUSIONS .....	38
6.2 INTERACTION IMPROVEMENTS .....	38
6.3 TUTORIAL IMPROVEMENTS .....	40
6.4 LIMITATIONS .....	40
REFERENCES .....	42

APPENDIX A.....	46
DRAWING TIMES.....	46
APPENDIX B.....	47
QUESTIONNAIRE QUESTIONS.....	47

## LIST OF TABLES

Table 1 Questionnaire statements used to compare versions.....	23
Table 2 Questions for assessing the VR application independently .....	23
Table 3 T-Test results for the comparison of the VR and Desktop application.....	35
Table 4 Participant times for VR Tutorial, VR Drawing, and Desktop Drawing rounded to nearest second .....	46

## LIST OF FIGURES

Figure 1 The Desktop/Keyboard and VR/Leap Motion applications.....	8
Figure 2 The platform and hands (left), different coloured voxels drawn using various brushes in VR Voxels (centre), the voxel mesh in Unity 3D [4] with one chunk selected(right).....	10
Figure 3 The hand-attachment menu and HUD in the VR application .....	11
Figure 4 The desktop application.....	14
Figure 5 The three Desktop/Keyboard reference model steps.....	18
Figure 6 VR Tutorial screenshots.....	19
Figure 7 The reference model and drawing platform.....	19
Figure 8 Slight changes to references models: (A) one addition 516 dissimilarity, (B) two additions 327, (C) one deleted voxel 216 dissimilarity, (D) one deleted voxel 67, (E)two random voxels (purple) against the desktop model 44,158, (F) against VR model 43,670 .....	21
Figure 9 Individual time spent with the desktop application and in the drawing scene of the VR application.....	24
Figure 10 Users who created more accurate VR models and spent more time on Desktop drawing than the VR drawing (left) preferred VR and had trouble with keyboard controls (right). See Appendix A for all times .....	26
Figure 11 A successful grasp to move the drawing platform (left) versus attempting to move the platform with both hands (right) .....	27
Figure 12 Common drawing techniques: using the brushes to approximate shapes (left), drawing up from the bottom (middle), overlaying the reference model on the drawing grid (right).....	31
Figure 13 Correctly placed voxels in both versions of the application .....	32
Figure 14 Participant P10's drawing method demonstrates limitations with counting matching full voxels .....	32
Figure 15 Average and Individual Difference scores.....	33

Figure 16 Models (purple) with the lowest and highest dissimilarity from the reference model (blue) with scores 2447 and 10392 .....	34
<i>Figure 17 Model (purple) with the lowest and highest dissimilarity from the reference model (blue) scoring 3743 and 8379 .....</i>	<i>34</i>
Figure 18 Comparison of applications from questionnaire data, ranking features from 1 (strongly disagree) to 5 (strongly agree) .....	35
Figure 19 Ease of Use by participant, users who found the keyboard controls difficult to use (below 3/neutral) always found the VR version easier to work with.....	36
Figure 20 The average ratings for the VR application, ranking feature quality from 1 (strongly disagree) to 5 (strongly agree).....	36



## **ABSTRACT**

Virtual Reality devices are entering the mainstream video game and software markets, offering new tools and challenges for developers. This thesis describes the development and evaluation of a 3D modeling system for the Virtual Reality environment. It presents the results of a user study conducted with twenty participants and analyzes their experiences and reactions. It concludes that users find the VR application lacking in both accuracy and interface design when compared to a desktop version of the application, but overall users still prefer the VR experience. This report recommends a series of improvements based on those results and highlights feature requests made by participants.

## **LIST OF ABBREVIATIONS USED**

VR	Virtual Reality
AR	Augmented Reality
HUD	Heads Up Display
UI	User Interface
3D	Three Dimensional
OBJ	3D Model File Format

## **ACKNOWLEDGEMENTS**

I would like to thank my supervisor Dr. Stephen Brooks, PhD student Mohammad Hossein Salimian, and the Faculty of Computer Science of Dalhousie University for providing me with all of the direction, advice, and tools to develop this thesis. I would also like to express my gratitude to my partner and family for their support and encouragement throughout my education.

## CHAPTER 1 INTRODUCTION

Recent advances in Virtual Reality (VR) and Augmented Reality (AR) devices offer novel, immersive experiences for a wide range of games, educational programs, and design applications. Today, professional artists often work with standard 2D desktop or tablet displays that limit their range of movement and require depth cues and multiple orthogonal views to supplement the lack of 3D visualization. VR's stereoscopic display system provides users with a true 3D experience. VR has been described as “an emerging and independent art category”, which will transform the way humans create art [1]. Controller and body tracking also allows artists a full range of physical motion, taking the design experience from the confines of a mouse or stylus to a level of freedom one might find with a traditional canvas or physical clay. In particular, the Leap Motion tracking system requires no hand-held devices and models a set of 3D hands within the VR application, which reproduce the user's movements and interact with objects in the 3D scene. This frees artists to use their hands in a more natural way, without learning to utilize controller buttons.

With these advancements comes new design questions and challenges. Mid-air interaction controllers are often responsible for arm fatigue [2], and VR applications can cause physical discomforts such as dizziness and motion sickness [3]. Achieving a desired degree of accuracy required for 3D modeling with hand tracking systems such as the Leap Motion is also difficult, especially for inexperienced users. An ideal tool for artistic expression should be unobtrusive and physically comfortable so that artists are free to focus on design and not control schemes or sore shoulders. This study was designed to evaluate ways to improve accuracy, develop possible solutions to arm fatigue and discomfort and assess user interface (UI) and other interactive elements in a VR drawing application.

To address these challenges, this study explores the development, testing, and analysis of a 3D voxel modeling application for the Oculus Rift VR headset and the Leap Motion controller system. Voxels, a portmanteau of “volumetric pixels”, are uniformly sized, individual cubes used to build 3D shapes. The system provides an adjustable 3D drawing platform and tools for editing voxel models. A second version of the system was developed for a traditional desktop display to aid in the comparative analysis of the VR application’s accuracy and ease of use. Each version was created using the Unity 3D [4] game engine. A set of twenty participants tested both the VR and desktop versions of the application for accuracy, ease of use, comfort, and engagement. This report will outline and analyze the results of the user study before providing recommendations for improvements and future development.

## CHAPTER 2 BACKGROUND AND RELATED WORK

### 2.1 VR Devices

This study combines the Oculus Rift virtual reality viewer with the Leap Motion hands-free controller, a third-party gesture-based system. The Oculus Rift [5] headset projects a stereoscopic display of 3D environments, tracking the user's head movement to provide a fully immersive experience. The Leap Motion [6] is a set of cameras and infrared LED's used to track arm and hand movement. The Leap Motion software uses this tracking data to build a set of 3D arms and hands for interaction in desktop and VR applications [7].

### 2.2 Accuracy and Ease of Use

Previous studies have assessed the accuracy, efficiency, and usefulness of gestures for 3D mid-air user interface designs for both VR and flat screen displays. The Oculus's stereoscopic design is likely to improve the user's ability to quickly and accurately reach for objects in a virtual setting over a monoscopic view, particularly when pointing to objects along the Z-axis, as is shown by Boritz and Booth's study [8]. Aigner et al. categorize in-air gestures as either selection, release, acceptance, refusal, removal, cancel, navigate, translate, rotate, and identify. While selection and release gestures were most likely to include grasping and the release of grasping, they found that users attempting to move objects are more likely to engage in pointing and "iconic" gestures, rather than direct manipulation of objects. Their participants showed a strong preference for pointing to navigate and identify objects. They also make note of cultural differences in gesture communication, which may be a significant limitation in the study of gesture interaction [9].

In 2016, Cui et al. developed a simple browser-based 3D modeling program using the Leap Motion controller. In their design, clay-like objects can be scaled, twisted, and tapered with hand gestures. Objects have three handles for rotation which are

manipulated using pinch gestures. Participants in their study found that modeling with the Leap Motion “feels pretty natural”, but some did not care for the floating 3D hand avatars used. And participants found the Leap Motion controls did not always provide a desirable level of accuracy. This could be attributed to the Leap Motion’s tracking capabilities [10]. The Leap Motion’s SKD has undergone several updates since this study was published, but achieving the level of detailed accuracy required for design applications remains a challenge. In 2017 they improved upon this study by comparing their system against traditional modeling software and found that modeling was faster using the Leap Motion controller than the mouse and users reported a preference for working with a 3D, mid-air modeling control scheme [11].

Shen et al. developed a volumetric data visualization system combining the Leap Motion with large screen displays. They found that sticking objects to controller hands was more efficient than gestures that mimic mouse usage and free-hand gestures were more effective than hand-held tools at manipulating objects. While their design was efficient compared to traditional control systems, it only improved accuracy by a small amount. [12]. Seif et. al designed and tested a similar medical training system that enables a user to manipulate and inspect 3D human anatomy objects [13]. The 3D modeling system developed by Vinayak and Ramani dispenses with the use of gestures all-together in favor of a more realistic, clay-like modeling experience. Though they found this method was limited by the Leap Motion’s tracking abilities and that users require more encouragement to edit the back faces of the models [14].

Mid-air, Leap Motion controls combined with desktop displays are also implemented in virtual book applications [15]. Viyanon and Sasananan have developed a VR interior decorating system using the Leap Motion. This system includes selecting items from menus and placing them in a 3D room using Leap Motion gestures. Their initial user study indicates that participants were able to learn the interface well, but some experienced dizziness or discomfort [16].

Modeling applications may benefit from haptic feedback or physical tools. In Cutting, Deforming and Painting of 3D meshes in a Two Handed Viso-haptic VR system, Faeth et. al developed and tested a VR modeling system that provides the user with physical sensory feedback using a stylus device [17]. Schkolne et al. tested a VR drawing application using a Responsive Workbench and hand tracking system capable of providing haptic feedback. Artists found the application promising as a means of improving creativity. The inclusion of tangible drawing tools improved the experience [18]. Arora et. al designed an application for the Microsoft Hololens that combines 3D modeling in AR with 2D sketching using a stylus. They employed a smoothing method to improve overall drawing quality and the physical sketching activity limits fatigue [19].

### **2.3 Physical Comfort**

Arm fatigue caused by gesture interaction with 3D interfaces, often called “gorilla arm syndrome”, is a growing concern among devices such as the Leap Motion. Guinness et al. evaluate solutions to this problem that include anchoring elbows on table-tops and modeling the interface to the user’s comfort space [20]. Nunnari et al. have experimented with encouraging users to switch postures regularly during interactions to alleviate fatigue, but indicate that further study is required to produce useful guidelines for designers [21].

The use of VR devices has also lead to reports of a feeling similar to motion sickness, often call “VR sickness” or “cyber sickness”. Davis et al. describe the most common symptoms of “cyber sickness” as “nausea, disorientation, headaches, sweating and eye strain” [22]. These symptoms may be severe enough to discourage the use of virtual reality devices. They define cyber sickness as a subset of motion sickness in which users experience a mismatch between perceived movement and body movement. There is not yet a concrete medical explanation for cyber sickness, although there are several theories. The most accepted theory, as described by Davis et al, suggests that incompatible movement and orientation information causes conflicts between sensory systems in the human body. They indicate that the leading factors in whether or not users will



experience cyber sickness include the age and gender of the user, the quality of the VR projection (lag and flickering screens are more likely to cause sickness), the ergonomic quality of the experience, and the amount of control the user has over movement within the VR environment [22].

## **2.4 3D Modeling Techniques**

This study also involves a degree of 3D modelling so it is worth briefly reviewing related methods of model construction. Common software packages for polygon modeling include Blender [23], Maya [24], and 3D Studio Max [25] and technical design applications such as AutoCAD [26] and Google Sketch-Up [27]. Popular Voxel modeling software includes Qubical [28] and MagicaVoxel [29]. Modeling 3D surfaces and environments on 2D displays presents several challenges. Techniques include direct drawing and editing of surfaces, scanning real objects and editing those, and sketch-based modeling.

