ual: creative computing institute

University of the Arts London: Creative Computing Institute
22/23 Creative Making: MSc Advanced Project

TactiScape VR

Creating and Studying the Effects of Simulated Tactility In 3D Sculpting through Free Form Vertex Displacement of Landscape Meshes in Virtual Reality.

Author: Advait Ukidve

Supervisor: Phoenix Perry

Video: https://www.youtube.com/watch?v=_FOe3XUQhgI

MSc Creative Computing
November 2023

i. Abstract
Building on the old computing adage of tactility enhancing the tangibility/comprehensibility of a computers, this study investigates and measures the impact of simulated tactility (or pseudo-tactility) on the tangibility, immersion, efficiency, and enjoyability of a 3D Sculpting System. It proposes and documents the development of TactiScape VR - which attempts to simulate the tactility of the physical world in VR through the use of 3D visualisation, corporeal interaction, and realistic feedback. Through a sequence of targeted user tests, it collects and analyses evidence of enhanced tangibility, visualisation, and enjoyability of the core pseudo-tactile Sculpting workflow at the cost of efficiency.

ii. Keywords
Tactility, Simulated Tactility, Tangibility, Comprehensibility, Virtual Reality, 3D Sculpting, Landscape Sculpting, Corporeal Interaction, etc.

iii. Acknowledgements
It has been an absolute privilege to spend the last year of my life at the Creative Computing Institute in the midst of a kind, inclusive, and passionate community of like-minded techies. I would like to thank the institute for giving me the opportunity and the ability to grow my expertise in trying to explore the ideas I'd been toying with in my own head. I'd like to thank Phoenix Perry, our course leader, and my supervisor, for the insights and discussions that were crucial to the development of this work. I'd like to thank Lieven van Velthoven for all the technical support and, at times, going beyond and turning the design and optimisation of the project into a personal mission of sorts. I'd like to thank all my peers and friends for their constant support, suggestions, and critique. Finally, I'd also like to thank the often-under-appreciated community of YouTube tutors and internet forum dwellers whose identities might be anonymous, but their impact is immeasurable towards this effort.

iv. Contents

v. List of Figures

1. Introduction to 3D Sculpting

1.1 What is 3D Sculpting?

3D Modelling for CAD (Computer Aided Design) has been a technique used in various industries like design, engineering, and architecture starting in the late 1960s. 3D Sculpting (3DS) is a more recent subset of 3D Modelling (3DM) and was developed for 3DM workflows that are less parametric in nature.

3DM is typically better suited for creating geometric shapes and hard surface models. It relies on mathematical algorithms to create precise shapes and is well-suited for models that require accuracy and precision. But this brings with it systems and methods of interaction that are centred around precision unlike traditional means of modelling. Visualising 3D models by zooming in, and making precise manipulations by pulling edges and vertices, although normal to professionals today, represents a paradigm of interaction quite unlike modelling physical objects.

3DS, on the other hand, is better suited to creating organic models such as characters, creatures, and landscapes, and more closely resembles traditional means of model-making. It allows artists to work with 'virtual clay', adding and removing material to create details, folds, and textures. Critically, unlike 3DM, it works using tools that simulate real-world sculpting tools like brushes, knives, and chisels that humans already have a cognitive understanding of. Consequently, 3DS is often easier to pick up and experiment with. For example, holding a virtual brush against the model deforms it in a manner similar to holding a sculpting tool against a physical block of clay. Using a 'pinch brush' makes the virtual model create a peak in the same way physical clay would if pinched by one's fingers.

This fundamental difference in modelling approach often makes 3DS far more tangible - easier to pick up for untrained professionals and smoother to use in the long run due to the predictability in its workflow. This predictability, coupled with no limitations of physical materials and forces, makes 3DS the most versatile and intuitive system for organic shape modelling - fusing the advantages of the virtual world with intuitive physical interactions.

But this fusion is not seamless.

1.2 What 3D Sculpting is not.

State-of-the-art 3DS, best evidenced in commercial applications like Maxon's Cinema4D (Maxon, 1990) and Zbrush (Maxon, 1999), Autodesk's Maya (Autodesk UK, 1998), and Blender (Blender Foundation, 2002), still necessitate use via 2-dimensional means of interaction. This is evident in both the visualisation and the interaction afforded by the system. For example, sculpting a 3D object in a desktop application like Blender usually involves the following steps repeated cyclically (See Figure 1.1):

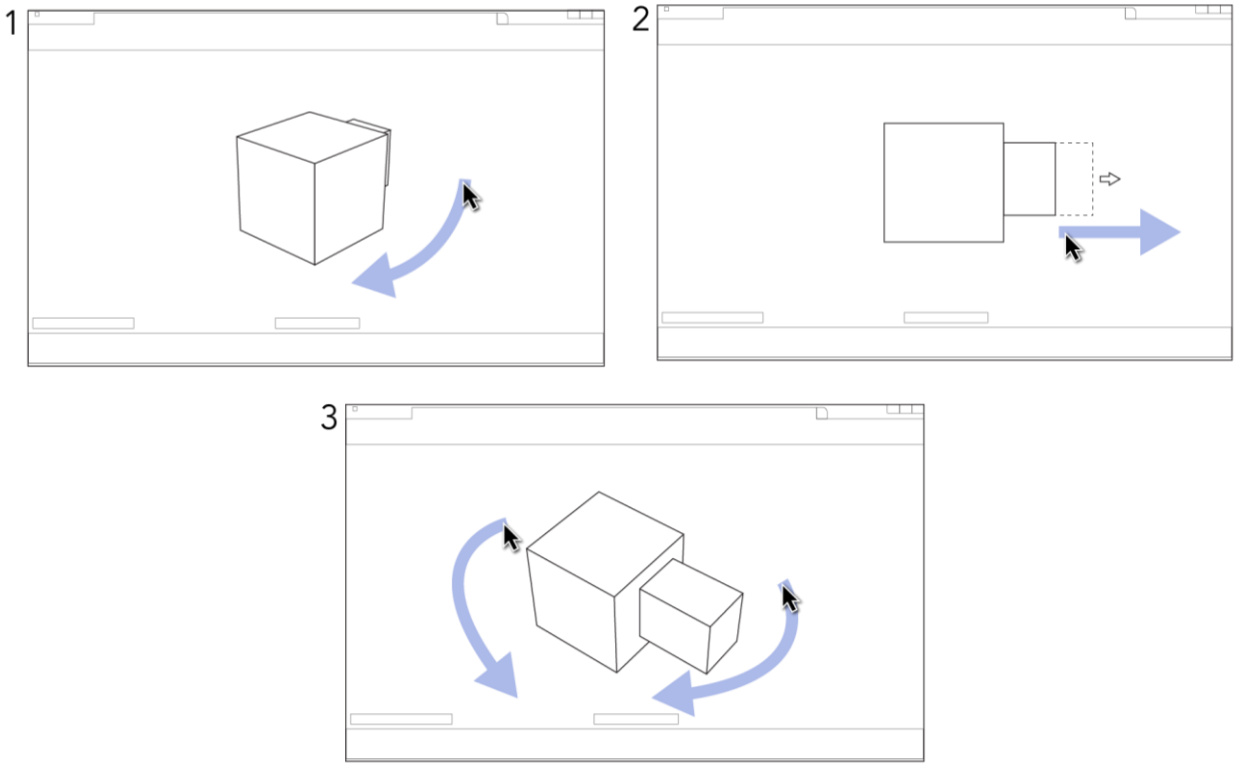


Figure 1.1: Visualisation and manipulation of 3D CAD Models with current 2D means.

1. Reading a decomposed 2-dimensional representation of a 3-dimensional object off a computer screen, usually by rotating it around to view it from a series of orthographic angles.
2. Manipulating a surface by means such as pulling or pushing in 3 dimensions through 2-dimensional mouse movements.
3. Ensuring the resultant manipulations are consistent with expectations by navigating the 3D again.

This application of a 2D interface to 3D object manipulation is an artefact of CAD's evolution rather than a deliberate and considered design choice (Zmoelnig, 2000) and proves to be a bottleneck to 3D designers as it forces them to decompose an essentially three dimensional task into a series of 1D and 2D tasks (Liang and Green, 1994). This leads to significant drawbacks - chief of which is an increased cognitive load on the user - and results in the usage of such systems often being restricted to trained professionals.

Furthermore, some researchers argue that the fundamental constraints of 2D representation and manipulation also limit creativity in 3D artists (Kim and Maher, 2005) and make them less immersed in the process.

3DS's promise of tangibility can hence only be fulfilled by evolving the systems of visualisation and interaction to match its fundamentally intuitive approach.

2. Literature Review

2.1 Tangible Computing and 3D Sculpting - Background

To understand the literature and experiments in tangible 3DM and 3DS, we must first take a step back and view tangibility in computing as a whole. Tangibility in HMI has been a topic of interest for almost as long as computation itself. Researchers, both away from and at the forefront of commercial development, have long toyed with ideas to make the increasingly complicated virtual world easy to understand and interact with based on metaphors present in everyday life. The most feasible ones so far have proven to be visual metaphors – represented broadly in what are termed Graphical User Interfaces (GUIs). GUIs, pioneered by XEROX in 1981 (Smith *et al.*, 1982), popularised by the Macintosh in 1984 (Apple Computer Inc., 1987), and widespread through the pervasiveness of Microsoft Windows, have become, by far, the easiest and most scalable interfaces in modern day computing. Yet they are far from the most intuitive.

Grounded on early research and ideas such as 'Ubiquitous Computing' (Weiser, 1991), 'Back to the Real World' (Wellner, Mackay and Gold, 1993) and 'Graspable User Interfaces' (Fitzmaurice, 1996), Ishii and Ullmer proposed the idea of Tangible User Interfaces (TUIs) in their 1997 paper titled 'Tangible Bits' (Ishii and Ullmer, 1997). These TUIs are positioned as an alternative to traditional GUIs. Building on thousands of years of cognitive familiarity with grasping and manipulating physical objects, Ishii and Ulmer proposed interfaces that allows users to manipulate 'bits' in virtual space by coupling them with everyday physical objects. They illustrated these ideas with research prototypes such as metaDESK, ambientROOM, transBOARD, and Ishii's later works such as Illuminating Clay (Ratti *et al.*, 2004) - extending and evolving the desktop metaphor to a series of tangible interactions. Works such as Wendy Mackay's flight strips (MacKay, 1999), Schafer et al.'s approach for simultaneous building of real and digital models (Schäfer, Brauer and Bruns, 1997), Build-IT (Fjeld, Bichsel and Rauterberg, 1999), and AlgoBlocks (Suzuki and Kato, 1995), built on similar ideas creating interfaces that placed a deep emphasis on tangibility, imbibing them with (a) physical world contexts; and (b) corporeal interactions.

Tangibility in 3D Modelling systems, following in the same path, was built on two unique functions that can be isolated for the sake of analysis – Tangible Visualisation and Tangible Manipulation. Visualisation refers to the 'output' of the system to the user - the representation of a 3-dimensional virtual objects through hardware. Manipulation, more broadly, represents the 'input' to the system - the user actions and commands that serve the purpose of observing, constructing, and editing virtual 3D models (see Figure 2.1). The following section explores some projects and literature in the field that serve a common interest of creating more tangible means of interaction by increasing the familiarity and, hence, intuitiveness of a machine.

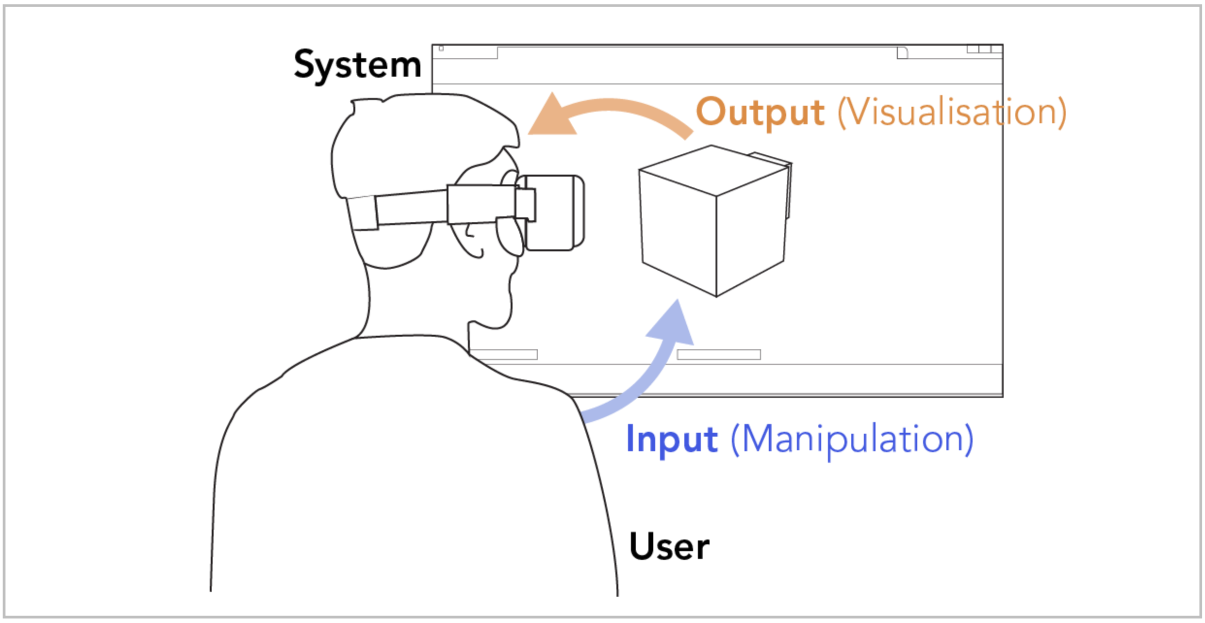


Figure 2.1: The input and the output of the system

2.2 Visualisation-Only Systems (Tangible Visualisation)

Several attempts have been made over the years to aid the visualisation of 3D models and make them more tangible. These include attempts using projection techniques, head mounted displays (HMDs) for AR and VR, and holograms akin to science fiction. All these projects serve the same end - that of representing a 3-dimensional object in 3 dimensions, as opposed to traditional 2-dimensional screens.

Most visualisation-only systems[1], from the early 3DM (Butterworth *et al.*, 1992) to the current generation Sketchup Viewer (Trimble Inc., 2000) follow this basic premise of an HMD used to visualise three dimensions in a more intuitive manner. Sketchup Viewer uses device cameras to superimpose models built in the desktop application over the real world at original scale. It allows for quick manipulations such as orienting, moving, and scaling. It acts as an easy visualisation tool for both the designers and the consumers, albeit while necessitating VR headsets. Arkio (Arkio ehf., 2022), built for architectural visualisation in AR, intelligently marks surfaces in existing environments and transforms them into virtual 3D models to be viewed at scale. R. Sidharta's 'Augmented Reality Tangible Interface' (Sidharta, 2005) uses an HMD along with gesture recognition to browse between, and position and orient 3D models for one or more people to enable collaborative inspection.

Elsewhere, visualisers such as the Holografika's (Holografika, 2004) HoloVizio 128WD Display, Actuality's Perspecta 3D Display (Donelan, 2010), Felix 3D Laser Display (Bahr *et al.*, 1996), and DepthCube (Sullivan, 2004) use pseudo-volumetric systems (Eliab Z. Opiyo and Horvath, 2010), letting users visualise 3D objects holographically as if co-existing in the same 3D space. These systems aid in overall intuitiveness but fall short on multiple accounts of viability such as model scalability, image quality, interaction with the system, etc. (E.Z. Opiyo and Horvath, 2010).

Visualisation-Only systems thus offer more tangible visualisation, but at the cost of often-unfeasible means (such as elaborate physical rigs and/or headsets) and limited use cases.

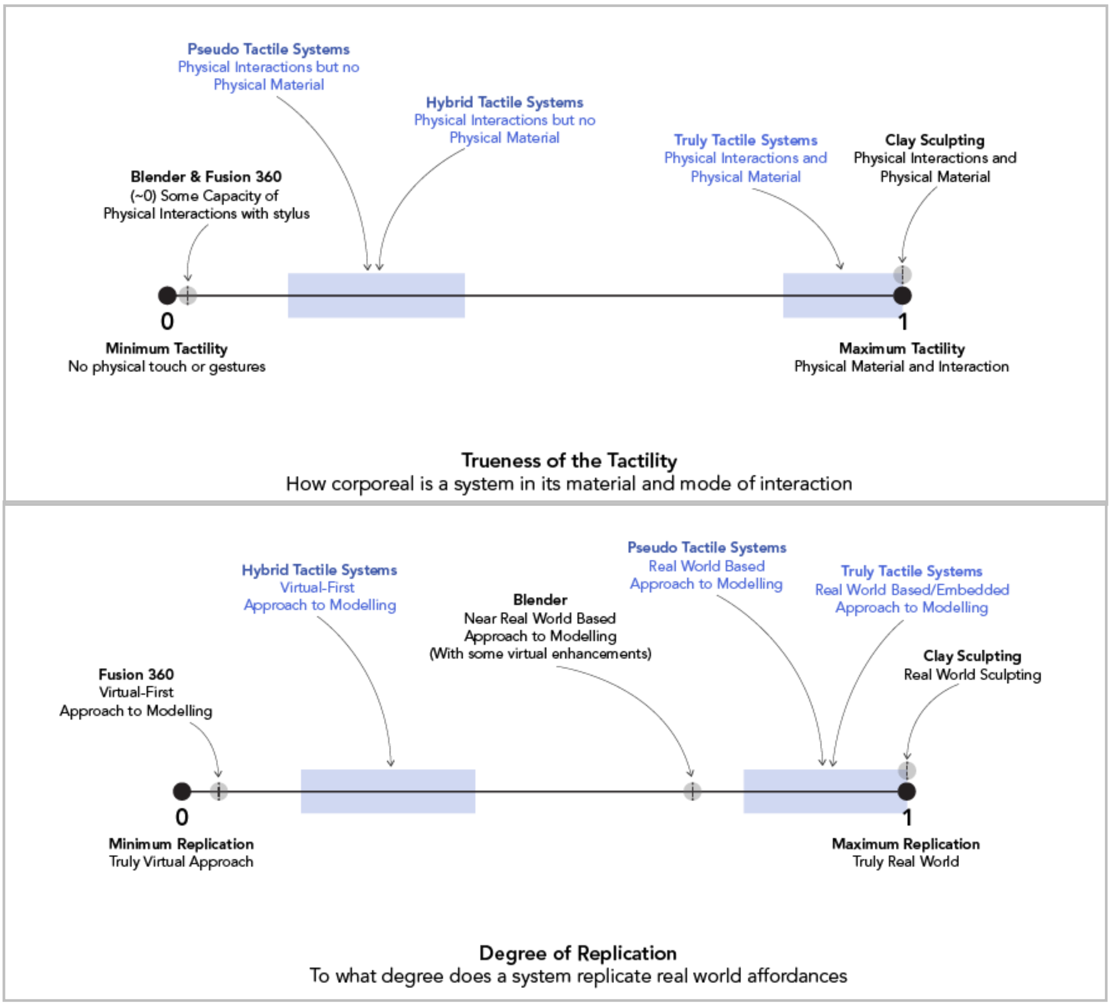
[1] Although all these applications allow for some amount of manipulation, it is mostly surface level and cannot be attributed as belonging to the conventional 3D modelling workflow.

2.3 Full Suite Systems (Tangible Manipulation + Visualisation)

Going beyond mere visualisation, several works attempt to create a tangible system for the manipulation of 3D objects. These often work in tandem with tangible visualisation systems to afford the users the ability to not just view, but also construct and edit 3D objects.

The primary and long explored way of furthering tangibility in CAD is through tactility (Buxton, 2007). Tactility, in this scope, can be defined as the capability of the system/software (or more specifically, the 3D object in question) to be touched – whether physically or by means of simulation using haptic feedback. Such tactility ensures a deeper understanding of the system one is exploring by mimicking cognitively familiar experiences with the spatial, relational, and constructive advantages of physical interfaces (Ullmer and Ishii, 2000). They allow for quick and intuitive prototyping based on real world physics and afford the user predictable experimentation leading to deeper understanding of the system.

For the sake of review and evaluation, this paper proposes the categories ‘Truly Tactile Systems’, ‘Hybrid Tactile Systems’, and ‘Pseudo Tactile Systems’ based on the level, function, and ‘trueness’ of its tactility (See Figure 2.2).



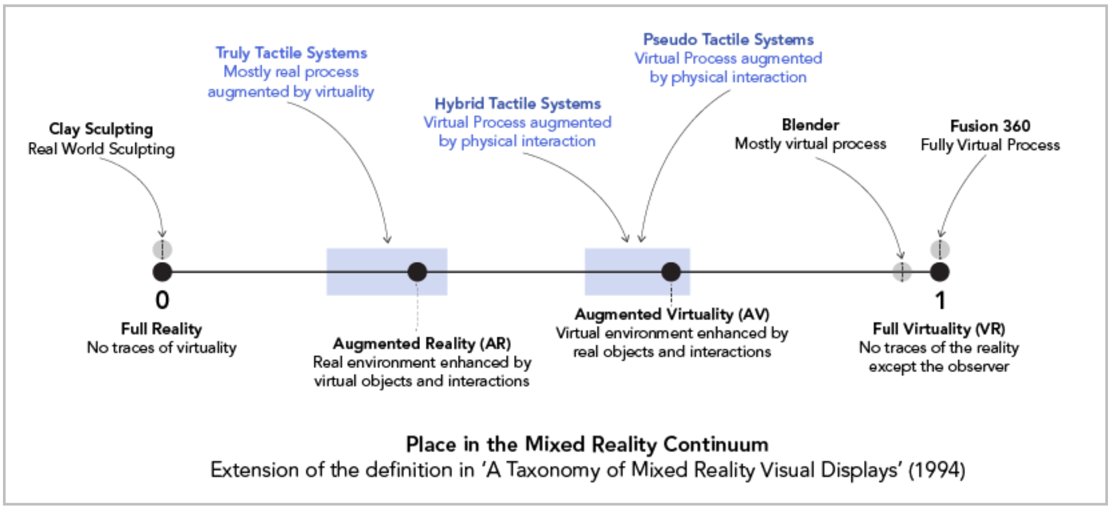


Figure 2.2 Diagram comparing existing 3D Sculpting systems and the various categories of Tactile systems based on – the trueness of its tactility, degree of replication, and its place in the Mixed Reality Continuum.

2.3.1 Truly Tactile Systems

The first kind of tactile system is one that embraces the physical world and all its affordances as a whole, while offering virtual 3D models as aa singular or additional output. These systems, classified as 'truly tactile' systems, translate physical models to virtual reality allowing users to interact normally with physical material.

Projects like SM-Block (Son, Jang and Choi, 2019) for example, use physical blocks like cubes, cylinders, and cones that snap together to form complex 3D models - replicating voxel modelling physically. Various 3D scanning technologies (Daneshmand *et al.*, 2018) encourage users to model physical sculptures and then translate the model into a point cloud using various photogrammetric and infrared sensing techniques. Various technologies are optimised towards capturing different types of models ranging from sand-based sculptures to hard surface voxel-like models. Fabric3D (Leal *et al.*, 2011) takes the unique approach of 3D modelling using fabric relying on a system of 8 depth-sending cameras to capture intricate folds and replicate them virtually.

Truly Tactile Systems (TTS), overall, allow for easy and intuitive modelling in physical space while requiring only a little bit of learning. Yet, they suffer from some of the same limitations that necessitated virtual 3D modelling in the first place. Often, the models required by engineering professionals are extremely intricate, of varying scale, or generally unviable of being constructed physically. Additionally, more artistic models used in video game design and animation require movement and malleability that is difficult to build with physical material. Moreover, given the space and maintenance they require to be set up, truly tactile systems can be deemed unviable at scale.

While inherently intuitive, truly tactile systems suffer from the obvious constraints of 'true' tactility - that of physical material and real-world physics.

2.3.2 Hybrid Tactile Systems

A second approach to tangible modelling systems is using Hybrid Tactile Systems (HTS) that, unlike TTSs, embrace the malleability of the virtual world that is at the very core of using CAD. The tactility of such systems is not used as the singular basis for all the system's affordances but instead as the means of interaction with virtual-first approaches to 3D modelling. As a result, these works break out from the constraints of real-world modelling and utilise functionality such as scaling, edge and vertex selection, and post-processing techniques only possible in a virtual environment, while still retaining the advantages of physical interactions.

TangiCAD (Abdelmohsen and Do, 2007) uses cubes similar to SM-Block as an alternative interface for virtual modelling. It involves multiple RF-embedded cubes that can be oriented differently to give out a range of navigation, creation, and editing commands to a virtual workspace.

C.H. Teng and S.S. Peng's unnamed system (Teng and Peng, 2017) uses a stylus called AR Pen and the Free Form Deformation method (Sederberg, 1986), to achieve the manipulation of an object's surfaces based on the deformation of an invisible hull object encapsulating it. ARpm (Fiala and Adamo-Villani, 2005) uses an HMD and a command-line interface with 3D Studio Max (Autodesk UK, 2023) to let the user see the 3D model they're operating on in AR and replicates a 3-button mouse using a hand-held pointer device. 3DM (Butterworth *et al.*, 1992) is similarly built to be an alternate interface for desktop CAD applications, replacing the 2D monitor with an HMD. Like ARpm, it lets users perform CAD functions using a hand-held pointer that accounts for direction in 3D. COVIRDS (Dani, Chu and Gadh, 1997) and Gao et al.'s unnamed system (Gao, Peng and Wan, 2000) both use a similar system but with voice commands in addition to a pointer.

Most of these works succeed in their primary intention of making 3D manipulation, in fact 3D. Yet they are often limited by the constraints of a parent CAD system and a necessary physical setup (styluses, cameras, HMDs, etc.). Most importantly, the intuitiveness of the system is brought down a mark by the inclusion of modelling techniques that don't mimic the physical world, but instead only use it as a means of interaction with complex and often unfamiliar virtual techniques.

2.3.3 Pseudo Tactile Systems

The third category, Pseudo Tactile Systems (PTSs), builds on both the previous systems and uses techniques built to closely replicate real world modelling in mixed reality environments. While not truly tactile as they don't afford physical touch, these systems simulate the tactile interactions afforded by Truly Tactile Systems while retaining the flexibility afforded by the XR displays often seen in Hybrid Tactile Systems. This combination arguably makes for the most intuitive, easy to use 3DS systems (Billinghurst and Kato, 2002).

Gravity Sketch (Gravity Sketch, 2017), available for most VR headsets, follows a Pseudo Tactile approach letting its users sketch, model, and sculpt in immersive 3D VR environments. It builds on the users' cognitive understanding of painting and interfacing objects in the physical world and translates these interactions into virtual output. Although concerns over the suitability of its use in early stages of the design process have been highlighted (National Taiwan University of Science and Technology, Taiwan and Lin, 2022), it succeeds in making the process of 3DM/3DS in VR intuitive and predictable. AdaptiBrush (Rosales *et al.*, 2021), posed as an alternative to a singular tool in Gravity Sketch, lets the users draw ribbon-based forms in VR with physical interactions that draw from the physics of ribbons in rhythmic gymnastics. Kim et al.'s unnamed system (Kim *et al.*, 2005) uses hand gestures to manipulate 3D objects by directly replicating the gestures used to perform the same tasks in the physical world. Although limited in its functionality, it allows users to predictably perform 3D manipulations upon objects visible through a 2D display. Adobe Medium (Adobe UK, 2016) and the now discontinued Google Tilt Brush (Alphabet Inc., 2015) use an additive/subtractive clay modelling system in VR that strongly resembles desktop 3D software.

While these come closest to the proposed tool, there are no significant studies that evaluate their impact on intuitiveness and ease in cognition, and possible effects on creativity, engagement, and efficiency of modelling when used in the design process.

2.4 Evaluation of Current State

Metric	Truly Tactile Systems	Extended Tactile Systems	Pseudo Tactile Systems
Trueness of Tactility	High	Low	Low
Degree of Replication	High	Low	High
Intuitivity of Means of Interaction	High	Low	High
Intuitivity of Sculpting Process	High	Low	High
3D Visualisation Capability	High	Medium	Medium
Versatility of Possible 3D Manipulation	Low	High	Medium
Ease of Setup	Low	Medium	Medium
Capability to Expand Functionality	Low	High	Medium

Figure 2.3: Table showing evaluation of the 3 categories of tactile sculpting systems along crucial metrics.

Between them, the various projects and approaches to tangible 3D sculpting offer formidable methods, critical design insights, and potential pitfalls to take into consideration while designing the systems of the future. TTSs highlight the importance of physical interactions, spatial sensibilities, and tactility. Where these fail, HTSs highlight the importance of logistically simpler means of interaction and building functionality beyond that afforded by physical material. PTSs present a unique and less-explored opportunity to build the most intuitive and versatile means of 3D Sculpting - given their approach of combining intuitive means of interaction with intuitive sculpting techniques. Even so, these projects often suffer from the same concerns over the long-term effects and scalability of VR technology, the implication of simulated tactility, and the lack of formal evidence of their benefits.

This paper attempts to address this gap in research by proposing a PTS built bearing the learnings from its predecessors in mind and attempts to evaluate the validity and potential benefits of such means of 3D sculpting within the design process. It poses the questions that stick out the most when reviewing current literature:

“How far can pseudo-tactility (or simulated tactility) go in reproducing the benefits of tactility? Does pseudo-tactility, when done right, produce the same effect on the intuitiveness of a system as TTSs? Does it create a system for 3D sculpting that is better than the existing ones? If yes, what metrics can we measure this against?”

3. Methodology and Design Proposal

To answer the research question, this paper proposes a tool for Pseudo Tactile 3D Sculpting – built to replicate the act of physical sculpting and all its affordances in a system that provides tangible input and output in the form of a VR environment and a physical interaction model.

3.1 Design Methodology

In order to design the tool, a detailed study was done with professional and student sculptors – both physical fine art sculptors and digital sculptors – to understand the techniques and mental models involved in creating their work. Physical sculptors were tasked to create simple models and their process and interactions with clay were observed to understand the essential affordances of tactile materials that were to be replicated. This was followed by an informal conversation to understand the design process and the thinking involved. Furthermore, the successes and failures of the systems analysed in the literature review whose shoulders TactiScape stands on were carefully reviewed. Finally, initial sketches of the proposed system were drawn and underwent critical review with the sculptors before development began. Section 4.1 highlights and elaborates on the specific design decisions born out of this process.