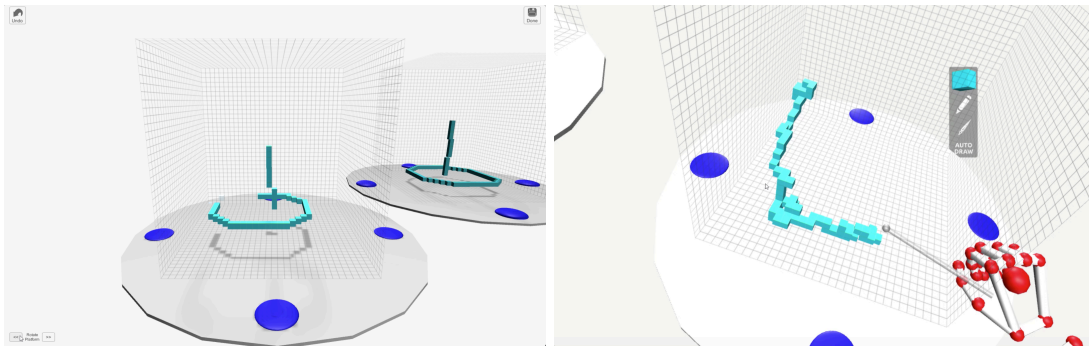
Steinicke et. al [30] propose using Interscopic Interaction Strategies, rather than VR or AR viewing devices. Their design creates a stereoscopic view from a desktop display, while maintaining a familiar 2D UI. They include a cursor that aligns with 3D surfaces in the model to allow for 3D interaction with a traditional mouse. A number of non-traditional modeling applications have been designed to improve accuracy and provide a more natural experience for artists and designers. Oe et. al [31] created an application that incorporates scanning real world objects using a four-sided scanning device and converts them to voxels. The results for this system are mixed. Some scanned shapes are not replicated well, but with improvements the developers hope to make a 3D modeling tool that allows for rapid prototyping. Grossman et. al [32] developed a system for scanning 2D designs for conversion to 3D models. Designers can input large scale sketches to be extruded into polygon surfaces using a hand controller tracking system. This application extends traditional modeling software's ability to display 2D and 3D versions of the same object while allowing designers to work with traditional hand-drawing techniques.

Techniques beyond direct drawing and scanning of surfaces are also available. Implicit modelling, a method for generating surfaces using control points, may also provide a sense of realism for interactive modeling due to their ability to “blend” together [33]. Procedural modeling combined with interactive modeling may offer solutions to problems with mid-air controllers. Igarashi and Hughes [34] tackled accuracy problems associated with touch screen gestures by developing a suggestive interface that provides a selection of edits to a 3D image based on lines drawn by the user. This system lets users preview changes and choose the desired output, or ignore them and try again. This approach works well for simple scenes, but may not scale well for very complex 3D designs.

## CHAPTER 3 SYSTEM DESIGN

The original VR Voxels system was developed for a directed study on VR and AR Systems in the summer of 2017. This system included drawing, painting, and selection tools. The application was developed using Unity 3D and C#. Several features have been updated, added, or removed to facilitate the user study. In addition, a version of the application using a desktop flat screen display and keyboard controls was also developed for use in the study.

This system presents the user with a 3D voxel drawing canvas that sits on a movable platform. When the application begins, the drawing canvas contains only empty voxels. Using pinch gestures and brushes, the user converts these empty voxels closest to the Leap Motion hands to full voxels to build and modify a 3D model. In the desktop/keyboard version of the application, the Leap Motion hands are replaced with a cursor, operated using a standard keyboard and mouse.



*Figure 1 The Desktop/Keyboard and VR/Leap Motion applications*

### 3.1 The Oculus Rift and Leap Motion

The Leap Motion Orion Beta SDK for Unity 3D, along with the Leap Motion Interaction Engine (1.1.1) [35] provide the hand objects, gesture recognition, and physical interaction functionality. For this application, pinch, grasp, and flat palm gestures are measured, as well as palm direction. Leap Motion's InteractionButton script is used to create UI

elements. When any part of a Leap Motion hand contacts an Interaction Button, the button switches to a “pressed” state. UI elements that attach to the Leap Motion hand use the Leap Motion’s built-in attachment and animation behaviours.

## **3.2 The VR System**

### **3.2.1 The VR Drawing Environment**

To create the voxel drawing canvas, VR Voxels expanded the free voxel terrain system designed by Alexandros Starvrinou [36]. Voxel units are stored in “chunks”, shown in figure 2 (right). Each chunk contains 16 x 16 equally sized units which represent a voxel. There are two types of Voxel objects: Empty and Full. For each full voxel present in a chunk, a cube mesh is drawn using triangle polygons. The AlexStv design implements a marching cube optimization to prevent redundant mesh triangles. The chunks are limited to 16 x 16 voxels to accommodate Unity 3D’s mesh triangle limitations. Each face of a voxel is assigned an integer value. A texture sheet is tiled on the full voxel model. Each tile on the texture sheet is chosen by an integer coordinate that can be used to “paint” voxels. When the paint tool is used, the texture coordinate on selected full voxels is updated. This changes the colour of the voxel.

The drawing application allows users to initialize a voxel canvas parent object that can combine anywhere from 2 to 125 of these chunks, but for the purpose of this experiment, the user is limited to a canvas size of 2 x 2 x 2 chunks, or 32,768 voxels. The canvas borders are indicated by a set of walls that display a grid (see Figure 1) texture, indicating individual voxel borders. Containing the voxel drawing within this canvas object allows the user to transform the entire drawing and to perform gestures outside of the canvas to avoid errors.

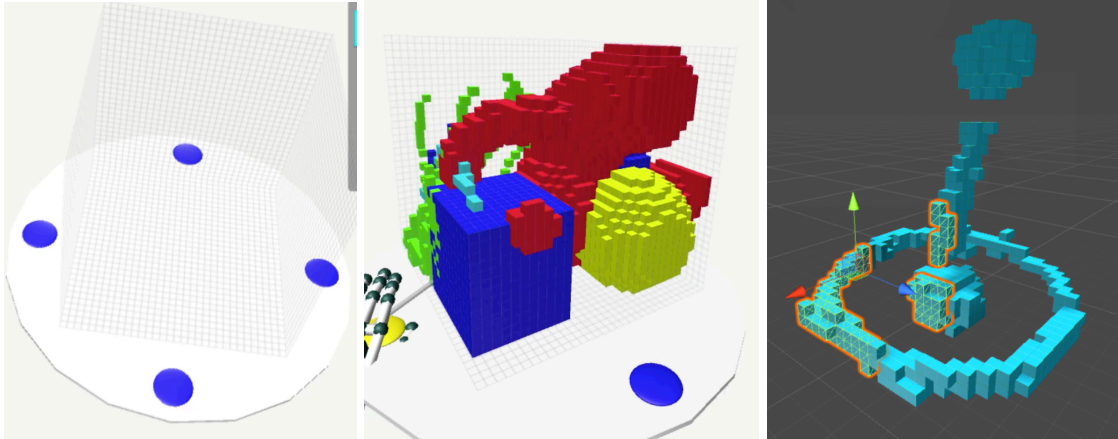


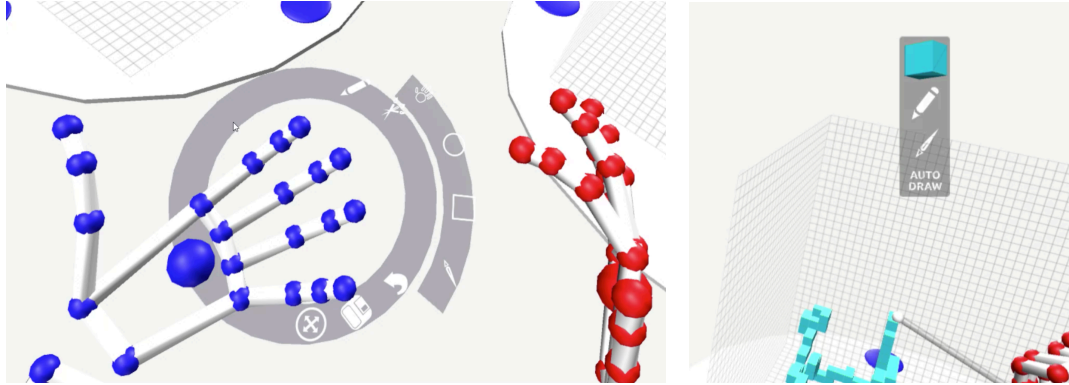
Figure 2 The platform and hands (left), different coloured voxels drawn using various brushes in VR Voxels (centre), the voxel mesh in Unity 3D [4] with one chunk selected(right)

The canvas is placed on a disk-shaped platform, resembling a potter’s wheel. The platform includes four flat, round buttons (shown in figure 2, left) programmed to detect the position and gestures of nearby Leap Motion controller hands. While a hand controller in a flattened position comes in contact with a platform button, the vector between the center of the platform and that button points to the interacting hand. This allows the user to rotate the platform as desired along the Y (vertical) axis. Lifting the hand away from the button or switching into a grasping gesture stops the rotation.

Similarly, the platform can be translated along any axis when a leap motion hand intersects with any platform button and the user performs a pinch or grasp gesture. In this way, the user can “pull” the platform around in the scene. The platform’s position and rotation can be reset to the default values using a button in the left-hand menu. This functionality allows the user to view and interact with the voxel canvas from all angles without needing to traverse the 3D scene, minimizing the chances of sensory mismatch.

### 3.2.2 The VR User Interface

The first iteration of the VR Voxels application included a basic user interface, built with Unity 3D’s default UI tools. With the latest Leap Motion SDK release, elements of this design were deprecated and in the current iteration they have been replaced with a design based on Leap Motion’s new interaction system.



*Figure 3 The hand-attachment menu and HUD in the VR application*

Upon turning the left palm toward the camera, a circular button menu reveals the drawing and file operation tools. Pressing these buttons (which are based on Leap Motion's InteractionButton object) with the right hand opens submenus. The menu is built using Leap Motion's Palm UI Pivot Anchor and Palm UI Pivot Animation objects shown in Figure 3. Brushes are 3D objects that can be used to draw sphere or cube-shaped groups of voxels. These brushes attached to the Leap Motion hands are hidden when this menu is open, to prevent interference. A Head's Up Display (HUD) indicates the drawing state, the brush, and drawing colour displayed in Figure 3. The HUD is parented to the viewport, but placed behind the drawing elements. This way, the HUD is visible at all times without interfering with or occluding drawing elements or appearing blurry. The HUD updates when the menu is closed. Icons are used to indicate brush type and drawing state. A single cube displays the current color.

### 3.2.3 The VR Drawing Tools

In this application, the drawing tools allow users to add, delete, and edit voxels in the drawing canvas. This includes the pinch detection system, the drawing and painting mode options, and brush options. The drawing tools take advantage of the Leap Motion's built-in pinch gesture recognition. Each hand object has pinch detector; an invisible child object located approximately where the index and thumb meet. The Leap Motion can determine when a pinch has started, when it ends, and returns a pinch strength between 0 and 1, which indicates an open hand and closed pinch, respectively. This application considers a pinch strength above 0.8 to indicate a pinch gesture.

The drawing system is a state machine attached to the VoxelEditor object. A draw state indicates whether the user is adding (setting a voxel to full state), deleting (setting to empty state), painting, or selecting voxels, and whether editing is in “auto draw” (edits happen automatically as long as a pinch gesture is held) or “release draw” (edits do not happen until the pinch gesture is released). For the purposes of this user study which focuses on modelling, the painting and selecting tools are disabled. This state is updated when certain buttons are pressed on the handheld UI.