3.2 Evaluation Methodology

In order to evaluate the tool practically, the testing focused on gathering qualitative and quantitative data comparing TactiScape VR to an instance of existing 3D Sculpting software. For the purpose, it utilised Landscape Sculpting in Blender which depends on both 2D visualisation and 2D mouse-based interaction. The participants were asked to perform similar 3D sculpting tasks in both environments with a time constraint, followed by a verbal interview on their experiences in the same. This aimed to gauge the differences between the two along metrics such as intuitiveness, efficiency in performing task, ease of visualisation, impact on individual design process, overall immersion and enjoyability of the experience, etc. The results of these have been highlighted in Section 5

4. TactiScape VR - The Tool

TactiScape VR was conceived as a 3D Landscape Sculpting tool that **simulates the tactility of physical clay sculpting** by replicating the process in VR – through 3D visualisation, corporeal interaction, and multi-modal feedback. It is built using a novel method called **Free Form Vertice Displacement** that translates mesh vertices using velocity inherited from the sculptors' hand behaviour. Based on the early research conducted with sculptors, some key affordances of physical sculpting were deemed necessary to the proposed tool. The following section describes and reasons the design decisions (Section 4.1), and documents the development of the tool (Section 4.2)

4.1 Design Choices

4.1.1 Replicating Real World Physics

In order to replicate physical sculpting, the virtual clay needs to react to the hand presence and applied force in a 'natural' way. To account for the corporeal variations in sculpting tools, TactiScape utilises a system of multiple swappable 'brushes'[2] meant to closely replicate different methods of clay sculpting. These brushes act as a source of force incident on the landscape. The brushes account for:

(a) Shape and Nature of Impact

The shape of the brush is crucial to the magnitude and direction in which the virtual clay is displaced at every point. For example, replicating the pushing of a fist into clay would mean the incident shape of the brush is spherical applying force radially outwards. 'Patting down' on clay necessitates a cuboidal or conical area of influence in the direction of the hand's velocity applying force approximately perpendicular to the clay surface's slope. Creating holes in clay with a finger means applying force incident from a small cylinder but limiting the impact radius.

(b) Force and Velocity

The force incident on the clay is directly tied to the velocity (magnitude and direction) of the hand. Rapid movement in the clay's direction would displace the clay further. Movement parallel to the clay surface would displace the clay sideways. Pinching the clay would apply force towards a 'peak' calculated by the vertices' average location, and faster pinching would create more dramatic peaks.

4.1.2 Visualisation in Virtual Reality

In order to create the most convincing simulation, it is crucial to let users visualise the landscape in 3D and to let them manipulate it through 3D gestures. TactiScape hence uses VR in place of a traditional 2D monitor. This allows the users to visualise their landscape at scale and traverse it in line with expectations from a physical space.

[2] An ideal programme would adapt its brushes dynamically to the user's gesture but that was beyond the scope of the project at this stage.

4.1.3 Systems of Feedback (Haptic, Visual, and Auditory)

To aid the simulation, TactiScape provides haptic vibrations and sound effects when the user interacts with the virtual clay. These interactions are chiefly of two types – (a) 'actively manipulating'; and (b) 'feeling' the clay without deforming it. Both these interactions produce slightly different sounds and sensations, and the system's output was built in order to replicate and highlight these differences. Additionally, TactiScape utilises a vertex shader and lighting to dynamically shade the landscape to aid the accurate visualisation of height during manipulation.

4.1.4 Using Tools Instead of Bare Hands

The initial iterations of TactiScape relied heavily on only using one's hands to manipulate the clay as these are most cognitively simple and hence intuitive. But as iterations went by, the idea of also replicating various sculpting tools was realised. This is in accordance with Malafouris' research (Malafouris, 2013) which points to the idea that tools, when used in conjunction with one's hands consistently, become cognitive extensions of the hands itself.

4.1.5 Landscape Generation

A critical observation from interviews with sculptors was that, often, the lack of a starting point can create a point of friction in the design process. This meant that the starting point for the sculpt had to be something more complex than a flat plane or a primitive shape. Nevertheless, an extremely complex starting shape could prove to be a hindrance as it sets a mould on one's creative process. TactiScape starts the user off with a randomly generated, realistic looking, yet uncomplex mesh to build off of. This is done using Perlin Noise (Zucker, 2017) to generate a 'primitive' landscape with large-scale generic terraforms like mountains, valleys, rivers, etc.

4.1.6 Why a Landscape Editor?

TactiScape's scope is limited to being a landscape editor that uses simple planar geometry as the base mesh to manipulate. This is due to the limitations of time and team available to build the tool. Even so, the resultant system acts as a proof-of-concept towards the development of a more versatile 3DS system using the same framework.

4.2 How It Works

4.2.1 Platform Specifications

The tool is built using C# in Unity (Unity Technologies, 2005) and the OpenXR standard (The Khronos Group, 2017) to ensure easy portability across platforms. The tool used in the research was built for and tested with the Meta Quest 2 (Meta Platforms, 2020).

4.2.2 System Overview

TactiScape VR is built using a variety of scripts that each perform specific functions:

a) The Mesh Generator creates a Unit Mesh object from an array of vertices (each with their 3D location vectors) and an ordered array of triangles.
b) The Mesh Interactor takes input from the VR Headset and Controllers such as position, orientation, and button press to manipulate the positions of the vertices in real time.
c) Other scripts provide visual, haptic, and auditory feedback, GUI, animations, exporting, and landscape scaling.

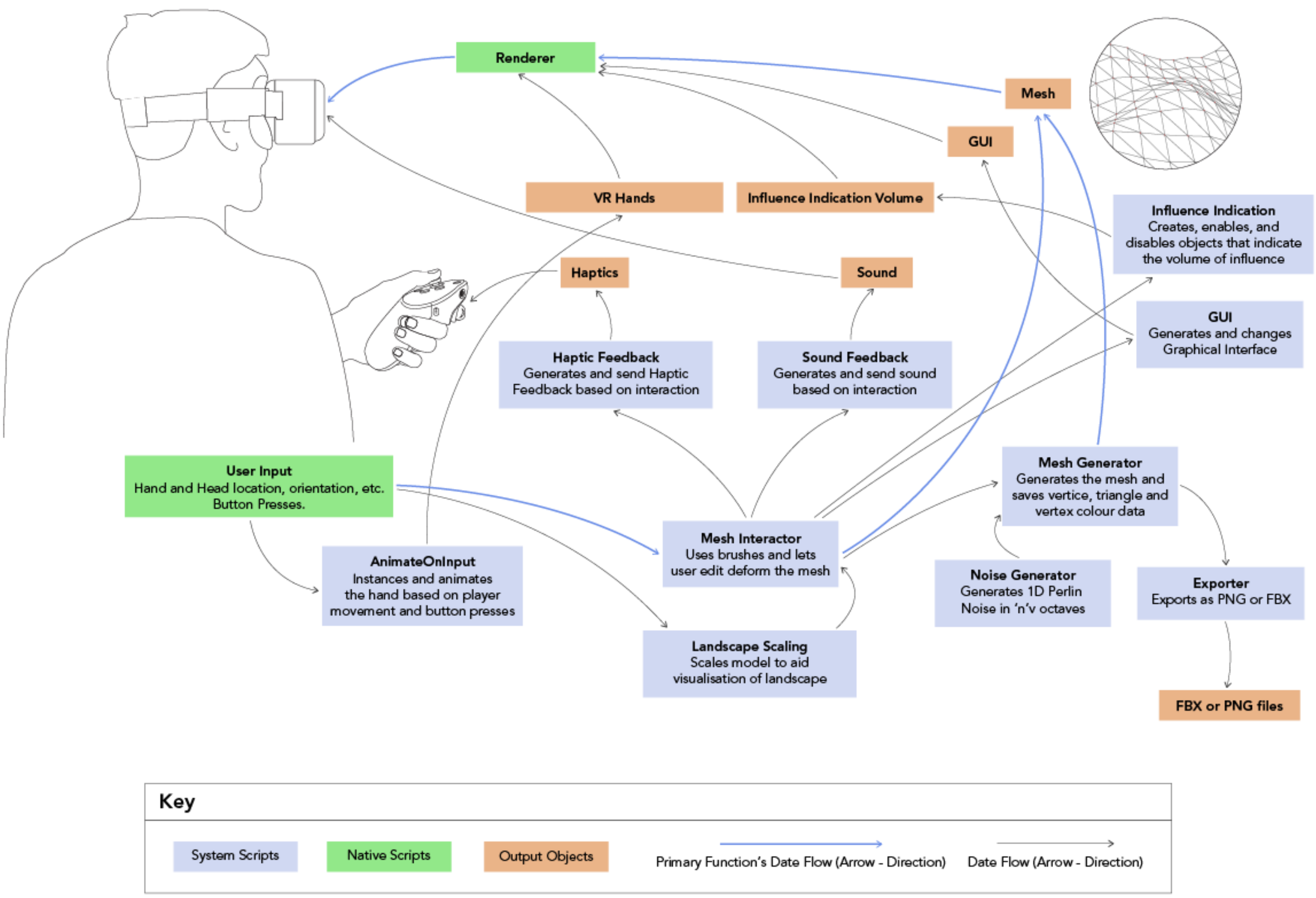


Figure 4.1: System overview and the flow of data to and from every component script of TactiScape VR.

4.2.3 Creating a Landscape Mesh (Landscape Generator)

The landscape generator creates a Unity Mesh plane object in 4 steps and by providing Unity with ordered arrays of vertices, triangles, and normals and vertex colours:

(a) Creating Vertices

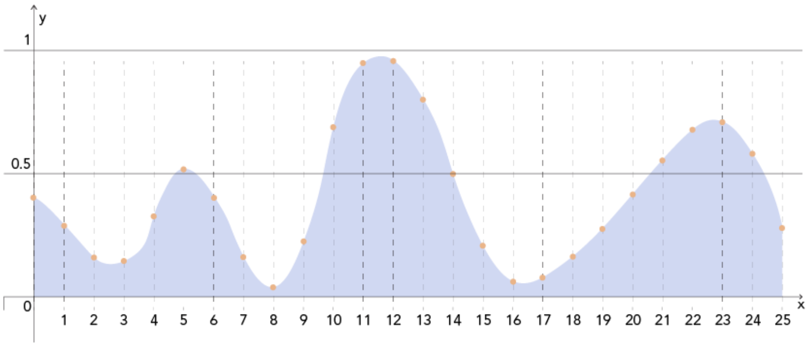
The vertices are created as 3D float vectors of the format (x, y, z) that record respective Cartesian locations. The x and z axis are fixed values with the final array looking as follows:

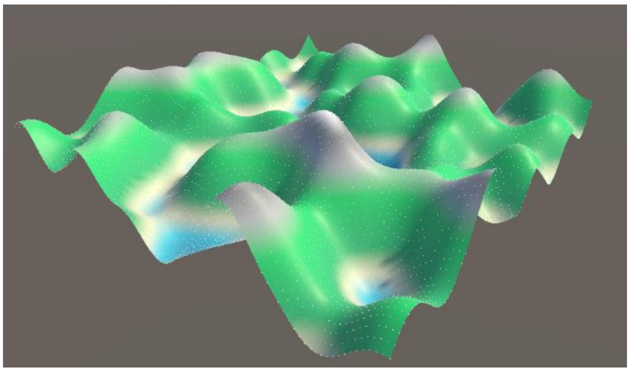
$$[(0, y_0, 0), (0, y_1, 1), (0, y_2, 2) \ldots\ (n+1, y_{n*m-2}, m-1), (n+1, y_{n*m-1}, m), (n+1, y_{n*m}, m+1)]$$

where n and m are the total number of rows and columns of quads

(b) Perlin Noise and Component Octaves

The y for each vertex represents the height and is generated according to Perlin Noise at run-time. This is done by using the x and z values of each vertex as arguments for 1D noise generation in three steps (a.k.a. octaves) of decreasing amplitudes to create local variation in the landscape while keeping a stable global structure.





Figures 4.2 and 4.3: One dimensional Perlin Noise generation for the height (y-component) of every vertex; 3D terrain generated using the algorithm.

(c) Creating Triangles and Quads

Next, triangles are created as an ordered array to indicate what vertices to connect such that the normals face upwards. For example, in the figure shown below, a quad can be created by joining a, b, and c first, in that order, and then c, b, and d. This process is repeated for all the vertices to create a plane.

(d) Vertex Shading

To create a sense of scale and easy visualisation of the landscape, the tool employs a vertex shader using the Universal Render Pipeline (URP). Each vertex is shaded according to its y with respect to pre-defined gradient. Blues lie below sea level (y = 0), and the landmass is shaded greens, greys, and whites to depict grass, rocks, and snow.

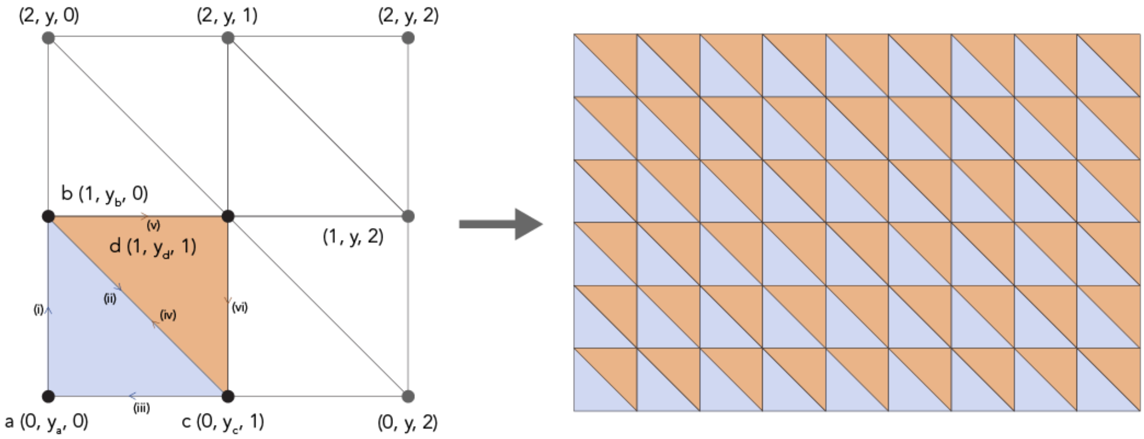


Figure 4.4: Creating triangles from the vertices to form quads and larger planes.

4.2.4 Manipulating the Mesh (Landscape Interactor)

The Landscape Interactor governs the primary function of TactiScape. It lets the user freely deform the generated mesh using data from the handheld VR Controllers such as their position, forward vector, and velocity. The data is used to create movement in individual vertices within the range of the user's hands according to a set of rules. The following section illustrates the two main rules (a) **The Brush Shape** – which decides the vertices that undergo a change in position; and (b) **The Brush Impact** - which decides the displacement of the vertices.

(a) Brush Shapes and Size

There are four main brush shapes in TactiScape bound by a common scaling factor – the brush size which indicates, generally, the furthest extent upto which a vertex can be 'selected' for movement. Selected vertices are displayed in red for quick visualisation of the brush's area of impact. The brush size can be limited to a few inches beyond the hand in virtual space to make for the most realistic replication, but this often hinders with a crucial affordance of a virtual environment – that of its ability to circumvent real world physics in the pursuit of convenience.

Spherical Shape

The simplest brush shape is spherical. This allows for the broadest range of vertices to be manipulated at once and simulates using the hand as a source of brute force by pushing against a surface using a fist. All vertices within distance 'd' from the user's right hand are selected for manipulation using the condition:

$$\left(\left(\sqrt[2]{(x_{rh}-x_v)^2+\ (y_{rh}-y_v)^2+\ (z_{rh}-z_v)^2}\right)\le d\right)$$

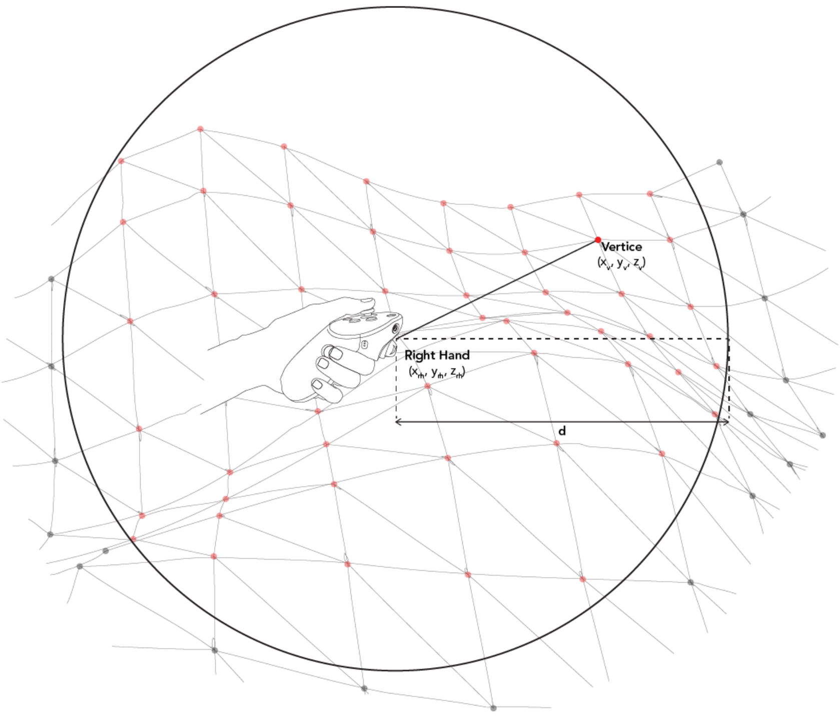


Figure 4.5: Spherical brush shape showing selected vertices of the landscape in red and unselected in grey.

Hemispherical Shape

Hemispherical brushes are more finely tuned to user needs by ridding influence from vertices that the user likely does not intend on manipulating while performing the actions. For example, pushing into a vertically inclined landscape to create an overhang necessitates using a spherical brush but the user may not intend to manipulate any vertices directly behind the hand. In this case, a hemispherical brush in the direction of the hand is more suitable.

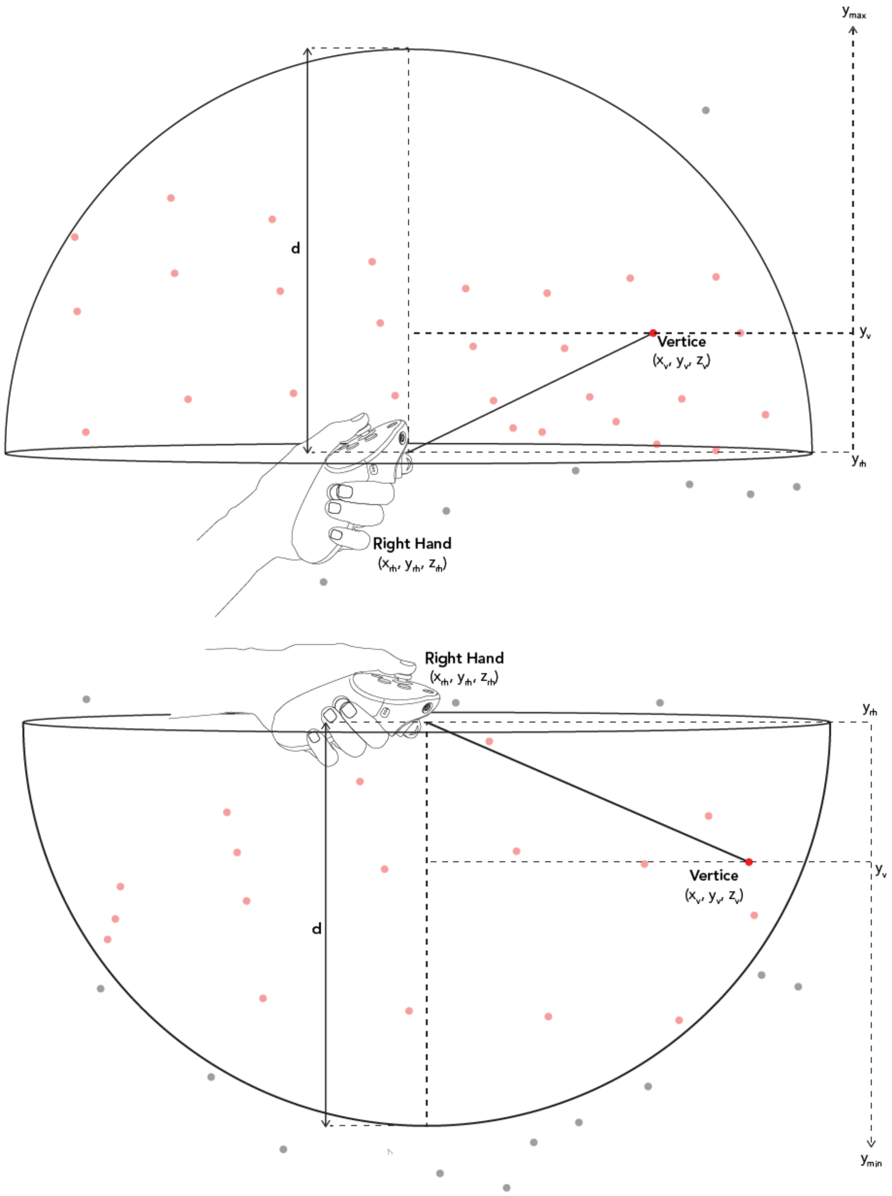
The program has 4 hemispherical brushes. Horizontal hemispherical brushes check for the height of a vertex in addition to its proximity to the right hand. Vertices are selected if they satisfy the following condition:

$$\left(\left(\sqrt[2]{(x_{rh}-x_v)^2+\ (y_{rh}-y_v)^2+\ (z_{rh}-z_v)^2}\right)\le d\right)\ \wedge\ (y_{rh}\le\ y_v)$$

(Upwards Hemisphere)

$$\left(\left(\sqrt[2]{(x_{rh}-x_v)^2+\ (y_{rh}-y_v)^2+\ (z_{rh}-z_v)^2}\right)\le d\right)\ \wedge\ (y_{rh}\ge\ y_v)$$

(Downwards Hemisphere)



Figures 4.6 and 4.7: Upwards and Downwards hemisphere brush shape.

Spherical Cone Shapes

Perhaps the most important brush shape is that of a spherical cone. This brush selects vertices that are within a spherical cone (of modifiable slope) of the hand with the area of influence closely resembling that of a flashlight. TactiScape uses two such brushes - Flashlight brush (refer Figure 4.8) and Stamp Brush (refer Figure 4.9). In addition to a proximity check, this brush shape checks the direction from it to the hand by taking the dot product of the right hand's forward vector and the direction vector joining the right hand to the vertex.

As calculating the dot product uses the cosine function, if the result tends towards 1, the object can be deemed 'in front' of the right hand. If the dot product tends towards 0, it can be said to be 'below' the right hand. The exact difference between the target dot products (0 or 1) and the resultant dot products is the 'margin' of the brush's selection and indicates the slope of the cone.

The two brushes used in the programme have a margin of 0.5 (slope of 60°). The calculations used to describe the two brushes are as follows:

The forward vector of the right hand is represented as $(x_{fv},\ y_{fv},\ z_{fv})$. The direction from the right hand to the vertex can be calculated as $((x_{rh} - x_v), (y_{rh} - y_v), (z_{rh} - z_v))$.

The dot product between these two vectors is then:

$$direction_{rh \to v} = \left(\left(x_{fv}(x_{rh} - x_v)\right) + \left(y_{fv}(y_{rh} - y_v)\right) + \left(z_{fv}(z_{rh} - z_v)\right)\right).$$

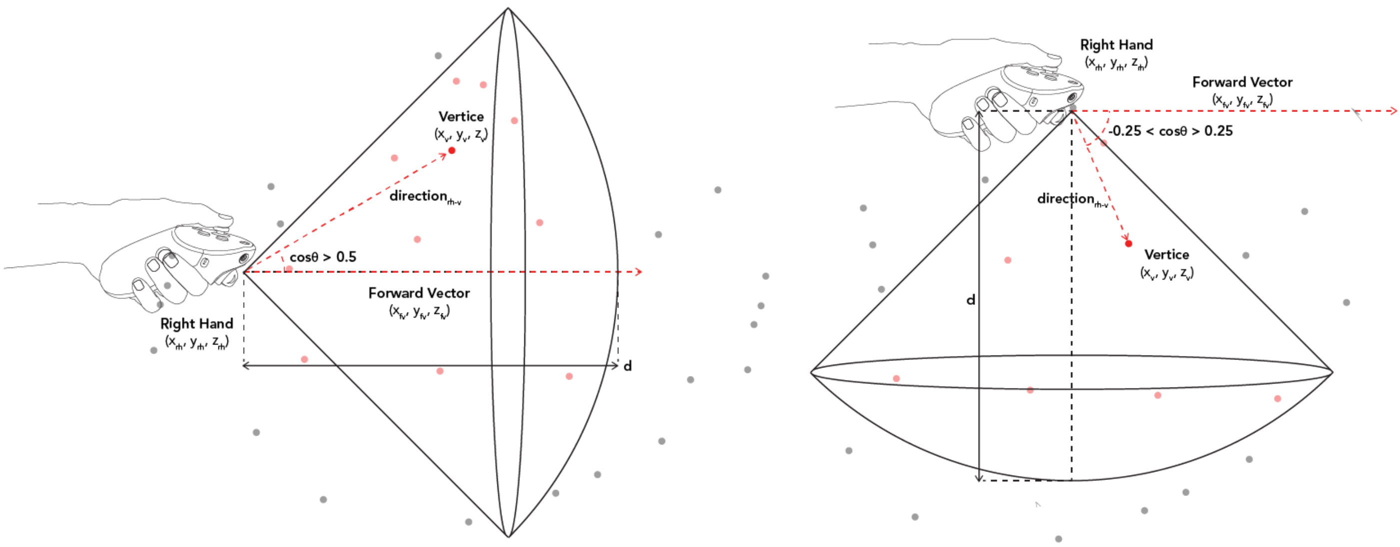
The conditions for the two brushes are:

$$\left(\left(\sqrt[2]{(x_{rh} - x_v)^2 + \ (y_{rh} - y_v)^2 + \ (z_{rh} - z_v)^2}\right) \leq d\right) \ \wedge \ \ (direction_{rh \to v} \geq 0.5)$$

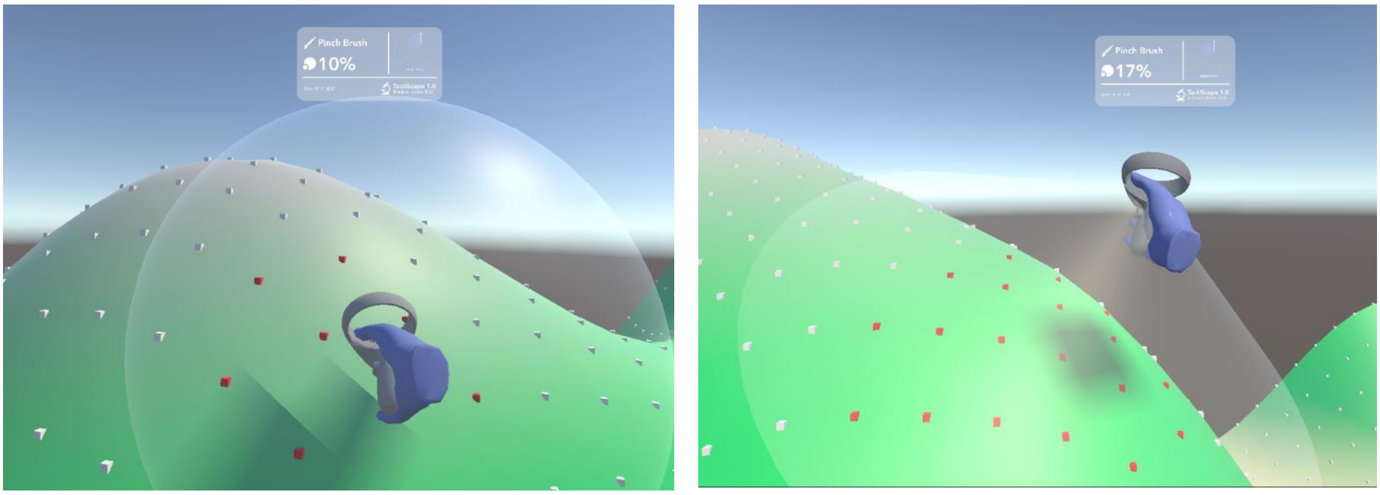
(Flashlight Brush)

$$\left(\left(\sqrt[2]{(x_{rh} - x_v)^2 + \ (y_{rh} - y_v)^2 + \ (z_{rh} - z_v)^2}\right) \leq d\right)$$
$$\wedge \ \left((\,direction_{rh \to v} \geq -0.25) \wedge (\,direction_{rh \to v} \leq \ 0.25)\right)$$

(Stamp Brush)



Figures 4.8 and 4.9: Flashlight brush shape; and Stamp Brush Shape



Figures 4.10 and 4.11: Flashlight brush shape; and Stamp Brush - in action indicating their size, shape, and which vertices will be manipulated.

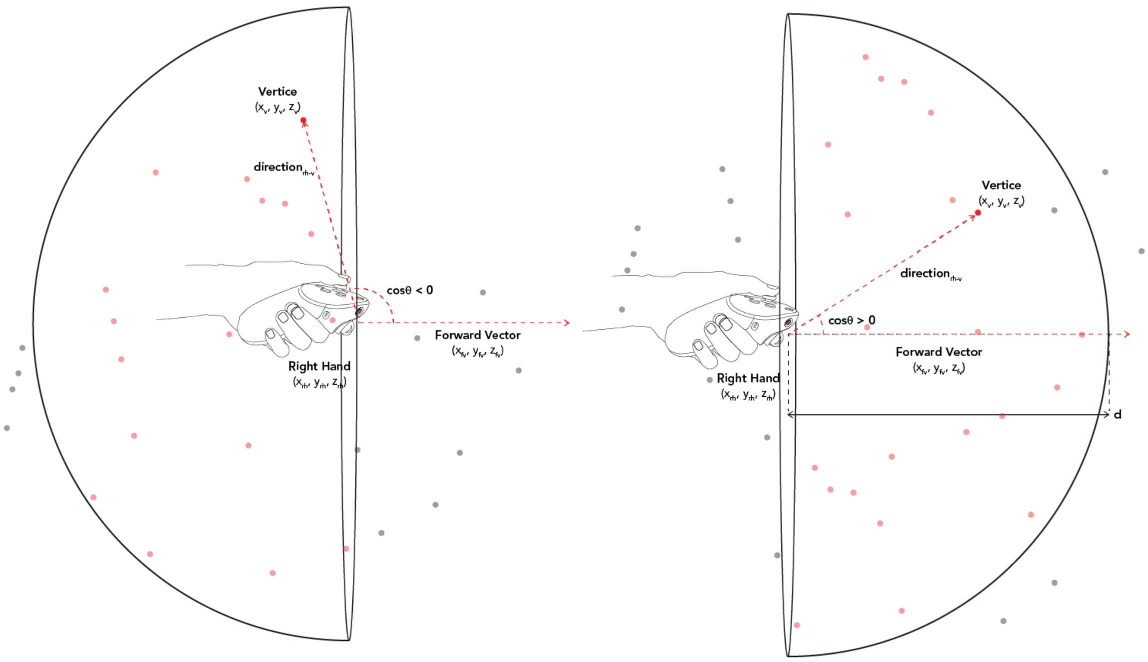
Vertical Hemisphere brushes (Figures 4.12 and 4.13) use the same formula but with a margin of 1 from target dot products of 1 (facing the same direction as the vertex) and -1 (Facing the complete opposite direction as the vertex. They are thus given by the conditions:

$$\left(\left(\sqrt[2]{(x_{rh}-x_v)^2+\,(y_{rh}-y_v)^2+\,(z_{rh}-z_v)^2}\right)\leq d\right)\ \wedge\ \left(direction_{rh-v}\geq 0\right)$$

(Forward Hemisphere)

$$\left(\left(\sqrt[2]{(x_{rh}-x_v)^2+\,(y_{rh}-y_v)^2+\,(z_{rh}-z_v)^2}\right)\leq d\right)\ \wedge\ \left(direction_{rh-v}\leq 0\right)$$

(Backward Hemisphere)



Figures 4.12 and 4.13: The Frontwards and Backwards Hemisphere brush shapes.