A secondary state machine tracks brush type. When a user switches brush type using the hand-held menu, the primary drawing state machine will update as well. There are four available brush types:

- Pinch Draw: indicated by a hand icon. This is the default brush setting. The voxel that corresponds to the position of the pinch detector is edited when the pinch gesture is performed. In “auto draw” mode, single voxels will continue to be edited as long as the gesture is held. A small cube indicates the position of the pinch in relation to the drawing canvas for each hand. When a drawing gesture is performed, the position of the detector or brush relative to the Voxel Canvas object is used to indicate the nearest voxel unit, indicated by uniform 3D coordinates.
- Sphere Brush: All voxels within a semi-transparent sphere attached to the Leap Motion hand are edited. In “auto draw”, each hand has a sphere brush. In “release draw” mode, the right hand has a single brush. When both hands pinch the sphere brush will scale according to the distance between both hands and draw a single sphere when the pinches are released.
- Cube Brush: Similar to the sphere brush, all voxels within the semi-transparent cube are edited. The cube brushes rotate in accordance with the voxel drawing platform.
- Pen Brush: The pen brushes operate in the same way as the Pinch Draw brush, but they provide the user with a set of extended pen-like objects. The cursor is

replaced with a pen tip at the end of a cylinder attached to the hand. Editing occurs further away from the hand preventing occlusion.

Whenever a pinch gesture begins, a stroke object is created, and the coordinate and texture values of all voxel changes made before the pinch gesture ends are stored in this object, which is placed on a history stack. When the user selects the Undo button, the changes at the top of the stack are reversed.

The paint tool is similar to the drawing tool, but rather than change the type of a chosen voxel, it updates the texture coordinate assigned to the faces of that voxel, if that voxel is full. A submenu containing selectable squares for each texture in the texture sheet is available in the handheld menu. When a new texture is chosen, it is displayed in the HUD. For the purposes of the user study, the colour/texture selection feature is disabled. A voxel selection tool, also disabled for this study, allows the user to “highlight” voxels using a pinch gesture. The coordinates for each selected voxel are stored in a list. Each time an edit is made, the VoxelEditor first checks if this list is empty. If it is not, edits can only be made to voxels within the list.

### 3.2.4 Saving, Loading, and Exporting

A drawing application should allow users to save creations for future editing and to export finished creations. The Voxel Canvas element’s dimensions and content are serialized in C#. The type and texture information for each voxel in the canvas is stored in a 3-dimensional array that is updated with every edit made by the user. The canvas information is saved using a BinaryFormatter object to a binary file with a unique name. The content of these files is deserialized when the user chooses to load a saved model and the canvas is updated with the saved dimensions and model.

A separate command exports the model mesh information to the Wavefront OBJ file format using Stefan Gordon’s OBJParser library [37]. The Wavefront OBJ format is a commonly recognized text-based file format recognized. For each vertex in the voxel mesh, this library appends the necessary vertex and face information to a text file. For



the purposes of this study, the binary information and OBJ files are saved together to a hard-coded folder using a timestamped naming system.

### 3.3 The Desktop-Keyboard System

#### 3.3.1 The Desktop-Keyboard Drawing Environment

The user study portion of this project required participants of all artistic skill levels to redraw a reference model. Measuring model accuracy across skill levels presents a challenge, because of the difficulty in determining an individual participant's spatial awareness and artistic skill-level. To provide a control test model, a keyboard and desktop version of the VR Voxels application using many of the same basic drawing functions was necessary.

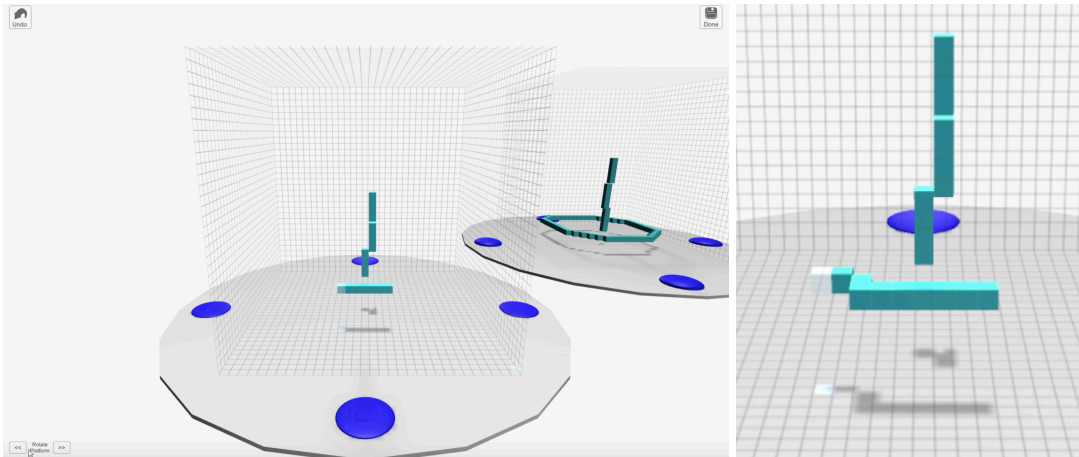


Figure 4 The desktop application

The drawing canvas is similarly enclosed within a set of transparent grid walls placed on a disk-shaped platform. This platform is rotated, at 45 degree intervals, on the Y axis when rotation buttons are clicked with the right mouse button. A set of simple 2D buttons provides undo functionality and a done/save function that saves and exports the model when the user is finished. The reference model is placed on a secondary platform and grid

to the right of the drawing platform. The reference model platform rotates in sync with the drawing platform.

### 3.3.2 The Keyboard Drawing Tools

In the keyboard version of the application, the only drawing options available are Add and Delete. A cursor indicates which voxel will be edited. The cursor is represented by a semi-transparent cube. This cursor is moved, one voxel unit at a time, in the X and Z planes using either the standard arrow keys or the A, S, W, and D keys. Movement in the Y plane is done using the Page Up and Page Down keys, or the mouse scroll wheel. The cursor can move anywhere in the 3D environment; it is not limited to the drawing area. Clicking on existing voxels with the mouse will move the cursor to the clicked point in the scene. The space key switches the selected voxel from Empty to Full or Full to Empty. A spotlight object, aimed at the drawing platform, is parented to the cursor object. This allows the spotlight to move with the cursor. This light produces a white circle that indicates the Z-axis position of the cursor in the 3D grid. The size of the circle grows as the cursor moves up vertically and shrinks as it moves down, allowing the user to better visualize the cursor's position in the scene.

## **CHAPTER 4 RESEARCH METHODOLOGY**

### **4.1 General Study Overview**

This study was designed to determine whether the VR version of this application provides enough accuracy and comfort to draw voxel models, and whether or not users enjoy the VR drawing experience. The user study presented participants with tasks for both the Desktop/Keyboard and VR applications and a follow-up questionnaire. Each version of the application included a reference model, which participants were asked to recreate with the tools available. Half of the participants performed the desktop tasks first, the other half performed the VR tasks first.

Twenty participants were recruited from the students and faculty at Dalhousie University and members of the community. Participants were recruited using the Dalhousie Computer Science mailing list and online communities. Participants ranged in age from 19 to 37. Of the participants who indicated their gender, six were women and thirteen were men. Two of the twenty participants indicated that they were left-handed. Four participants indicated that they were not associated with the Computer Science department at Dalhousie, although two of those worked in related fields. No VR, Leap Motion, or design experience was required to participate in the study. Nine participants indicated that they had tried a VR device in the past and ten indicated that they had used a hand tracking system, such as Leap Motion or Kinect in the past, but no participants had extensive experience with any of these devices. Nine participants used their computer for graphic design or visual art once or twice a month.

### **4.2 Research Questions**

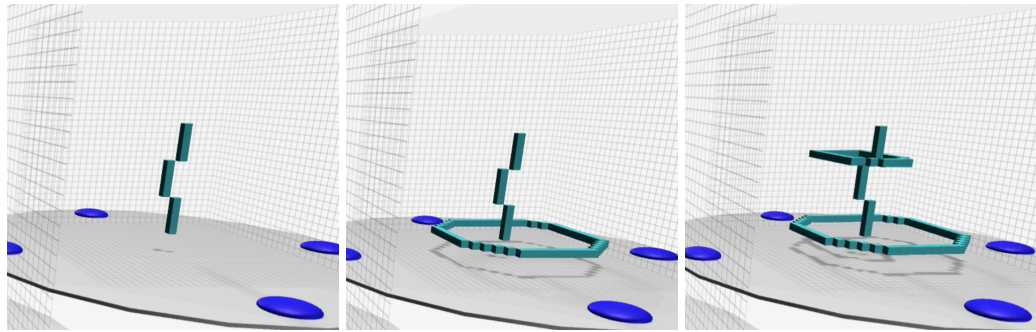
The purpose of this study was to assess the quality of the VR application by answering the following research questions:

- Do users experience arm fatigue or discomfort with a 3D work area that can be moved, and rotated?
- Do the drawing options in this application provide enough precision to draw a voxel model that matches a reference voxel model presented to the user?
- Are the gesture-based user interface and canvas controls easy to understand and use?
- Do users find the application useful for creating 3D models and do they enjoy the overall experience?
- Do users enjoy the experience of the VR application compared to a keyboard/desktop experience?

### **4.3 Desktop/Keyboard Activity**

Because participants were not recruited or grouped based on artistic skill, the desktop/keyboard task list was designed to provide each participant with an opportunity to recreate a voxel model using familiar devices to assess their individual 3D drawing skill-level. This application also allowed participants to make comparisons when answering the questionnaire. Users were presented with the desktop application, a standard keyboard, a brief set of instructions from the investigator, and a “cheat sheet” indicating the keyboard and mouse control scheme. Participants were also encouraged to ask for help if necessary.

Participants were seated at a desk and instructed to recreate a voxel model, presented on a secondary platform in three pieces, shown in figure 5. After the user was satisfied with a piece of the model, he or she was instructed to click a Done/Save button with the mouse, and a second piece was added. The final model was saved and exported as a binary file and OBJ file. After this task, participants answered a set of questionnaire questions related to the keyboard application.



*Figure 5 The three Desktop/Keyboard reference model steps*

## **4.4 VR Activity**

Prior to beginning the VR task, the investigator provided each participant with instructions on how to wear and adjust the Oculus headset, and how the Leap Motion works. Users were given tips for using the Leap Motion effectively and told to ask for help when necessary. Participants were instructed to stand for this task in a room designated for VR use. Users were told there would be no need to walk around with the headset on, but if they did take a step in any direction, the investigator would be present to monitor safety. Participants were also instructed to take breaks if desired.

### **4.4.1 Tutorial**

To help participants learn the basic gestures for the VR application, a short tutorial scene preceded the reference model recreation. In the tutorial, a short text message and “Get Started” button were displayed. After clicking the “Get Started” button, Back and Next buttons, attached to the drawing platform appeared, and a set of text instructions, images, and animations were presented in the HUD.