(b) Brush Impact

The behaviour of the vertices selected using the described brush shapes is influenced by the Brush Impact. The tool uses three main types of brush impact algorithms to dictate the displacement of vertices in the area of influence. This movement is affected by the following factors:

1. **Brush Strength**: The brush strength is a float value between 0 and 1 used as a multiplier.
2. **Direction:** The direction is given by a 3D vector that is calculated as a weighted sum of the desired direction of movement and the velocity of the user's hand.
3. **Velocity Magnitude:** The user's hand velocity magnitude is directly proportional to the magnitude of the displacement.
4. **Distance from the Vertex:** The hand's distance from the vertex is inversely proportional to the vertex's movement to create a feather effect characteristic of clay sculpting. A vertex closer to the hand will be pushed further away.
5. **Device Framerate:** The time taken by the device to render subsequent frames is inversely proportional to each vertex's movement. This is a common technique and is used to ensure smooth movements regardless of the device.

The following are the Brush Impacts created using these factors:

Free Form Velocity Brush

The first brush allows for mesh deformation in a way that closely resembles physical sculpting. The vertices, and hence the mesh material, can be pushed or pulled in any direction. This manipulation is tied to the direction and magnitude of the incident force by multiplying the velocity of the right hand as a 3D vector. This allows for weaker and stronger pushes, grabbing and pulling, and precise manipulation. The movement of the vertices is given by:

$$\Delta\, VertexPosition(x_v, y_v, z_v) \;\propto\; \frac{Velocity(x_{rhv}, y_{rhv}, z_{rhv}) \times brushStrength}{\left(\sqrt[2]{(x_{rh}-x_v)^2 + (y_{rh}-y_v)^2 + (z_{rh}-z_v)^2}\right) \times timeBetweenFrames}$$

$$where \quad Velocity(x_{rhv}, y_{rhv}, z_{rhv}) \;=\; \Delta\, HandPosition(x_{rh}, y_{rh}, z_{rh})$$

Or in code:

```
Vector3 vertex += Vector3 HandVelocity * float brushStrength * float Time.deltaTime / float distance;
```

(The Vector3 and Time objects are instances of native Unity static classes)

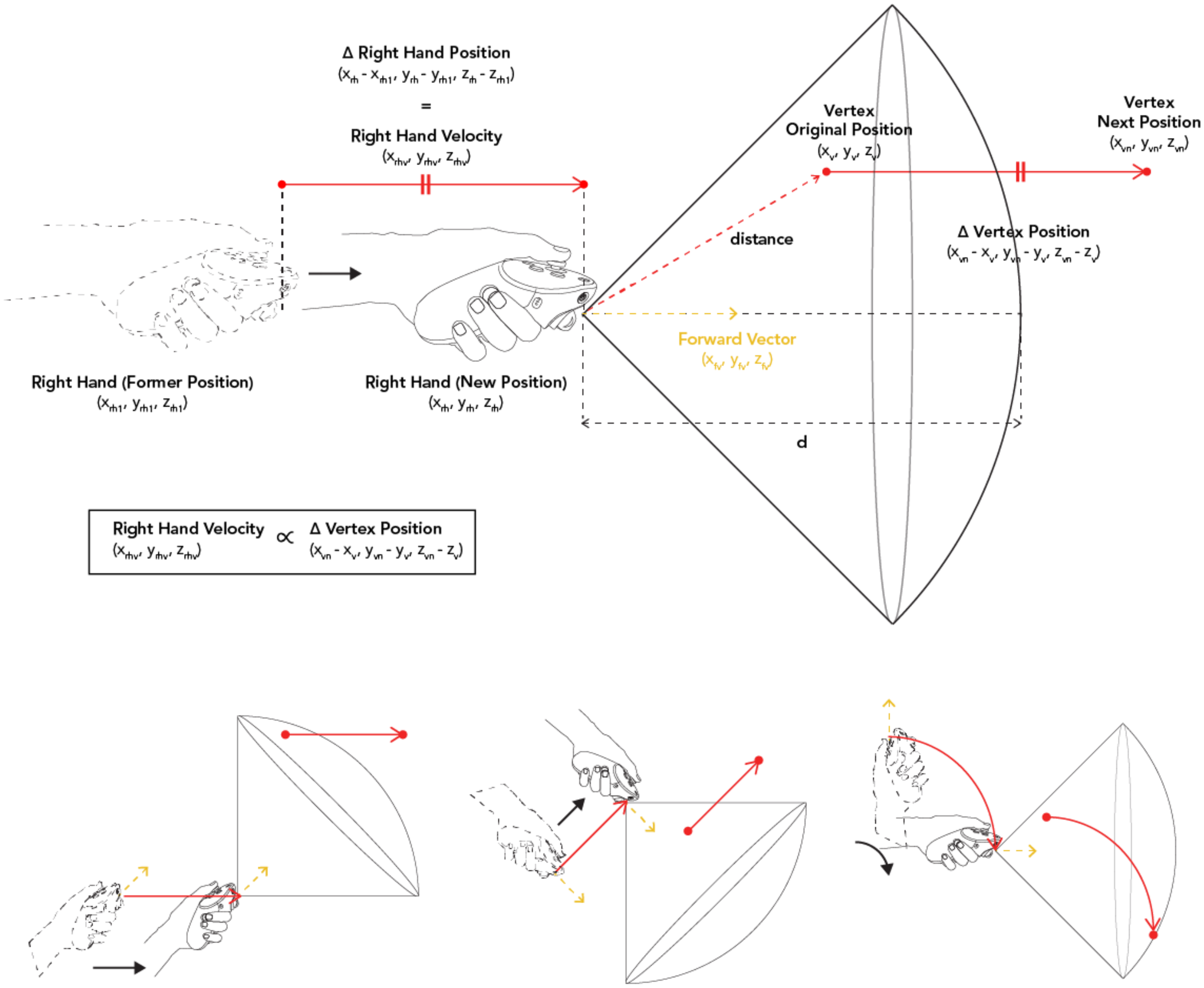
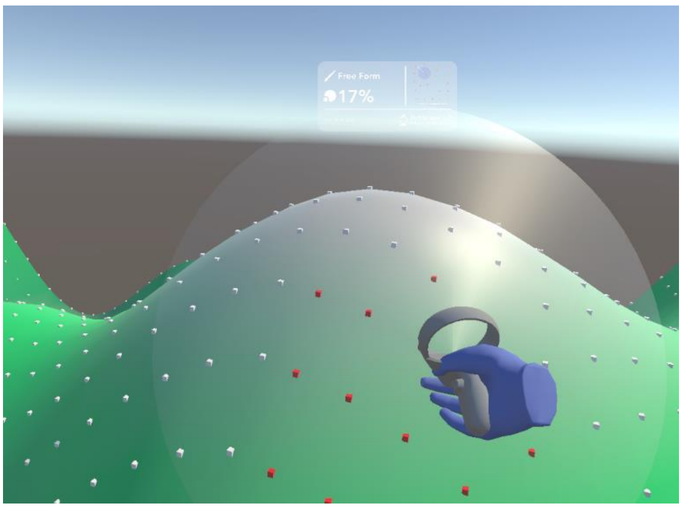
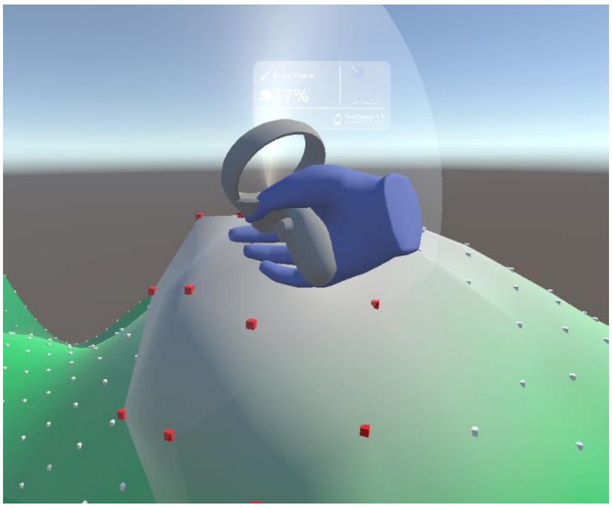


Figure 4.14: The Free Form Velocity Brush mathematics; and example use cases.





Figures 4.15 and 4.16: The Free Form Displacement in action, grabbing terrain and pulling it outwards along multiple axes.

Push Down/Pull Up Brush

A variant of the Free Form Velocity Brush is the Push Down/Pull Up brushes that allow for motion only in the y-axis. This allows the user more practical manipulation to replicate the act of patting down on clay vertically. The brush pulls up or pushes down based on the direction of the hand's velocity. The movement is hence given by:

$$\Delta\, VertexPosition(x_v, y_v, z_v) \;\propto\; \frac{Vector(0,1,0) \times Velocity(\, x_{rhv}, y_{rhv}, z_{rhv}) \times brushStrength}{\left(\sqrt[2]{(x_{rh}-x_v)^2 + (y_{rh}-y_v)^2 + (z_{rh}-z_v)^2}\right) \times timeBetweenFrames}$$

Or in code:

```
Vector3 vertex += float HandVelocity.y * float brushStrength * float Time.deltaTime / float distance;
```

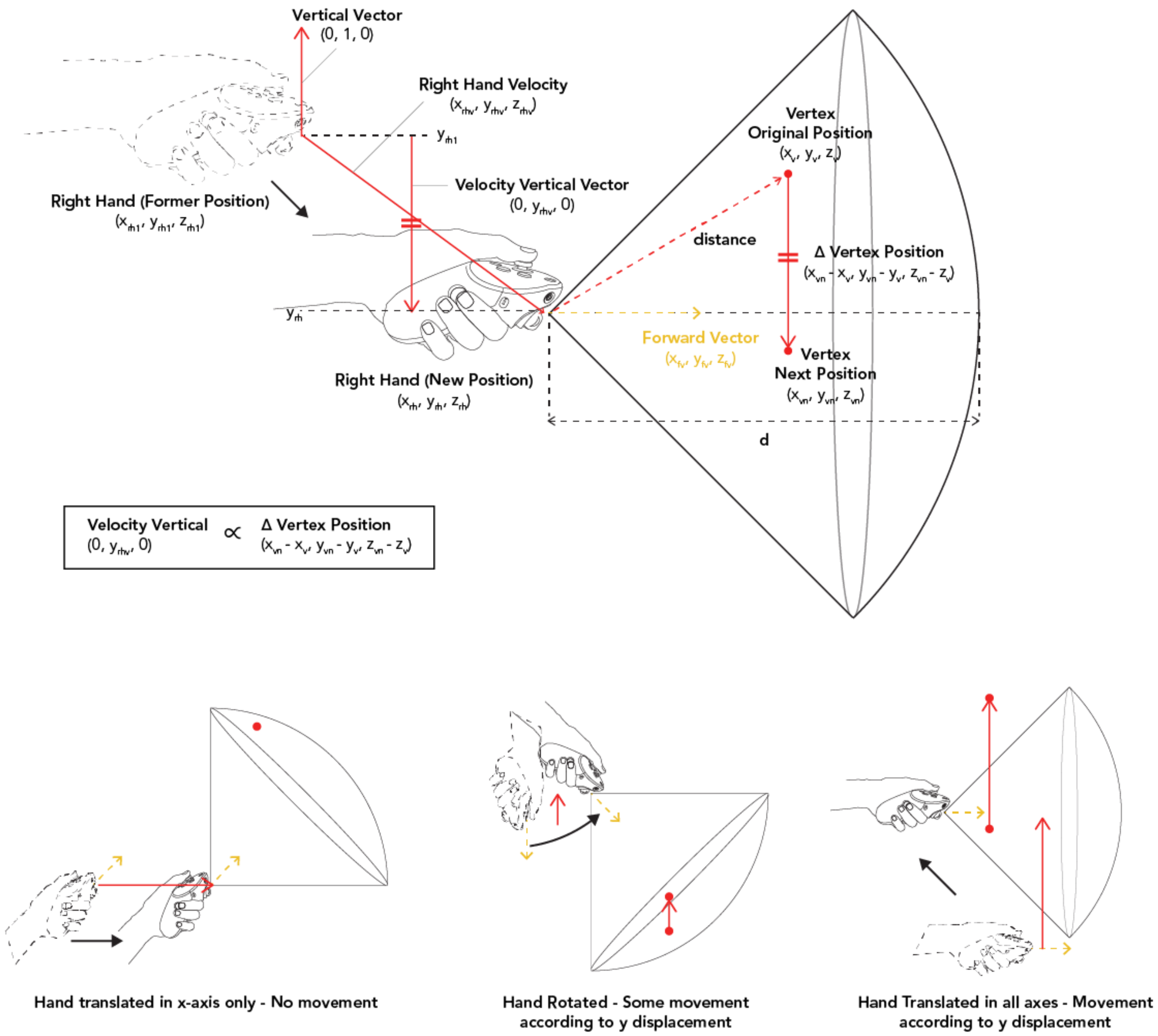
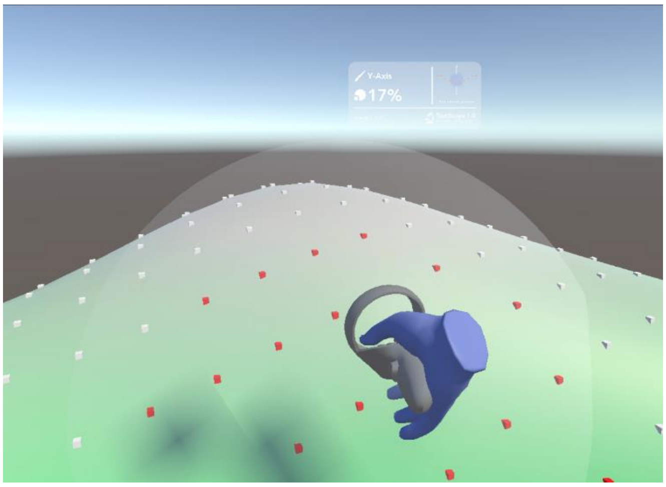
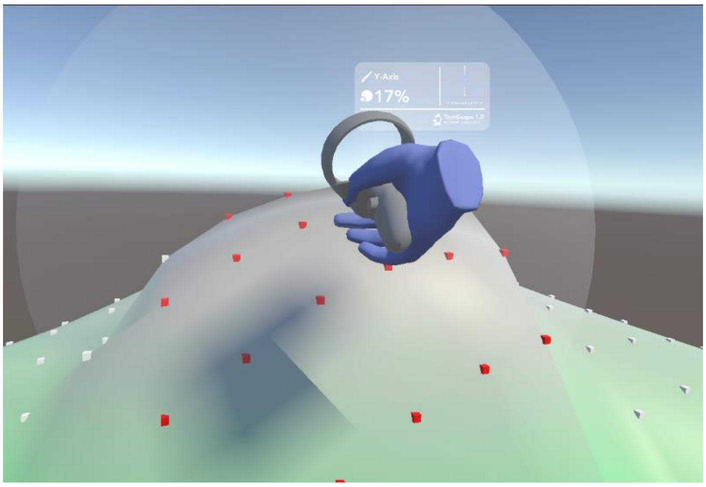


Figure 4.17: The Pull Up/Push Down Brush mathematics; and example use cases.





Figures 4.18 and 4.19: The Push Down and Pull Up Brush in action, pulling a peak further up.

Flatten Brush

The third brush uses a unique system for implementation and serves a niche purpose. While it is possible to flatten surfaces using the Push Down and Pull Up brushes, the functionality doesn't account for unevenness in the surface. The flatten brush averages the location of all the vertices in the area of influence and transforms them towards the average height.

The average height is calculated using the y-components of the vertices in question:

$$Average\ Height = y_{avg} = \frac{(y_1 + y_2 + y_3 + y_4 + y_5 \ldots + y_n)}{n}$$

It is then used to transform every vertex towards the average height given by:

$$\Delta\, VertexPosition(x_v, y_v, z_v) \propto \frac{Vector(0{,}1{,}0)\,(\,y_{avg} - y_v) \times brushStrength}{\left(\sqrt[2]{(x_{rh}-x_v)^2 + (y_{rh}-y_v)^2 + (z_{rh}-z_v)^2}\right) \times timeBetweenFrames}$$

Or in code:

```
if (Vector3 vertices[i] is within brush range) new List<Vector3> relevantVertices.Add(vertices[i])

Vector3 AveragePos = (Vector3 relevantVertices[0] + Vector3 relevantVertices[1] … Vector3
relevantVertices[relevantVertices.Count()].y) / float relevantVertices.Count()

Vector3 vertex += (Vector3 AveragePos – Vector3 vertex) * new Vector3 (0,1,0) * float brushStrength
* float Time.deltaTime / float distance;
```

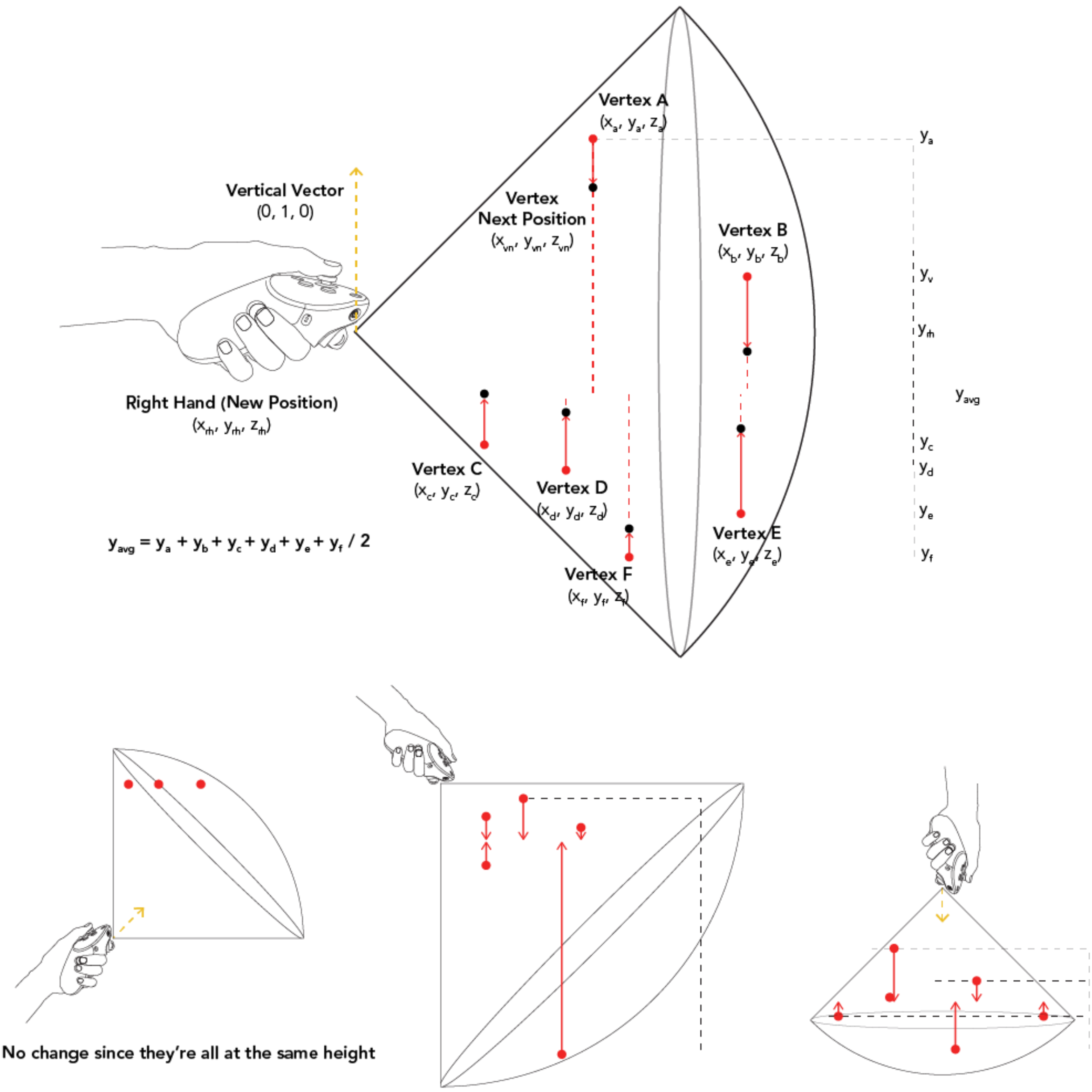
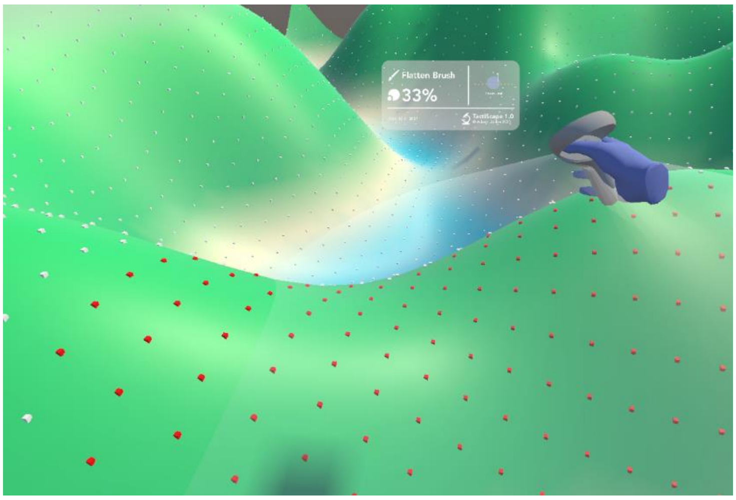
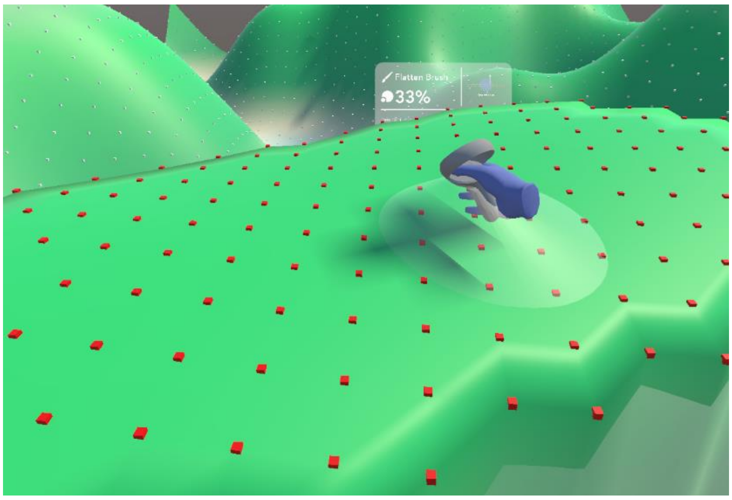


Figure 4.20: The Average Flatten Brush mathematics; and example use cases.





Figures 4.21 and 4.22: The result of using the average Flatten brush – Before and After

Radial Pinch and Push Brush

The final two brushes use functionality independent of the hand's velocity. The Pinch Brush is replicates pinching one's fingers to create a peak in clay. Vertices are transformed radially towards the hand. In direct contrast, the push brush of the type pushes vertices radially away from the hand to create a crater resembling pushing one's fist into clay to displace it.
The movement is given by:

$$\Delta\, VertexPosition(x_v, y_v, z_v) \;\propto\; \frac{(Hand(\, x_{rh}, y_{rh}, z_{rh}) - VertexPosition(x_v, y_v, z_v)) \times brushStrength}{\left(\sqrt[2]{(x_{rh}-x_v)^2 + (y_{rh}-y_v)^2 + (z_{rh}-z_v)^2}\right) \times timeBetweenFrames}$$

Or in code:

```
Vector3 vertex += (Vector3 HandPosition – Vector3 Vertex) * float brushStrength * float
Time.deltaTime / float distance;
```

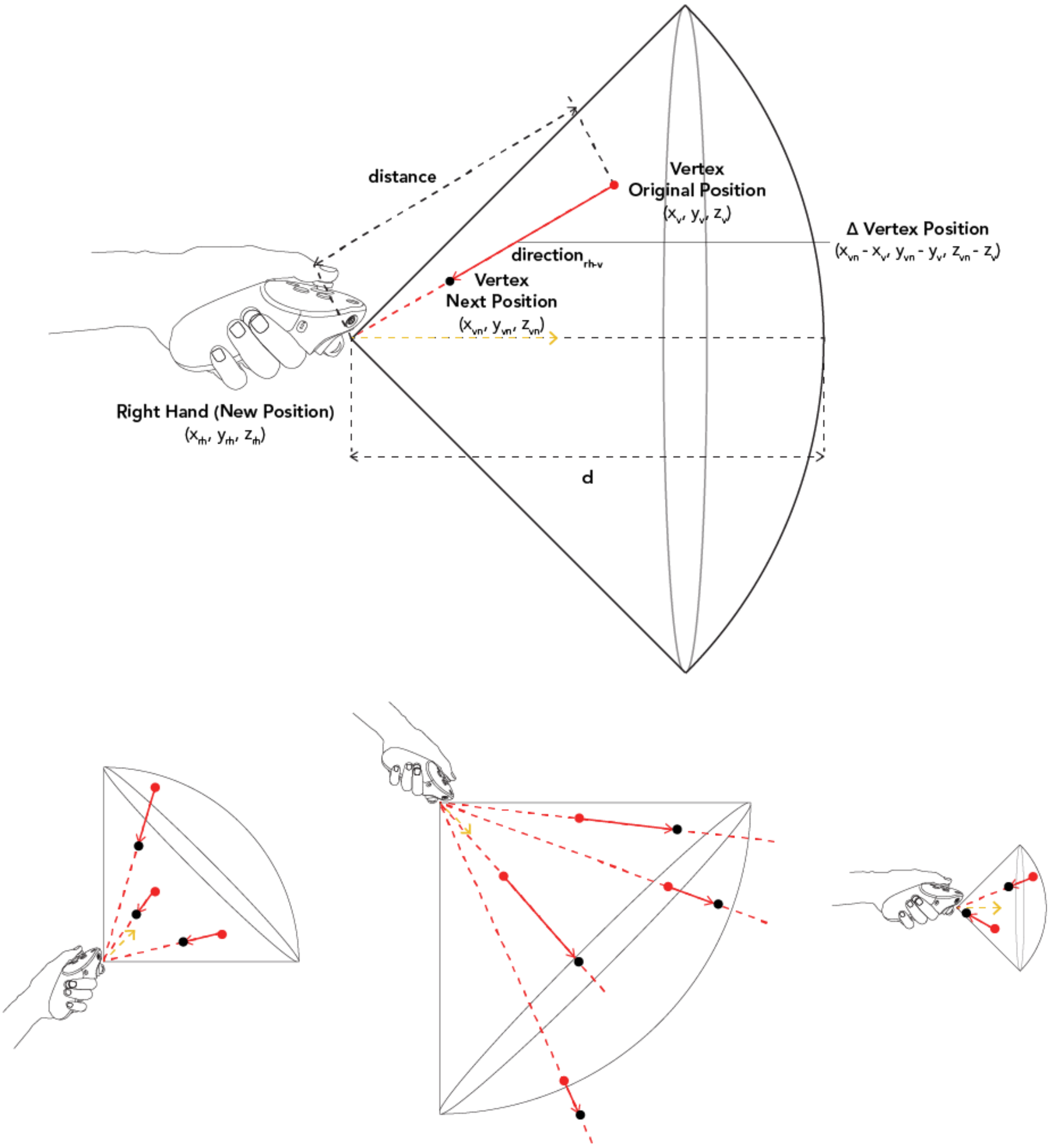
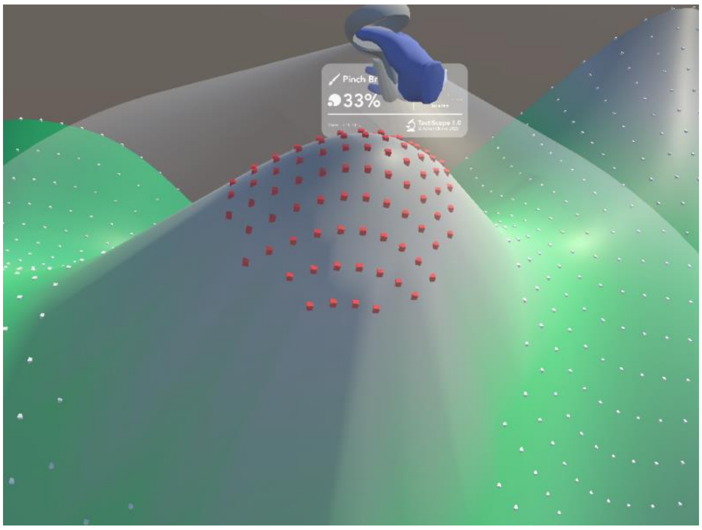
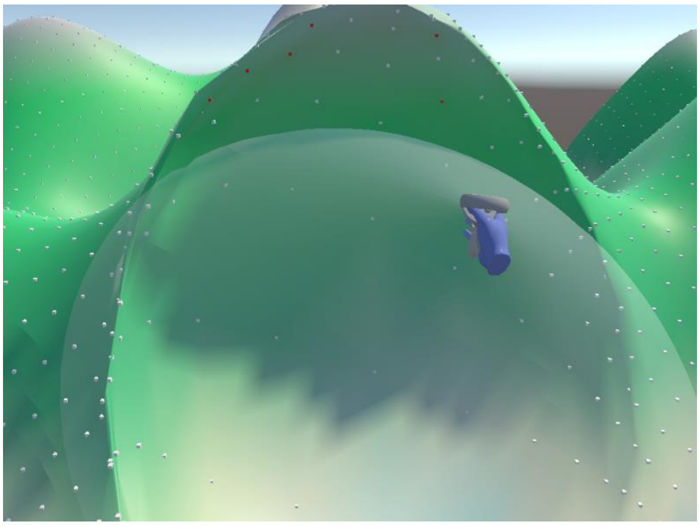


Figure 4.23: The Radial Pinch and Push Brush mathematics; and example use cases.