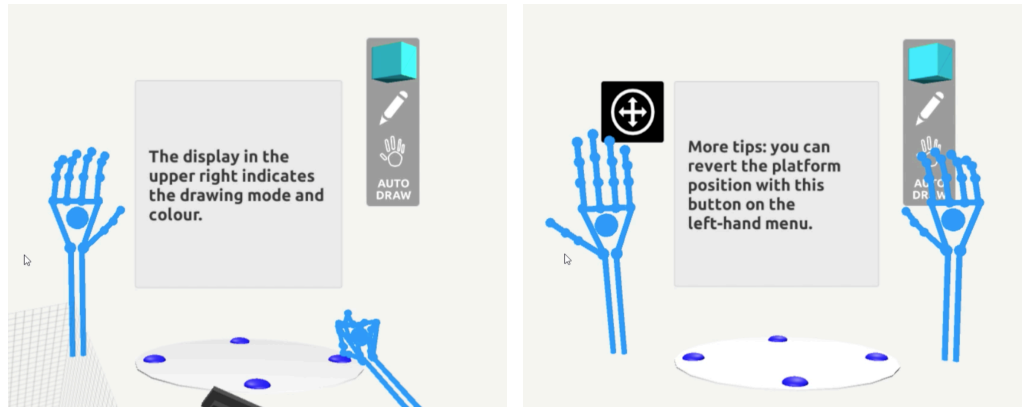


Figure 6 VR Tutorial screenshots

The tutorial outlined the objects in the scene: the drawing platform, HUD, Leap Motion hands, and UI. A set of blue hands, as shown in figure 6, performed the grasping and flat palm gestures used to move and rotate the platform, the gesture used to open the menu, and the pinch gesture used for drawing. The tutorial walked participants through drawing and deleting with the basic pinch brush setting. Participants were told that these instructions were enough to complete the tasks, but that if they needed more information, or wanted to try secondary brushes, they were welcome to ask the investigator. Participants were reminded that the scene would reset after the tutorial, so they would not need to worry about the quality or retention of practice drawings. As soon as the final slide in the tutorial was displayed, a button prompt allowed participants to begin the drawing task.

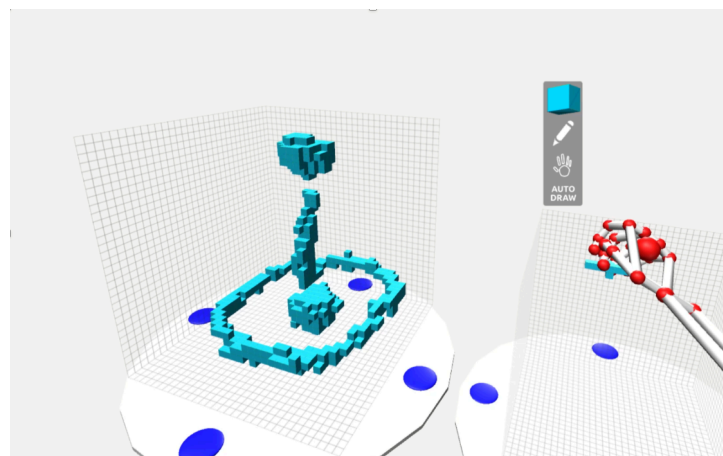


Figure 7 The reference model and drawing platform

#### 4.4.2 Model Recreation

After the tutorial, the drawing platform position was reset and all voxels were deleted. The reference model, shown in figure 7, was placed on another platform beside it. Participants were instructed to recreate the model to the best of their abilities. This reference model was presented in full, rather than a series of steps. The model consisted of four separate pieces: a centre-bottom spherical shape, a bottom circle, and central stroke, and a top spherical shape. When participants finished recreating the model to their satisfaction, they were instructed to save the model using the left-hand menu.

### 4.5 Analytical Setup

#### 4.5.1 3D Models

The nature of voxels allows for a one-to-one exact comparison of the drawing grids for both the reference model and the models drawn by participants. The model information, stored in the binary file saved by each participant, is printed to a multidimensional array of 1's and 0's where 1 indicates a full voxel and 0 indicates an empty voxel. These arrays are then compared. The reference model grid for each application contains a total of 32768 voxels. In both models, the majority of voxels (32203 in VR and 32664 in Desktop) are empty. Because of this, most participants' models show a high percentage of overall accuracy based on this simplistic comparison given that the many empty spaces are in accordance. So, to provide a clearer indication of accuracy, the counts for full and empty voxels are separated. This simple method of analysis also fails to account for minor shifts between the reference and participant models, so a more sophisticated comparison of the reference and use model is needed.

A second test is required to indicate shape similarity. D.Y Chen [38], et al developed an algorithm that determines similarity between 3D models. Their application captures 60 separate orthographic images of each model, each image from a different angle, and determines a difference score between corresponding images. Forty-five difference

values are calculated between each orthographic snap-shop, ranging from 0 to 255, and summed. The ten minimum results from this collection are then summed to create a dissimilarity value.

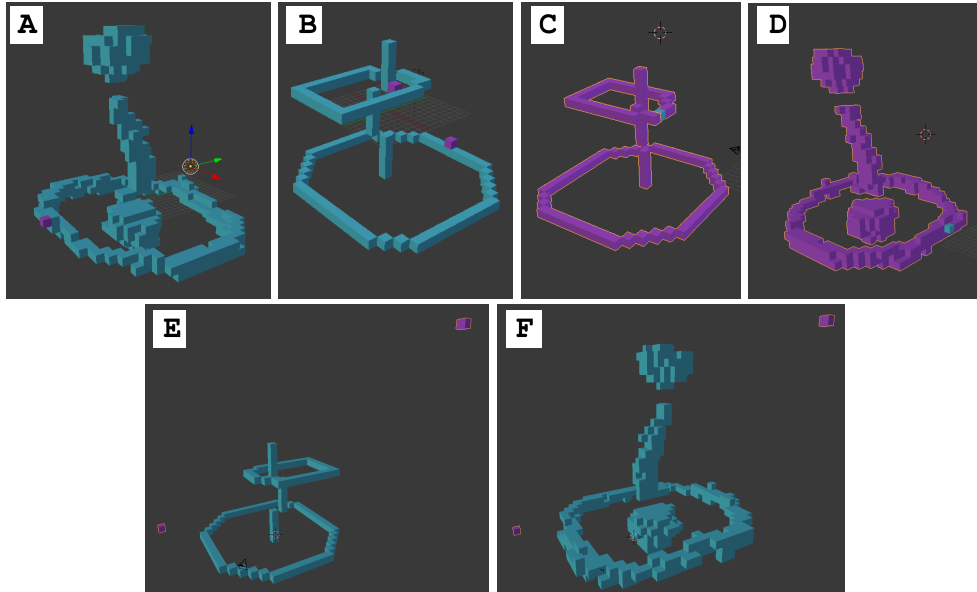


Figure 8 Slight changes to references models: (A) one addition 516 dissimilarity, (B) two additions 327, (C) one deleted voxel 216 dissimilarity, (D) one deleted voxel 67, (E) two random voxels (purple) against the desktop model 44,158, (F) against VR model 43,670

A score of zero indicates that the models are the same. While the highest possible value for this equation would be 114750, it is unlikely a model might reach this level of dissimilarity. In the context of this study, slight differences (removing or adding a single voxel to the reference model) produce dissimilarity scores ranging from 200 to 600. Very different models produce scores between 13000 and 45000. Figure 8 demonstrates minor differences and their dissimilarity scores between 67 and 516, versus large differences between 44,158 and 43,670.

#### 4.5.2 Recordings and Logs

The desktop version of the application recorded mouse clicks, keyboard button presses, and UI button interactions with timestamps and printed them to a log file. The VR application similarly logged drawing and brush state changes, the beginning of the



tutorial and drawing scene, and when models were saved. This information was used to note start and end times and button presses. The Windows Game Bar was used to take screen and audio recordings for both applications. Observed behaviours that were not captured by the screen recording such as participants' body movements were also noted by the investigator.

Events captured by records or notes were organized according separate sets of categories: issues/design flaws/positive aspects/difficulties with the applications, drawing strategies/behaviours, and feature requests. The most common categories were used to make qualitative assessments of the application and recommendations that might improve user experience beyond what was expressed in the questionnaires.

#### 4.5.3 Questionnaires

The questionnaire was broken into three sections: demographic survey, desktop/keyboard related questions, and VR related questions. The keyboard and VR sections included scalar questions ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). These questions were then grouped into categories (shown in tables 1 and 2) both for version comparison and independent ratings of the VR application's features. Table 1 shows the grouping of questions used to analyze the difference between main aspects of the desktop and VR version: enjoyment, ease of use, and accuracy. Table 2 displays the grouping used to further asses the VR application independently.

Table 1 Questionnaire statements used to compare versions

Category	Desktop/Keyboard Questions	VR Questions
Enjoyment	I would use a keyboard application like this one in the future.	I would like use a VR application like this one in the future
Ease of Use	The keyboard application was easy to use. I was able to use the keyboard drawing platform to rotate the drawing comfortably.	Overall, the VR application was easy to use. I was able to use the VR drawing platform to move and rotate the drawing comfortably.
Accuracy	In the keyboard app, I was able to draw/delete voxels where I wanted. In the keyboard application, I was able to draw a model that matched the reference model to my satisfaction.	In the VR app, I was able to draw the voxels where I wanted them to go. In the VR app, I was able to delete the voxels I wished to delete. In the VR application, I was able to accurately recreate the reference model.

Table 2 Questions for assessing the VR application independently

Category	Questions
Enjoyment	I would like use a VR application like this one in the future. I was pleased with the VR drawing I created.
Overall Ease of Use	Overall, the VR application was easy to use. The VR tutorial was easy to follow and remember.
Accuracy	In the VR app, I was able to draw the voxels where I wanted them to go. In the VR app, I was able to delete the voxels I wished to delete. In the VR application, I was able to accurately recreate the reference model.
UI Ease of Use	The icons in the VR tools menu were easy to understand. The VR tools menu was easy to open. In the VR version, it was easy to switch between the draw and delete tools.
Physical Comfort	My arms became tired before the test was over (numbers for this question are reversed) I was physically comfortable using the VR application. I was able to use the VR drawing platform to move and rotate the drawing comfortably.

The questionnaire also included a set of written answer questions to allow participants to note anything they liked or disliked about each application. Participants were also asked to list any changes or features they would like to see applied to the VR version. These comments were categorized according to content for use in qualitative evaluation and design recommendations.

## CHAPTER 5 RESULTS AND ANALYSIS

### 5.1 Timing Analysis

The average time spent drawing in the desktop and VR versions of the application were very close at 13 minutes 12 seconds with a standard deviation of 9:10 and 15 minutes 19 seconds with standard deviation of 11:49, respectively. Figure 9 shows the time distributions. Eleven of the participants had similar times for both applications, while the other nine had larger gaps.

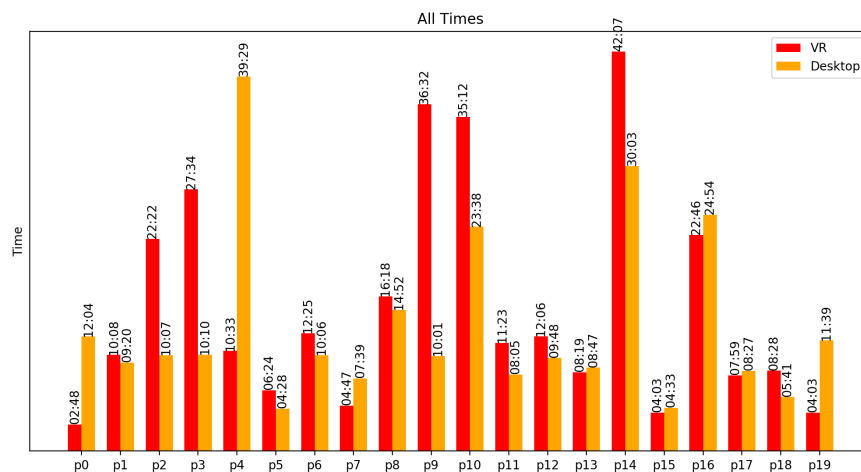


Figure 9 Individual time spent with the desktop application and in the drawing scene of the VR application

### 5.2 Discussion of the Desktop/Keyboard Task

For this task, participants were asked to recreate a reference model using a standard desktop computer setup with keyboard and mouse controls. There were several common questions, feature requests, and observations made during this task. At least six participants began this task by asking if it were possible to view the reference model from the top or if they could move the reference model. Most users found the static view of the reference model insufficient and expressed a desire for multiple view options, similar to

the ones available in common 3D modeling software. Of the three pieces that the user was required to model, users had the most trouble with the third piece. This piece was at the top of the model and users had trouble determining its position and size. Two users recreated the shape well but drew it in the wrong rotation.

Two users expressed difficulty understanding the uniform size aspect of voxels; they asked how to “stretch” or draw a longer voxel to recreate the rectangular shape created by a set of stacked voxels in the image and one suggested adding an outline to voxels. Six participants counted voxels on the reference model out loud. These responses indicate that users had trouble differentiating voxel separation in the desktop application.

The depth cues in this application were also insufficient for most participants. Some suggested replicating the circle created by a spotlight beneath the cursor object on each wall of the drawing grid. Three participants thought they were stacking voxels on top of each other while recreating the first piece, only to find that they were placing voxels behind one another unintentionally.

Users who mentioned during this task that they were familiar with the WASD control scheme (common in many computer games) expressed more comfort with the application. But one user struggled to understand moving the cursor in the Y axis with the mouse wheel and another user recommended replacing mouse controls with key controls.

Five users spent significantly more time (2 minutes or more) on their desktop drawing than their VR drawing (this time does not include the tutorial task) which, combined with observed confusions or frustration, suggests they struggled with the desktop application. Of these five, four participants received a lower dissimilarity score for their VR model than for their desktop model. The dissimilarity score, discussed further in sections 5.2, is a degree of difference from the reference model determined using the algorithm explained in 4.5.1. A lower dissimilarity score indicates the participant’s model matches the reference in terms of shape better than a model with a high score. Figure 10 shows that from this subset of four users, all preferred the VR application more than average and

most indicated a lower than average ease of use with the keyboard controls. One noted in the questionnaire (discussed in more detail in section 5.4) that “Experiencing 3D in 3D felt more natural”.

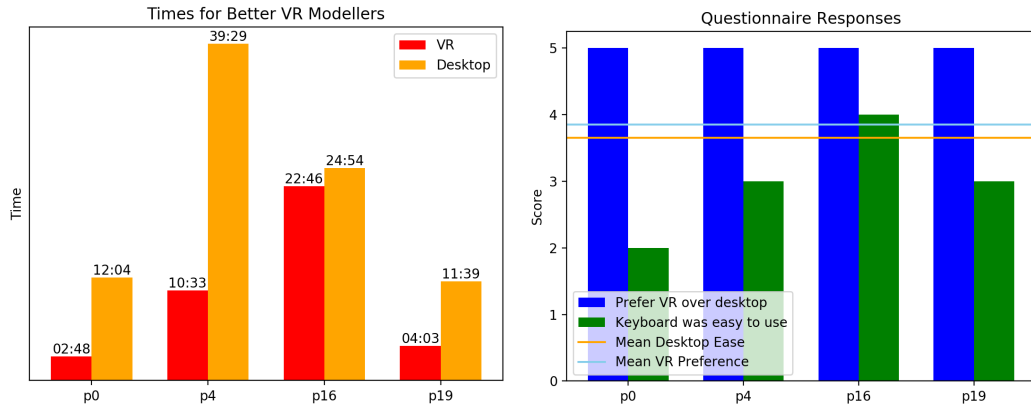


Figure 10 Users who created more accurate VR models and spent more time on Desktop drawing than the VR drawing (left) preferred VR and had trouble with keyboard controls (right). See Appendix A for all times

These four were all female participants. Only six women participated in the study, so more research is necessary to determine if there is a significant difference based on gender, but based on conversation in audio logs, these users may simply have had less experience with common desktop computer game control schemes, which may be a more accurate explanation.

## 5.3 Discussion of the VR Task

### 5.3.1 Tutorial

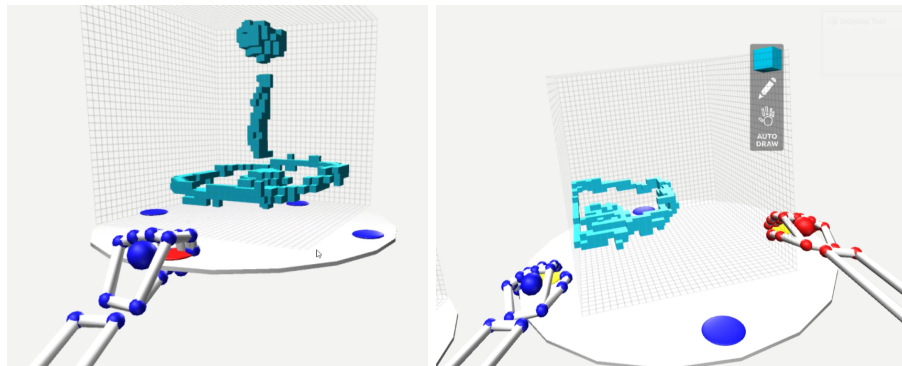
Most participants either got ahead of the tutorial steps or started exploring the system without starting the tutorial. Because of this, the investigator explained aspects of the drawing environment and reminded them of the Next and Back buttons. The Next button in the tutorial screen was often difficult to operate. Users also needed more explanation of the grasp and flat palm controls for moving and rotating the platform than what the

tutorial offered. Overall, the tutorial was insufficient without added verbal instruction from the investigator.

### 5.3.2 Platform and Drawing Environment

Every participant had trouble performing the grasping gesture in the correct spot to move the platform. This indicates that the radius around the control buttons should be wider. The red highlight for these buttons was also over-sensitive and led participants to believe that their grasp was successful when it was not. Half of the participants used both hands to move the platform (shown in figure 11), even though the tutorial animation only showed one hand grasping. In conversation, one participant indicated that this made more sense because a real object of this shape and size would require two hands. One participant also tried to move the platform by pushing it from the centre.

Seven participants had trouble with the rotation gesture. This gesture worked better when it was explained that the centre of the palm needed to come in contact with the control button. In some cases, it was apparent that a participant was attempting to grasp the platform but accidentally activated the rotation. While some users took a step back or forward as necessary, six participants walked around their drawing. This may have been due to personal preference, but these six users also appeared to have difficulty controlling the position of the platform and may have been walking around after giving up in frustration.



*Figure 11 A successful grasp to move the drawing platform (left) versus attempting to move the platform with both hands (right)*

No users indicated having difficulty gauging depth, although some leaned directly into their drawing or the reference model. But at least seven had trouble with occlusion. In particular, it was difficult to tell if voxels were being deleted when pinching was occurring behind existing voxels. Two participants tried to draw the bottom spherical piece of the model by reaching up from beneath the platform. One user asked to see from underneath the platform and others asked if they could tilt the drawing in the X or Z-axis. This indicates that the immersive 3D environment is superior to depth cues in a 2D setting, but occlusion remains an obstacle.

### 5.3.3 Comfort

In terms of comfort, users received help adjusting the Oculus headset and those who spent longer times drawing were asked periodically if they were comfortable. Taller users had to do more reaching at the beginning of the task to move the platform to a comfortable height. But it is noteworthy that no users reported motion sickness or dizziness. Though, some users mentioned that the temperature of the testing room made the experience of wearing a headset less enjoyable, as it was slightly warm. Users who had an easier time with the buttons and controls expressed a high level of enjoyment of the VR experience. One described it as “super awesome”, another said “it feels nicer [than the desktop application]”.

### 5.3.4 Pinch Drawing

Leap Motion detects pinch strength and held pinches. Rather than hold a pinch for each stroke, four users repeatedly tapped pinching or performed a gesture that mimicked sprinkling salt. These gestures were subtle enough to be recorded as pinches by the Leap Motion. One user noted that the pinch was awkward to perform and recommending another gesture.

The majority of users indicated during recording that the pinch gesture detection used for drawing was too sensitive or difficult to control, and the pinch cursors were difficult to

see. They accidentally drew voxels and either erased them or used the UNDO button, especially when entering or exiting the drawing grid with a hand. Users also had trouble remembering to keep their hands in an open gesture when entering the drawing grid or accidentally pinching inside the grid while accessing the menu or moving the platform. As one participant put it, “One problem I’m having is getting used to thinking about my hand gestures”. Two users suggested disabling the non-drawing hand for this reason. But new users to VR and mid-air hand tracking are not accustomed to being consciously aware of hand poses and positions. According to the logs, users clicked the UNDO button an average of 38.4 times. This number includes UNDO use in both the tutorial and drawing scene. Participants agreed that a REDO button would be helpful to avoid accidentally undoing too many strokes. At least three participants also encountered bugs or design flaws with the UNDO functionality, including instances where UNDO appeared to stop deleting strokes after several clicks, and one case in which UNDO appeared to reverse a deletion that had not occurred.

Some participants found that the Leap Motion became unresponsive to pinch gestures which required moving the hand out of view and back. It is unclear why this occurs, although it appears to happen when users look away from a hand that is in the middle of drawing a stroke. Because this task required looking back and forth between the drawing and reference model, this was a common occurrence, and it interrupted the users’ concentration.

### 5.3.5 User Interface

Eight users appeared to struggle either with opening the hand-held menu or keeping it open. The angle that participants held their left hands while accessing the menu varied, with some users holding their palms toward the ceiling, rather than toward their faces. At least four users tended to hold the menu very close to their faces, which may have interfered with performance. Five users expressed difficulty reading the menu icons and most required help reading them in the tutorial scene.



Every participant had at least some trouble clicking the buttons near the top of the menu. Most participants tried to access these buttons by placing the right hand in front of the left, confusing the Leap Motion. Most indicated that the Save button on the side of the menu was not difficult to use, although the investigator usually reminded them to save at the end of the drawing task. This suggests that the menu buttons were too small, too close together, and awkwardly positioned for inexperienced users. Two users recommended placing the UNDO button or other menu items in a separate space that remained accessible without needing to perform a gesture.

Nine users attempted to reach for the Heads Up Display (HUD). One user expressed a dislike for the HUD entirely and found the fact that it “followed” the viewport uncomfortable. Users tended not to notice when accidentally switching between drawing or brush modes, suggesting that the HUD does not do enough to provide information about state and tool changes.

### 5.3.6 Drawing Techniques

Participants tried a variety of drawing strategies and techniques to replicate the reference model, shown in figure 12. At least six participants tried to place the reference model within the drawing grid to assess similarity or trace the model. The participant whose model best matched the reference model in shape used this method. This technique was difficult to perform for users who did not have comfortable control over the platforms, and because all voxels were the same colour, the models were difficult to tell apart.

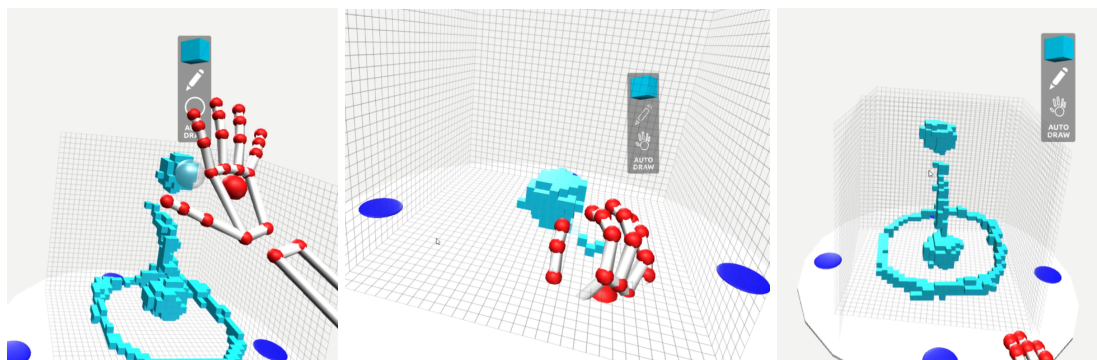


Figure 12 Common drawing techniques: using the brushes to approximate shapes (left), drawing up from the bottom (middle), overlaying the reference model on the drawing grid (right)

Most participants began drawing with the bottom pieces. This may be due to the fact that the models were easier to view from the top. A few participants tried drawing from beneath the platform. Several drew with their hands facing up, and one mentioned that the pinch was hard to see, suggesting that the Leap Motion hands caused some undesired occlusion.