Figures 4.24 and 4.25: The Radial Pinch and Push Brush – Results of using them in TactiScape VR

4.2.5 Feedback Systems

(a) Haptic

TactiScape VR utilises a system to provide haptic feedback when the user's hand interacts with the landscape. The haptic feedback is provided with two different settings – when the user is merely touching or running their hand across the material clay, and when they're actively manipulating the mesh. These provide haptic feedback of varying amplitudes (intensity) and duration. While direct manipulation provides feedback proportional to the user's hand velocity, simply touching the mesh provides feedback based only on the velocity being non-zero (with a margin of error for the natural shake of the hand). These conditions are given by:

$$\left(\left(\sqrt[2]{(x_{rh}-x_v)^2+\ (y_{rh}-y_v)^2+\ (z_{rh}-z_v)^2}\right) \leq d\right) \wedge \ \left((x_{rhv} \geq margin) \vee (y_{rhv} \geq margin) \vee (z_{rhv} \geq margin)\right)$$

(Interacting and Touching)

$$amplitude\ of\ haptic\ feedback\ \propto\ \Delta\, VertexPosition(x_{rh}, y_{rh}, z_{rh})$$

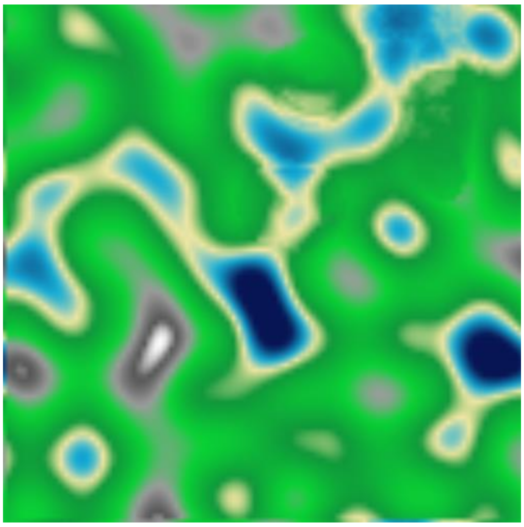
(For Interacting Only)

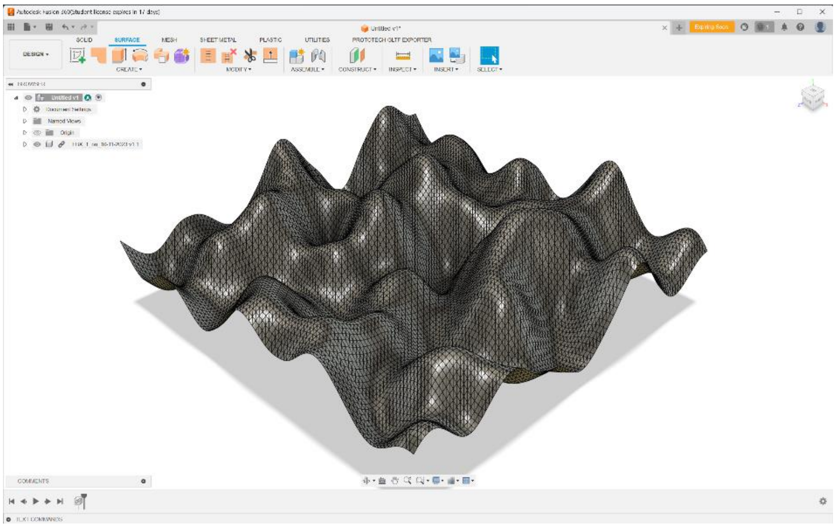
(b) Auditory

Audio feedback is used to signal touching the virtual clay. It is presented as 3-4 sounds recorded using folie techniques and is triggered the same way haptic feedback is.

(c) Visual: Area of Influence
Visual indications of the area of influence are of two types: (a) 'Selected' vertices turning red; and (b) Translucent mesh matching the Brush Shape. The latter is achieved through brush shaped meshes selectively enabling or disabling based on the selected brush. Figures 4.10 to 4.25 above shows this area of influence functionality in action.

4.2.6 Export
As with any other sculpting medium, exportability is a crucial functionality. TactiScape VR offers export into two formats for easy reproducibility and visualisation.





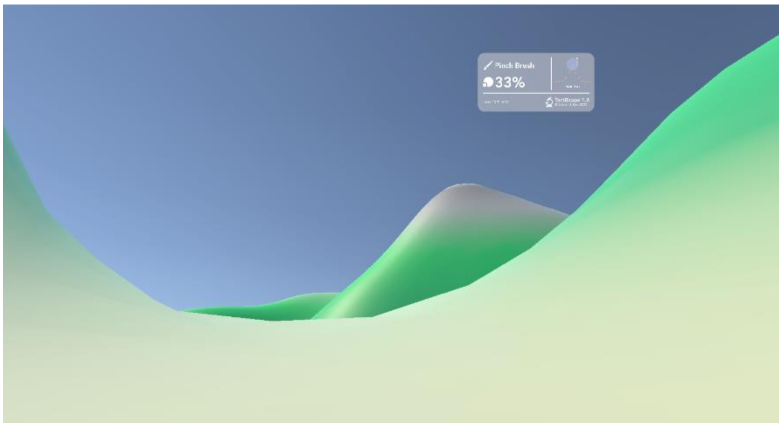
Figures 4.26 and 4.27: The PNG Depth Map; and the FBX opened up in Fusion 360 for further edits.

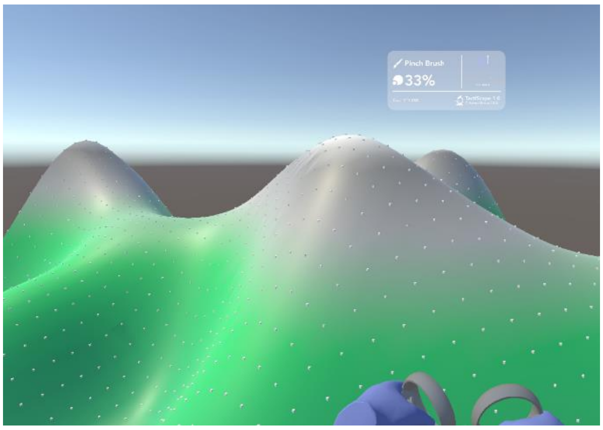
(a) Exporting as an FBX
Using Unity's native FBX Exporter, TactiScape exports the entire scene as a FBX reproducible in any supported CAD programme for further editing, rendering, and deployment.

(b) Exporting as a PNG Depth Map
Alternatively, the tool also allows for the export of the created landscape as a PNG Depth Map for quick if sometimes inaccurate visualisation. To achieve this, the script captures the y of each vertex and shades the corresponding pixel on a 2D canvas according to the gradient or in greyscale. This comes at the cost of being unable to capture (a) movements in vertices along x and z axes; and (b) any overhangs.

4.2.7 Landscape Scaling

In order to visualise the landscape at real world scale, a crucial affordance of VR, the tool allows users to scale the created mesh up and down at will. Scaled up meshes allow the user to traverse the landscape on the same scale as the player permitting easy tangible visualisation and the ability to identify any unforced errors at scale.

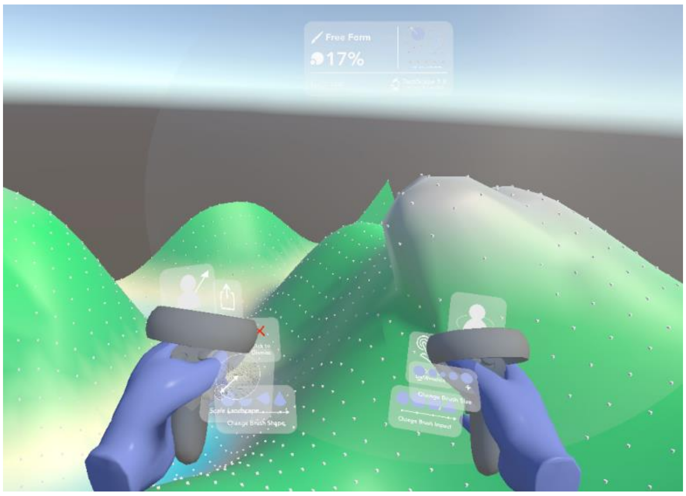


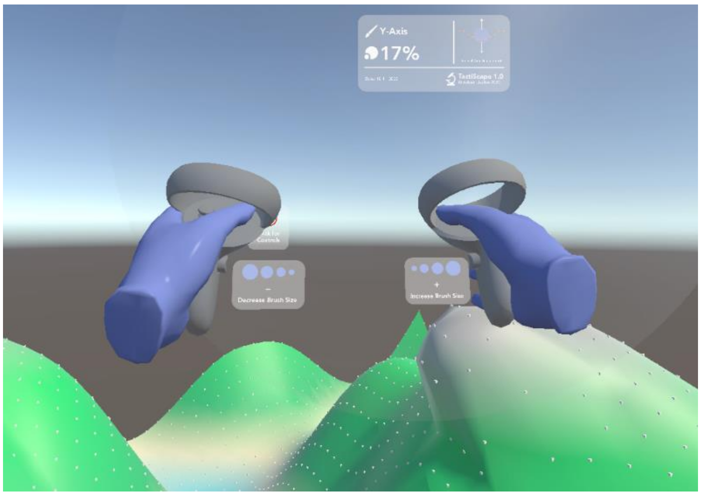


Figures 4.28 and 4.291: The landscape scaled to view at real world scale to aid visualisation.

4.2.8 Graphical User Interface

The GUI is made using 2D sprites that indicate the function of all the buttons and an overlay panel that shows the active brush and brush size.





Figures 4.30 and 4.31: The 2D Sprites showing Control Scheme and the Overlay Panel

5. Testing and Evaluation

5.1 Framework of Testing

5.1.1 The Experiment

The experiment was conducted in 4 parts and took between 15-45 minutes each:

(a) **Introduction** – The participants were given a brief introduction on the research and its aims and encouraged to treat the session as a means of constructive feedback. They were asked questions to establish their positionality including their background and proficiency.

(b) **Testing TactiScape vs Blender** – The participants were asked to describe a landscape they wanted to build based on the limitations presented. They were asked to create it in Blender using its sculpt mode in an environment that matched the one used in TactiScape. No introduction to its tools were given and basic proficiency in using a mouse and keyboard was assumed. Following this, participants were asked to create the same/similar landscape in TactiScape. Participants were asked to perform both these tasks in around 5 minutes.

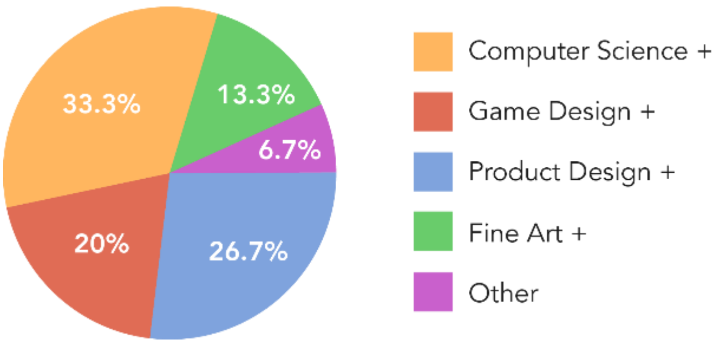
(c) **Qualitative Discussion** – Next, participants were quizzed on their experience in both sculpting environments to gather qualitative observations on both. The series of questions can be found in Appendix C.

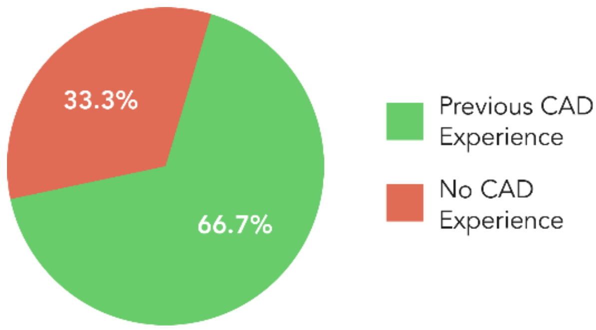
(d) **Quantitative Feedback** – Finally, participants were asked to fill a short feedback form where the two platforms were compared quantitatively along the highlighted metrics on a scale of 1-10. You can find a link to the online form in Appendix C.

Please refer to Appendix A for images from TactiScape testing.

5.1.2 The Participants

The total number of participants who took a part in the testing and evaluation of the tool was 15. While the number was limited, they were chosen for the various positionalities needed for a broad evaluation. These included 5 participants with no previous experience in 3D Sculpting, free from any bias towards either, 5 others with limited proficiency and 5 with high proficiency and previous experience in other CAD tools to gather vocational insights. The participants also represented a broad range of professional and academic backgrounds bound to utilise 3D Sculpting software in their own design process.





Figures 5.1 and 5.2: Composition of the participants

5.2 Results

5.2.1 Intuitiveness (Ease of Understanding and Using the Various Functions)

Quantitative Analysis

- Participants were asked to give both Blender and TactiScape scores from 1-10 on how (a) Simple and Natural it felt; (b) Easy it was to learn; and (c) Easy it was to understand and find the right tools.
- Blender (6.66) and TactiScape (6.33) showed a marginal difference in (a). This could be partially explained by the cohort's overall familiarity with traditional CAD software as people with no previous CAD experience rated TactiScape higher than Blender by an average of 2.6 points. (See Figure 5.3)
- Similarly, the overall cohort indicated to finding the brushes and various functionalities easier in Blender although participants with no CAD experience rated TactiScape higher with an average of 1.6 points. (See Figure 5.5)
- TactiScape was rated higher in ease of learning with the gap being significantly larger for non-CAD users. (See Figure 5.4)

Qualitative: Understanding the System

- When quizzed about how they think the tool works, all the participants supplied answers that covered important aspects of the experience to varying degrees.
- 20% of the participants were able to understand the technical aspect of it in addition to the user-facing aspects.
- One participant with no background in CAD or Computer Graphics supplied the elegantly put answer: *"It's a grid of points, and I can do whatever [I want] to the points that turn red."*

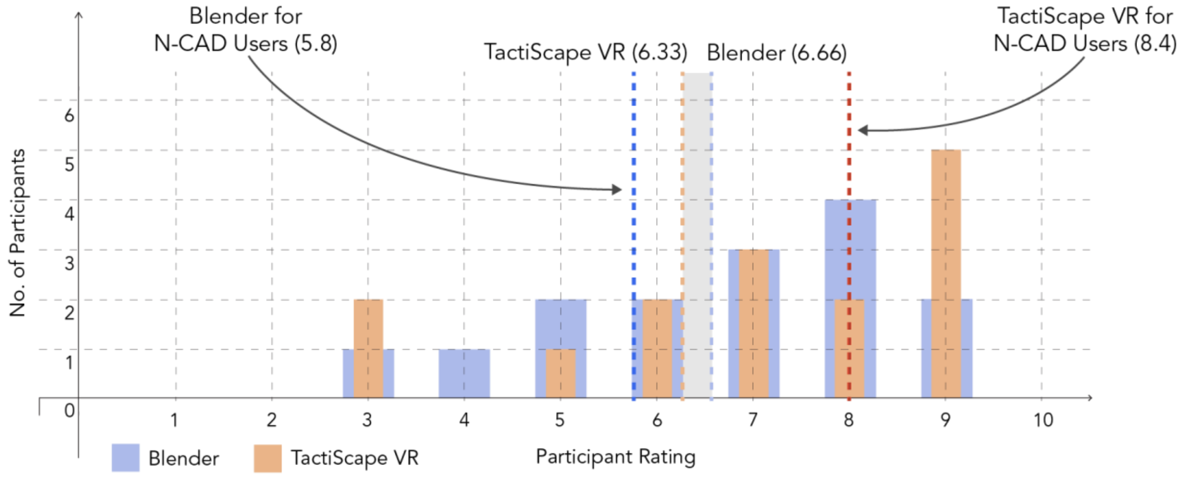


Figure 5.3: Participant answers to how 'simple and natural' each of the tools felt.[3]