Three participants tried drawing thick strokes using the sphere and cube brushes and then used the delete tool to thin them out. Some users who drew primarily with the standard pinch brush chose to use the sphere or cube brushes for deleting, rather than try to delete one voxel at a time. Six participants tried to use the cube or sphere to replicate the spherical pieces of the model. Participants also varied widely in what degree of similarity they found acceptable, with one participant taking five minutes to create a basic representation of the main shapes, while another spent nearly 18 minutes sculpting the lower spherical piece.

## 5.4 Model Analysis

When models are compared using the simple one-to-one voxel matching method, the overall accuracy of the desktop task was slightly higher than the VR task, with 99% and 96.7% matching respectively. However, it was already noted that this is not an effective

comparison approach with the majority of voxels are empty, and therefore are mostly in agreement in the empty areas.

But when full and empty voxels are viewed separately, the VR application shows a higher percentage of correct full voxels. That is, full voxels added by the user that are placed in the same coordinate as a full voxel in the reference model as a percentage of all full voxels in the reference model. Participants averaged approximately 11.1%, or 65.5 correctly placed full voxels in the VR task. The average of correctly placed full voxels in the desktop version was lower, at 4.1%, just 4.3 correctly placed voxels. This suggests that even when users were able to recreate shapes well, they were not able to discern placement of the reference models on the grid do to location offset issues. This may suggest that better depth indicators are necessary.

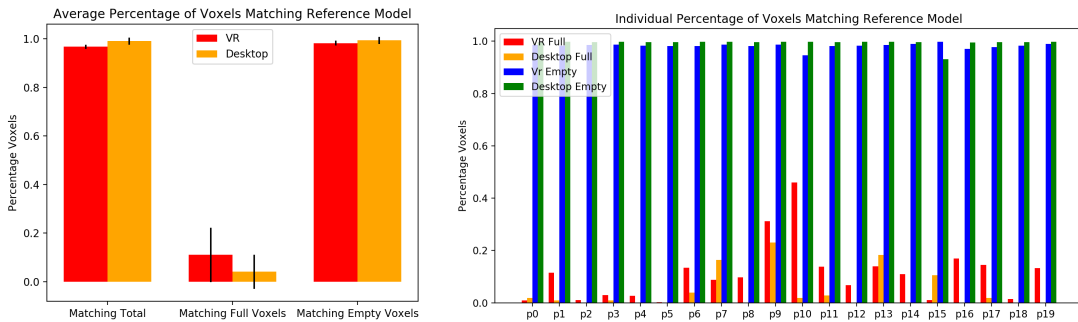


Figure 13 Correctly placed voxels in both versions of the application

Figure 13 indicates the average voxel matching scores and individual scores. Participants who scored low on the VR matching of full voxels also tended to score low on the

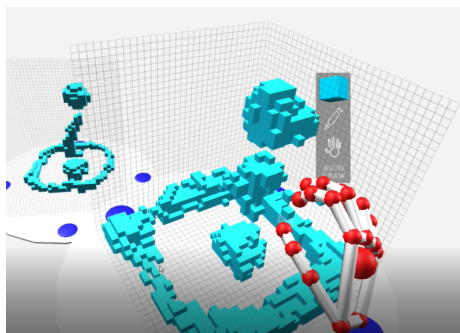


Figure 14 Participant P10's drawing method demonstrates limitations with counting matching full voxels

desktop version. But participants who scored better on the VR version were more varied with the desktop. Only three participants were able to place a higher percentage of voxels correctly in the desktop version than the VR version. For example, participant P10 scored very high on the VR version, but very low on the

desktop version. This is due to P10's method of drawing thick strokes, filling large chunks of voxels. P-values of approximately 0.024 for full voxels, 0.007 for empty voxels, and 0.00000032 for total voxels indicate that there is a statistically significant difference between these results at 95% confidence rate.

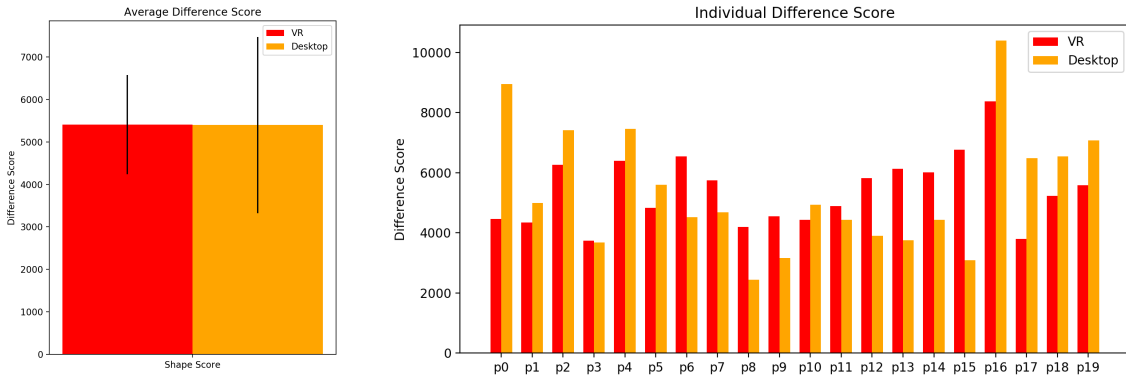


Figure 15 Average and Individual Difference scores

The average dissimilarity scores, shown in figure 15 are 5406.4 for VR and 5398.35 for desktop. Overall, users were able to reproduce the desktop reference model more accurately, but the difference is only very slight. The t-test indicates that these values are not significantly different. Most individual difference scores, as shown in figure 15, suggest that 3D drawing and spatial skills carried over between both versions of the application. The average participant with a better VR model had an 807.6 difference in dissimilarity score, while the average for better desktop models was 815.65, with standard deviations of 1171.09 and 1088.25, respectively. These means are not significantly different. These results suggest that the overall ability for users to draw accurately using the keyboard was not significantly better than their ability to draw in VR, but this may be due to the difficulty some users had with the WASD control scheme or with recognizing the depth cues provided. The dissimilarity score may also be less forgiving to the thinner desktop reference model. Figures 16 and 17 display the lowest and highest scoring models compared with the reference model.

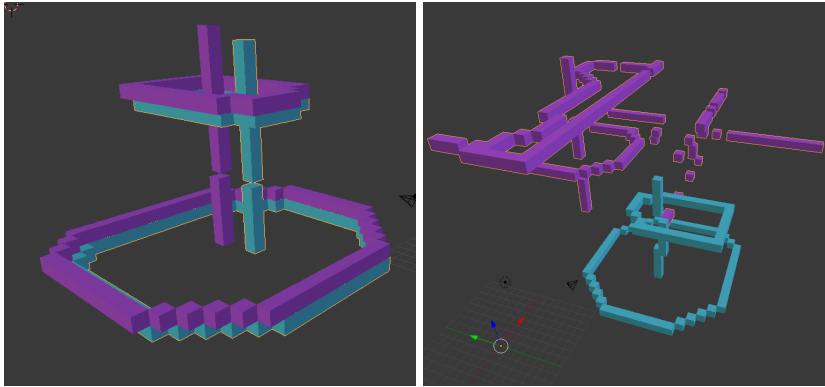


Figure 16 Models (purple) with the lowest and highest dissimilarity from the reference model (blue) with scores 2447 and 10392

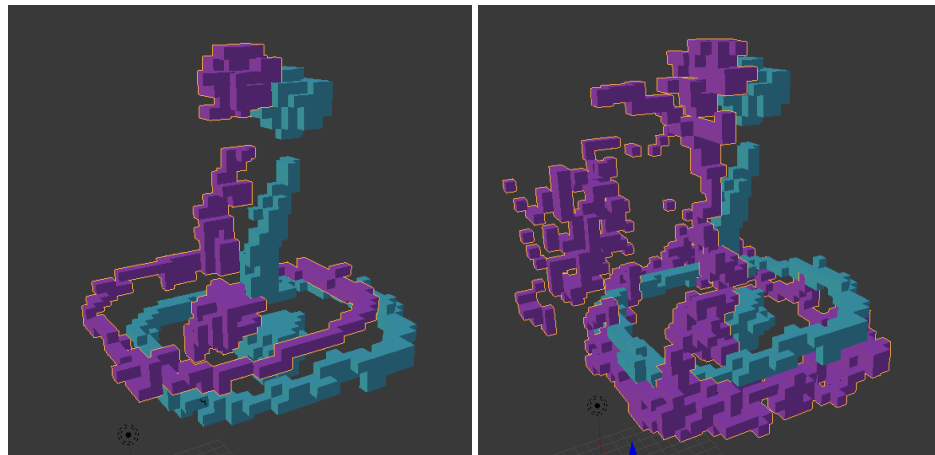


Figure 17 Model (purple) with the lowest and highest dissimilarity from the reference model (blue) scoring 3743 and 8379

## 5.5 Questionnaire

### 5.5.1 Version Comparison

When asked for an overall preferred version, four users expressed a preference for the desktop application, two had no preference, and fourteen preferred the VR application. Shown in figure 18, on average, participants indicated that they found the VR application more enjoyable than the desktop version by more than a full rating point. The ease of use for the VR application was slightly higher than the ease of use for the desktop application. But they rated accuracy for the VR application below the desktop version.

An independent t-test on these responses indicates that the difference in enjoyment ratings are statistically significantly different at 95% confidence, with a p-value of

0.00051. The accuracy rating is also different, with a p-value of 0.011, but the ease of use ratings are not significantly different at 95% confidence, with a p-value of 0.1702. This suggests that users did not find a significant difference in the overall difficulty of the applications, but they enjoyed the VR experience more despite the poorer accuracy.

Table 3 T-Test results for the comparison of the VR and Desktop application

Questionnaire Category	T-Value	P-Value
Enjoyment	3.795	0.0005
Ease of use	1.398	0.170
Accuracy	-2.645	0.013

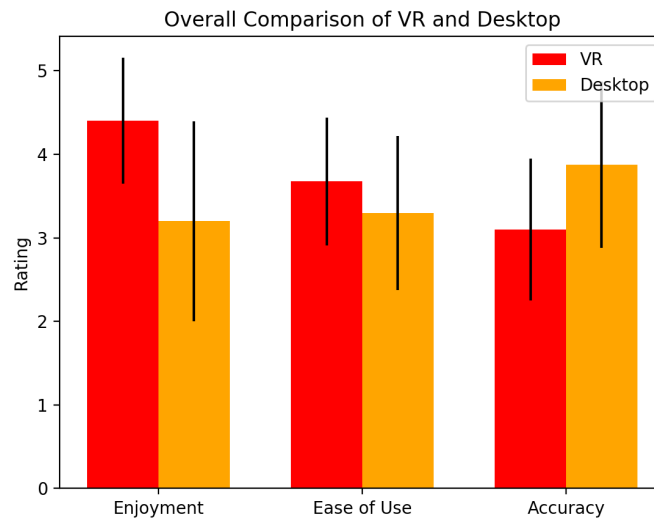


Figure 18 Comparison of applications from questionnaire data, ranking features from 1 (strongly disagree) to 5 (strongly agree)

Of the six users who rated the desktop application’s ease of use below 3, all rated the VR application above 3, while the other 14 users were mixed in their preference.

Interestingly, only one of the four users who appeared to struggle with the desktop version rated the ease of use a 3 or below. Participant P10, shown in the set of six users who preferred the VR version, experienced a crash and had to restart the VR application, but still found it easier to use.

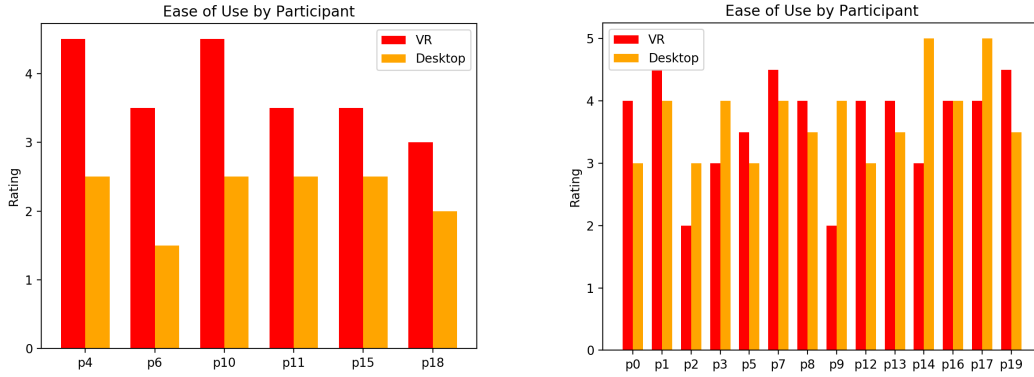


Figure 19 Ease of Use by participant, users who found the keyboard controls difficult to use (below 3/neutral) always found the VR version easier to work with

### 5.5.2 VR Ratings

Figure 20 shows that the average VR ratings indicate that participants find the platform and tutorial easy to work with, even though the UI is more challenging, and the accuracy is poor.

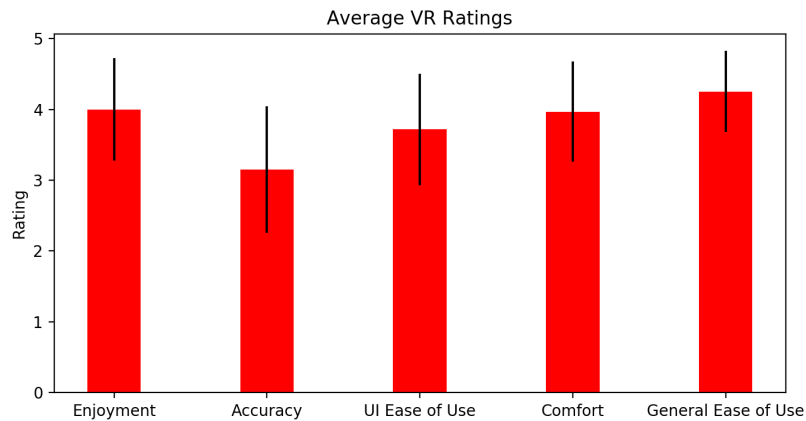


Figure 20 The average ratings for the VR application, ranking feature quality from 1 (strongly disagree) to 5 (strongly agree)

When asked if their arms became tired before the study was over, on a scale from 1 for strongly disagree to 5 for strongly agree, eleven participants answered “strongly disagree”, with mean answer of 1.95. No participants reported discomfort related to sensory mismatch. One user noted that his back was slightly sore afterwards, but he was not sure if this was due to that application being calibrated too low for his height or from standing to perform the drawing task.

### 5.5.3 Comments and Feature Requests

From the questionnaire, eight users indicated that they liked the level of accuracy and control of the desktop application. Two users expressed a preference for the simplicity. Five users noted a dislike for the lack of rotation angles beyond 45 degree intervals. Others disliked the static nature of the platforms and had trouble discerning depth in the 2D environment.

In the VR application, users most liked the ability to move the platforms around, the ability to move around inside a 3D environment, and just generally using VR. One user noted “Being in a computer world was amazing”. Participants disliked the lack of accuracy, difficulty using the menu, and problems moving the platform.

Between the questionnaire and audio logs, there were several popular feature requests. These included a REDO button, bigger menu buttons, a menu detached from the left hand, the ability to select existing voxels and move them, different colours for the reference model, and improved accuracy and controls. Some users asked for the ability to change voxel colours, but this feature was disabled. Two users disliked the fact that both hands could draw and suggested a way to switch to single hand drawing. One user recommended preview voxels, rather than cursors or brushes. One user recommended putting the pen brushes in a position that would be similar to the way users hold a pencil. Some users asked for the ability to differentiate voxels, view a copy of their cursor in the same position in the reference model grid, and the ability to overlay a semi-transparent version the reference model in the same space as the drawing grid.



## **CHAPTER 6      CONCLUSIONS AND RECOMMENDATIONS**

### **6.1 General Conclusions**

This study assessed the accuracy, comfort, and ease of use of a VR voxel drawing system in comparison to a desktop system with twenty participants. Users spent, on average, two more minutes drawing in VR. The models that participants created were, overall, slightly more accurate using the desktop application, with notable exceptions for participants who were unfamiliar with the keyboard control scheme. According to questionnaire responses, participants enjoyed the VR application, but found the accuracy and some controls lacking. If these features were improved, voxel drawing applications like this one might become a popular design tool in the future. This study also yielded many feature requests and observations that would improve VR experience for artists.

It is also worth noting that the desktop version of the application did not necessarily provide an easy test for drawing ability for all users. The difficulty that some participants had with the desktop application was unexpected, but their ability to work comfortably with the VR application was promising. VR and mid-air hand tracking may offer an easier path to digital art and design than traditional applications for users who are not already accustomed to traditional desktop control schemes.

### **6.2 Interaction Improvements**

The hand-held menu was likely the biggest problem with the current design and was challenging for nearly every participant. The buttons were positioned awkwardly and were difficult to press. Flat buttons might be replaced with cubes or spheres that can be grasped, tapped, or swatted. They should also be larger and further apart. Buttons should not be placed around the fingertips, as this encourages users to obscure one hand with the other, which can lead to the Leap Motion only recognizing one hand. The hand-held menu should be accessed infrequently. A better design might only place all of the main

menu items (Draw Mode, Brushes, File, etc.) on the hand menu. When one is selected, a secondary menu could open free from the hand and only close after the user makes a selection, or presses a close button. The UNDO button, along with a REDO button should be available at all times, or these actions could be performed with a voice command.

Accidental drawing was common. The pinch detection was too sensitive and should be adjusted. A delay in activation when a hand enters the drawing grid may also alleviate some mistakes. A highlight or colour change, or improved appearance of the pinch cursor, may also prevent unintended edits. While the pinch detection was too sensitive for most users, the grasp detection for the drawing platform was not sensitive enough. Because some users tried to move the platform with both hands or tried to push the platform, it may be more effective to make an outer circle, rather than a set of small handles, that moves and rotates the platform in a natural, physics-based way. Allowing users to enable or disable this action with a separate gesture or voice command would also prevent accidental movements. The Leap Motion hand models could also be replaced with a custom design that is either semi-transparent or has thinner “bones” to prevent occlusion issues.

Several users asked to be able to select and move existing voxels. Preview indicators, such as semi-transparent voxels for brush edit previews, and outline overlays would make it easier to distinguish and plan drawings. Because several users chose to lean into their drawings, the ability to isolate a small selection and scale it up may make it easier to focus on small portions of a design without needing to crouch and lean. In addition to improving pinch drawing detection, the pinch strength variable should be used to create a pressure sensitive pinch brush that draws thicker strokes as the strength increases.

### **6.3 Tutorial Improvements**

The VR tutorial was ineffective and should also be redesigned. The tutorial buttons were difficult to operate for some users, and many participants ignored the text instructions. Rather than displaying a set of animating hands, a full-body demonstration, either a video or a simple model using motion capture, would convey gestures more thoroughly. The tutorial should also serve as a way to tailor a user profile. As users perform gestures in accordance with the tutorial, the application could save the angles, speed, and pinch strength that are comfortable for that particular user. This information can be used to calibrate the gesture detection for improved accuracy and comfort.

### **6.4 Limitations**

When conducting a user study, one should expect to encounter a phenomenon called “demand characteristics”. Participants in many types of studies have shown a desire to behave in ways that they believe is beneficial to the investigator [39]. In this case, some users may have felt a need to be uncritical of the applications, or to score application features too high on the questionnaires. One participant who struggled with the VR controls even expressed his opinion apologetically, while another participant who struggled with the keyboard controls scored their ease of use quite high. In cases such as these, the investigator can encourage everyone to be honest, because negative feedback is very helpful for future development.

A selection of participants that are associated with the computer science field also serves as a double-edged sword, in this regard. While these students often provide large lists of feature requests and suggestions, they may also be overly forgiving regarding bugs and design flaws because they are familiar with the development process. Ultimately, these biases are difficult to mitigate, so it is important to view all questionnaire and audio feedback in this context.

In that vein, it is also important to consider the novelty of VR technology. Some participants may have been more excited to try out VR in general, which coloured their opinion of the application. Because most participants had little or no experience with VR headsets, they had to rely on the investigator to help adjust their headsets, and may not have known their ideal adjustment settings. No participants indicated that they were colour blind, but most users who wore glasses chose to remove them. One user indicated in conversation that they had a writing disability that could affect spatial skills, but it did not appear to affect their experience.

The disabled and incomplete features in the VR application, changing colours, painting existing voxels, selecting voxels, and filters may have improved user experience or made the application more difficult to use, but were not included in the study due to time constraints. The use of brushes was allowed in the study, but it was not explained in depth unless a participant wished to use them and needed help. Once the application is improved, it may be worthwhile testing these features in the future.

There are also several limitations to the accuracy measurements. While the dissimilarity scores were within the same range, the reference models for the VR and Desktop applications were similar but not identical. Making direct comparisons between a participant's ability to replicate one and not the other can tell us how much their general drawing skills may have affected the accuracy of their VR drawing, but it may also suggest that one shape was simply easier to copy. The keyboard control scheme also proved to be difficult for some users to operate effectively. The direct voxel measurement does not tell us much if the user failed to draw the shape in the correct position. The smaller number of full voxels in the desktop application means that some users placed no full voxels correctly, regardless of shape accuracy.

In retrospect, there are a number of background questions that may have helped improve analysis of usability. Some participants indicated a they were familiar with WASD desktop gaming control schemes, but this should have been included in the questionnaire.

## REFERENCES

- [1] C. Peng and L. Xiaotong. "Analysis of Artistic Language in the Virtual Reality Design". 2017. In *Proceedings of the International Conference on Video and Image Processing (ICVIP 2017)*. ACM, New York, NY, USA, 259-263. DOI: <https://doi.org/10.1145/3177404.3177446>
- [2] J. D. Hincapié-Ramos, X. Guo, and P. Irani. "The consumed endurance workbench: a tool to assess arm fatigue during mid-air interactions". 2014. In *Proceedings of the 2014 companion publication on Designing interactive systems (DIS Companion '14)*. ACM, New York, NY, USA, 109-112. DOI: <https://doi.org/10.1145/2598784.2602795>
- [3] S. V. Mammen, A. Knotte, and S. Edenhofer. "Cyber sick but still having fun". In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology (VRST '16)*. 2016. ACM, New York, NY, USA, 325-326. DOI: <https://doi.org/10.1145/2993369.2996349>
- [4] "Unity". Unity3D.com. 2018. [Online]. Available: <https://www.unity3d.com/>. [Accessed: 07-Jul-2018].
- [5] "Oculus," *Oculus Rift* | *Oculus*, 2018. [Online]. Available: <https://www.oculus.com/>. [Accessed: 07-Jul-2018].
- [6] "Reach into the future of virtual and augmented reality," *Leap Motion*, 2018. [Online]. Available: <https://www.leapmotion.com/>. [Accessed: 07-Jul-2018].
- [7] A. Colgan, "How Does the Leap Motion Controller Work?," *Leap Motion Blog*, 18-Jan-2017. [Online]. Available: <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>. [Accessed: 07-Jul-2018].
- [8] J. Boritz and K. S. Booth. "A study of interactive 3D point location in a computer simulated virtual environment". 1997. In *Proceedings of the ACM symposium on Virtual reality software and technology (VRST '97)*. ACM, New York, NY, USA, 181-187. DOI=<http://dx.doi.org/10.1145/261135.261168>
- [9] R. Aigner, D. Wigdor, H. Benko, M. Haller, D. Lindbauer, A. Ion, S. Zhao, J. Koh, "Understanding Mid-Air Hand Gestures: A Study of Human Preferences in Usage of Gesture Types for HCI," Microsoft Research Technical Report MSR-TR-2012-111, Nov. 2012. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/understanding-mid-air-hand-gestures-a-study-of-human-preferences-in-usage-of-gesture-types-for-hci/> [Accessed on: July 1, 2018].
- [10] J. Cui, A. Kuijper and A. Sourin. "Exploration of Natural Free-Hand Interaction for Shape Modeling Using Leap Motion Controller". *2016 International Conference on Cyberworlds (CW)*, Chongqing, 2016, pp. 41-48. doi:

10.1109/CW.2016.14

- [11] Jian Cui and Alexei Sourin. "Interactive shape modeling using leap motion controller". 2017. In *SIGGRAPH Asia 2017 Technical Briefs (SA '17)*. ACM, New York, NY, USA, Article 3, 4 pages. DOI: <https://doi.org/10.1145/3145749.3149437>
- [12] J. Shen, Y. Luo, X. Wang, Z. Wu and M. Zhou, "GPU-Based Realtime Hand Gesture Interaction and Rendering for Volume Datasets Using Leap Motion," *2014 International Conference on Cyberworlds*, Santander, 2014, pp. 85-92. doi: 10.1109/CW.2014.20
- [13] M. A. Seif, R. Umeda and H. Higa, "An attempt to control a 3D object in medical training system using leap motion," *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Okinawa, 2017, pp. 159-162. doi: 10.1109/ICIIBMS.2017.8279705
- [14] K. R. Vinayak. "A gesture-free geometric approach for mid-air expression of design intent in 3D virtual pottery". 2015. *Computer-Aided Design, Volume 69, 2015, Pages 11-24, ISSN 0010-4485*. doi: <https://doi.org/10.1016/j.cad.2015.06.006>.
- [15] S. Sugiyama, M. Shibata and S. Hiratsuka, "A study of displaying 3D electronic text using augmented reality via Leap Motion," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, 2016, pp. 003015-003020. doi: 10.1109/SMC.2016.7844699
- [16] W. Viyanon and S. Sasananan, "Usability and performance of the leap motion controller and oculus rift for interior decoration," *2018 International Conference on Information and Computer Technologies (ICICT)*, DeKalb, IL, 2018, pp. 47-51. doi: 10.1109/INFOCT.2018.8356839
- [17] A. Faeth, M. Oren, J. Sheller, S. Godinez and C. Harding, "Cutting, Deforming and Painting of 3D meshes in a Two Handed Viso-haptic VR System," *2008 IEEE Virtual Reality Conference*, Reno, NE, 2008, pp. 213-216. doi: 10.1109/VR.2008.4480776
- [18] S. Schkolne, M. Pruetz, and P. Schröder. "Surface drawing: creating organic 3D shapes with the hand and tangible tools". 2001. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 261-268. DOI=<http://dx.doi.org/10.1145/365024.365114>
- [19] R. Arora, R. H. Kazi, T. Grossman, G. Fitzmaurice, and K. Singh. "SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ". 2018. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Paper 185, 15 pages. DOI: <https://doi.org/10.1145/3173574.3173759>

- [20] D. Guinness, A. Jude, G. M. Poor, and A. Dover. "Models for Rested Touchless Gestural Interaction". 2015. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (SUI '15)*. ACM, New York, NY, USA, 34-43. DOI: <https://doi.org/10.1145/2788940.2788948>
- [21] F. Nunnari, M. Bachynskyi, and A. Heloir. "Introducing postural variability improves the distribution of muscular loads during mid-air gestural interaction". 2016. In *Proceedings of the 9th International Conference on Motion in Games (MIG '16)*. ACM, New York, NY, USA, 155-160. DOI: <https://doi.org/10.1145/2994258.2994278>
- [22] S. Davis, K. Nesbitt, and E. Nalivaiko. "A Systematic Review of Cybersickness". 2014. In *Proceedings of the 2014 Conference on Interactive Entertainment (IE2014)*, Karen Blackmore, Keith Nesbitt, and Shamus P. Smith (Eds.). ACM, New York, NY, USA, Article 8, 9 pages. DOI=<http://dx.doi.org/10.1145/2677758.2677780>
- [23] "Blender". Blender.org. 2018. [Online]. Available: <https://www.blender.org/>. [Accessed: 07-Jul-2018].
- [24] "Maya". Autodesk. 2018. [Online]. Available: <https://www.autodesk.ca/en/products/maya/overview>. [Accessed: 07-Jul-2018].
- [25] "3D Studio Max". Autodesk. 2018. [Online]. Available: <https://www.autodesk.ca/en/products/3ds-max/overview>. [Accessed: 07-Jul-2018].
- [26] "AutoCAD". Autodesk. 2018. [Online]. Available: <https://www.autodesk.ca/en/products/autocad/overview>. [Accessed: 07-Jul-2018].
- [27] "SketchUp". Sketchup.com. 2018. [Online]. Available: <https://www.sketchup.com/>. [Accessed: 07-Jul-2018].
- [28] "Qubicle". Minddesk.com. 2018. [Online]. Available: <https://www.minddesk.com/>. [Accessed: 07-Jul-2018].
- [29] "MagicaVoxel". Github. 2017. [Online]. Available: <https://ephtracy.github.io/>. [Accessed: 07-Jul-2018].
- [30] F. Steinicke, T. Ropinski G. Bruder and K. Hinrichs. "3D Modeling and Design Supported Via Interscopic Interaction Strategies". 2007. *Human-Computer Interaction. HCI Applications and Services. HCI 2007. Lecture Notes in Computer Science, vol 4553*. Springer, Berlin, Heidelberg. Doi: [https://doi-org.ezproxy.library.dal.ca/10.1007/978-3-540-73111-5\\_127](https://doi-org.ezproxy.library.dal.ca/10.1007/978-3-540-73111-5_127)
- [31] T. Oe, B. Shizuki, and J. Tanaka. "Scan modeling: 3d modeling techniques using cross section of a shape". 2012. In *Proceedings of the 10th Asia pacific conference on Computer Human Interaction (APCHI '12)*. ACM, New York, NY, USA, 243-250. DOI: <https://doi.org/10.1145/2350046.2350096>

- [32] T. Grossman, R. Balakrishnan, G. Kurtenbach, G. Fitzmaurice, A. Khan, and B. Buxton. "Interaction techniques for 3D modeling on large displays". 2001. *In Proceedings of the 2001 Symposium on Interactive 3D graphics (I3D '01)*. ACM, New York, NY, USA, 17-23. DOI=<http://dx.doi.org/10.1145/364338.364341>
- [33] J. Bloomenthal and B. Wyvill. 1990. Interactive techniques for implicit modeling. *SIGGRAPH Comput. Graph.* 24, 2 (February 1990), 109-116. DOI: <http://dx.doi.org/10.1145/91394.91427>
- [34] T. Igarashi, J. F. Hughes. "A suggestive interface for 3D drawing". 2006. *In ACM SIGGRAPH 2006 Courses (SIGGRAPH '06)*. ACM, New York, NY, USA, Article 10 . DOI: <https://doi.org/10.1145/1185657.1185771>
- [35] "Orion Beta". *Leap Motion / Developer*. 2017 [Online]. Available: <https://developer.leapmotion.com/orion/>. [Accessed: 07-Jul-2018].
- [36] A. Stavrinou. "Unity Voxel Tutorial Part 1: Generating Meshes from Code," *Student Game Dev*, 2013. [Online]. Available: <http://studentgamedev.blogspot.com/2013/08/unity-voxel-tutorial-part-1-generating.html>. [Accessed: 07-Jul-2018].
- [37] S. Gordon. "Parsing Wavefront OBJ files in C#". *Stefan Gordon.com*. 2015. [Online]. Available: <https://www.stefangordon.com/parsing-wavefront-obj-files-in-c/>. [Accessed: 07-Jul-2018].
- [38] D. Chen, X. Tian, Y. Shen and M. Ouhyoung. " On Visual Similarity Based 3D Model Retrieval". 2013. *Computer Graphics Forum*, 22: 223-232. doi:10.1111/1467-8659.00669
- [39] B. Brown, S. Reeves, and S. Sherwood. "Into the wild: challenges and opportunities for field trial methods". 2011. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1657-1666. DOI: <https://doi.org/10.1145/1978942.1979185>



## APPENDIX A

### Drawing Times

Table 4 Participant times for VR Tutorial, VR Drawing, and Desktop Drawing rounded to nearest second

Participants	VR Tutorial Time	VR Drawing Time	Desktop Time
p0	07:20	02:48	12:04
p1	09:03	10:08	09:20
p2	19:54	22:22	10:07
p3	06:35	27:34	10:10
p4	16:25	10:33	39:29
p5	04:53	06:24	04:28
p6	11:15	12:25	10:06
p7	07:48	04:47	07:39
p8	11:44	16:18	14:52
p9	13:37	36:32	10:01
p10	17:43	35:12	23:38
p11	13:23	11:23	08:05
p12	13:32	12:06	09:48
p13	09:39	08:19	08:47
p14	04:24	42:07	30:03
p15	11:46	04:03	04:33
p16	18:37	22:46	24:54
p17	10:39	07:59	08:27
p18	08:33	08:28	05:41
p19	06:31	04:03	11:39

## APPENDIX B

### Questionnaire Questions

I would use a keyboard application like this one in the future.
The keyboard application was easy to use.
In the keyboard app, I was able to draw/delete voxels where I wanted.
In the keyboard application, I was able to draw a model that matched the reference model to my satisfaction.
I was able to use the keyboard drawing platform to rotate the drawing comfortably.
What did you like/dislike about the keyboard application?
I preferred using the VR application over the keyboard application.
In the VR app, I was able to draw the voxels where I wanted them to go.
In the VR app, I was able to delete the voxels I wished to delete.
It was easy to save the finished VR model.
The icons in the VR tools menu were easy to understand.
The VR tools menu was easy to open.
In the VR application, I was able to accurately recreate the reference model.
I was able to use the VR drawing platform to move and rotate the drawing comfortably.
In the VR version, it was easy to switch between the draw and delete tools.
My arms became tired before the test was over.
The VR tutorial was easy to follow and remember.
Overall, the VR application was easy to use.
I was physically comfortable using the VR application.
I was pleased with the VR drawing I created.
I would like use a VR application like this one in the future.
Which application (desktop/keyboard, Virtual Reality) did you prefer?
What did you like/dislike about the VR application?
What changes would you make to the VR application?