[3] TactiScape VR and Blender are compared using different colours. The lines represent the average of the participant ratings, and the grey area visualises the difference between the averages. **N-CAD stands for No previous CAD experience.**

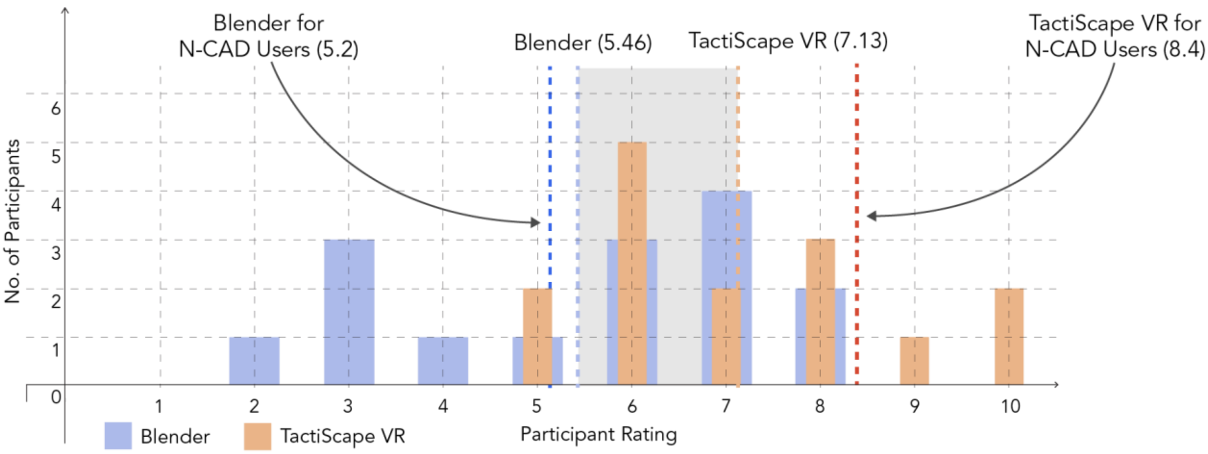


Figure 5.4: Participant answers to how easy it was to learn both tools.

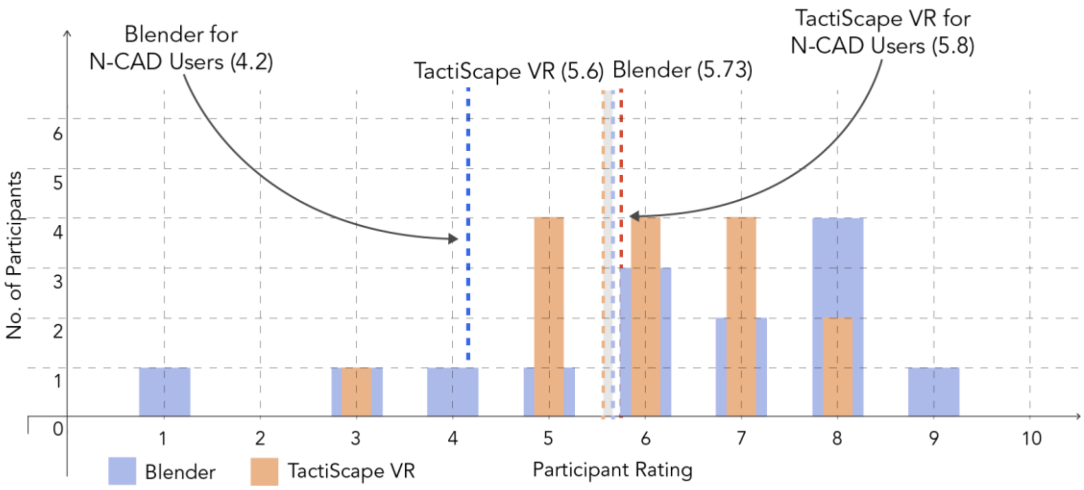


Figure 5.5: Participant answers to how easy it was to find and use all the functionalities of each tool.

Qualitative: Using the System

- Close to 50% of the participants struggled to understand the control scheme and UI with many admitting to taking a little time before they were comfortable with it. 40% of the participants explicitly asked the researcher for help finding the right advanced controls.
- Participants appreciated the interaction needed to manipulate the landscape. When quizzed about whether it was intuitive to them, 80% responded affirmatively with some adding concerns about it needing a little getting used to.
- Almost 90% of the participants spoke of the comfort of the interaction once they had crossed the initial learning curve, which would be anywhere between 0 and 5 minutes. They generally shared the sentiment put forth by one of the participants *"It takes a couple of minutes, but then it is very obvious."*
- 60% of the participants with previous CAD experience and 20% with no previous experience (totalling to around 50% of all participants) said they would prefer a 'panel menu' akin to existing CAD software.

- Participants with previous experience in CAD generally showed better understanding of aspects such as controlling the type of brush, brush size, and brush impact.
- Some participants also made the difference between the freedom TactiScape appears to offer as opposed to Blender. One participant (game designer and visual artist) said *"Blender feels like a design tool whereas this one [TactiScape] feels like an artist's tool. It is artistically very cool. [It] lets you do whatever the **** you want!"*
- Participants were also generally happy with the haptic feedback with 20% mentioning it before they were specifically asked about it. One participant noted *"The haptic feedback made it feel like you were "doing something in the real world, like, having a real impact on something."*
- Still other were more critical of the haptic feedback with 20% disagreeing with its inclusion in the software.

5.2.2 Ease of Visualisation and Navigation

Participants were then asked their thoughts on how the visualisation and their experience navigating the landscape.

- Participants with no CAD experience showed a strong appreciation for the way the landscape is visualised in TactiScape with 100% marking it better.
- Even among other participants, 60% marked it equal or better than Blender.
- One of the participants reasoned it as *"I like it because I know exactly where I am and how to go to another place [in/of the landscape]"* while another attributed its comfort to their familiarity with playing video games, saying *"Movement in any direction is very sorted in this, it's [the learning] very quick."*
- 40% of participants with CAD experience explicitly appreciated means to get more detailed visualisations of the landscape with one adding *"In Blender, I need to create 2-3 [simultaneous orthographic] views [to edit vertices properly]. But that is very easy in this tool because you can just travel to that point and see it in 3D."*
- Participants were also highly appreciative of the gradient shader used in TactiScape, maintaining that it helped them visualise the terrain far better.
- The top complaints were that TactiScape, unlike Blender, doesn't offer a bird's eye view that one can edit from.
- Furthermore, some participants with previous CAD experience complained about the lack of orthographic views.
- Participants also raised concerns about being unable to rotate the model itself and being restricted to a minimum distance away from what was under them.

- Participants were asked to rate the two tools on how easy it was to visualise and navigate through the landscape. TactiScape VR was deemed significantly better on average with participants with no CAD experience showing an average variance of a whole 4.6 points.

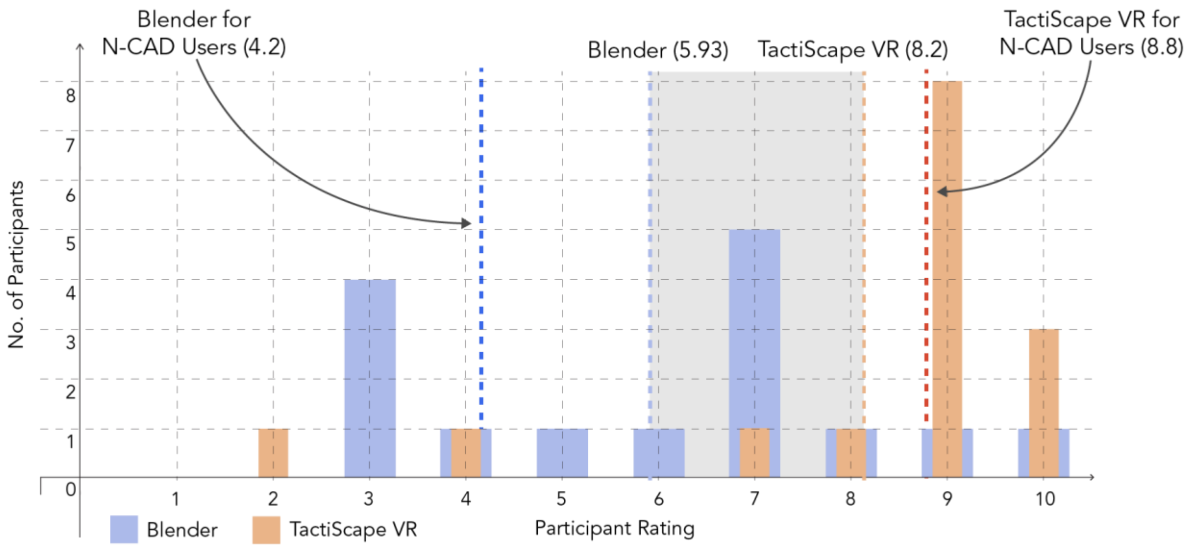
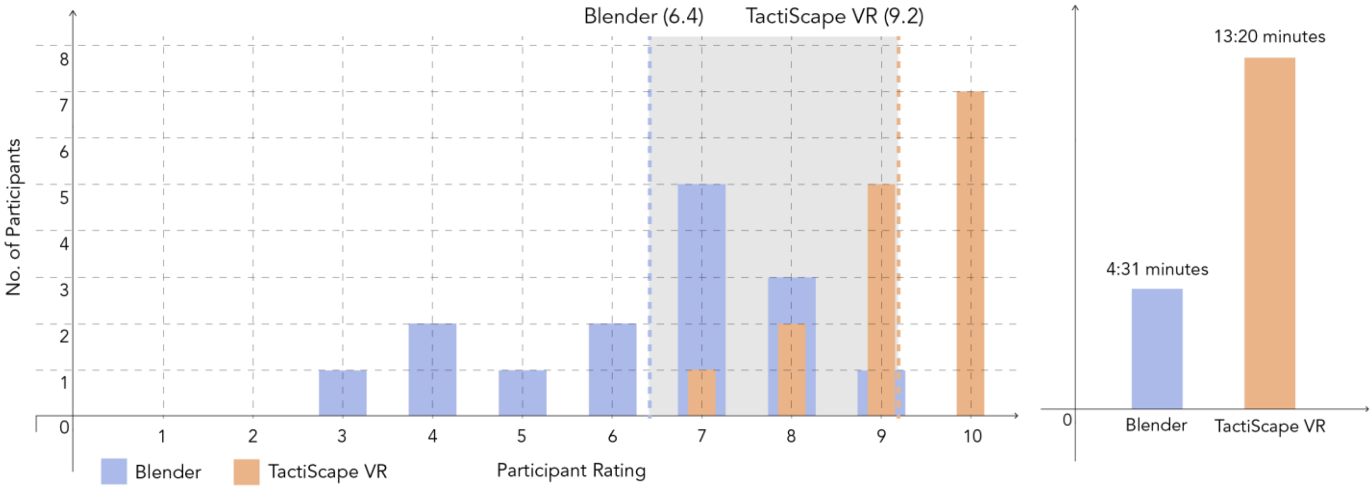


Figure 5.6: Participant answers to how easy it was to visualise and navigate through the landscape.

5.2.3 Overall Immersion and Engagement

- In tying into the previous question, participants were asked about how immersed they were in the world. Their engagement was also calculated based on the time they spent in each tool.
- Participants invariably reported higher engagement and immersion when they were using TactiScape as opposed to Blender.
- Participants attributed this to the feeling of corporeality in the world, with one noting *"I was a lot more immersed [in TactiScape] because I was right there."*
- One participant even enthusiastically went on at length about the world they had created in TactiScape and the fictional people that lived there.
- The participants were asked beforehand to use both applications for around 5 minutes with no hard upper or lower limits. 100% of the participants spent more time in TactiScape with a difference of between 3 and 20 minutes higher than Blender.
- One participant spent 27 minutes in TactiScape as opposed to 4 in Blender, noting *"I was having so much fun, I didn't know when 20 minutes had passed, seriously."*

- Participants were asked to rate how engaged and immersed they were in each tool on a scale of 1-10. The average engagement and immersion for TactiScape VR was rated 2.8 points higher than Blender with most participants tending towards a rating of 9 or 10.
- The average time difference between Blender (avg. 4 min 31 sec) and TactiScape (avg. 13 min 20 sec) was 8 minutes and 49 seconds.



Figures 5.7 and 5.8: Participant answers to how engaged and immersed they were while sculpting in each tool; and average time they spent sculpting in each tool.

5.2.4 Efficiency in Performing a Task

Participants were also asked to evaluate how successful they were in building what they had visualised before starting in each of the two tools.

- Most participants (around 85%) admitted to being only partially successful in building what they had visualised in TactiScape while 60% admitted to being partially unsuccessful in Blender.
- Interestingly, participants with no previous CAD experience claimed to be equally successful in both environments.
- Participants also tended towards the general sentiment that it was more efficient to make larger scale crude forms in TactiScape, but that the efficiency would be lost with more intricate details.
- When asked to attribute their failures in TactiScape, participants supplied a variety of reasons including (a) Lack of mesh density and smooth curves, (b) Lack of precision-based tools, (c) Discomfort with VR, and (d) A lack of tools they'd expect from a traditional CAD.

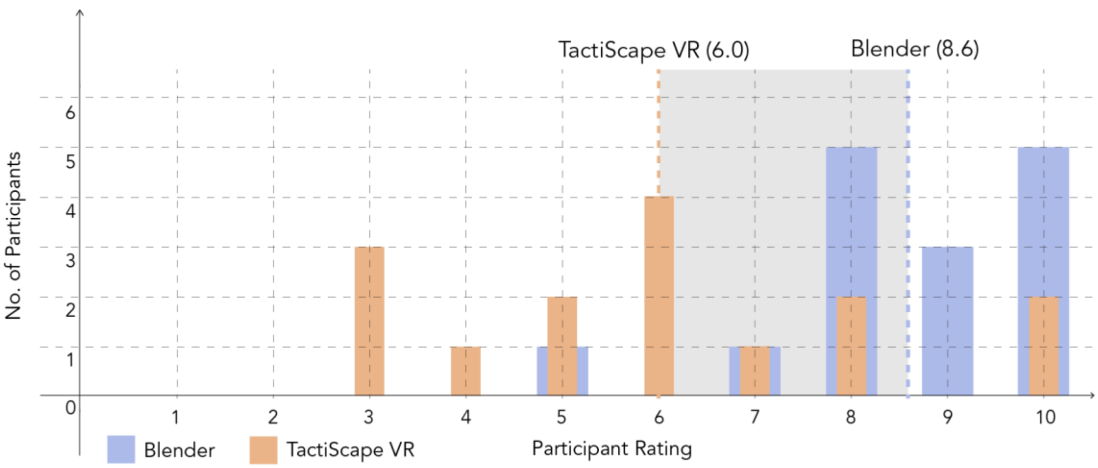


Figure 5.9: Participant answers to how efficient they felt while sculpting in each tool.

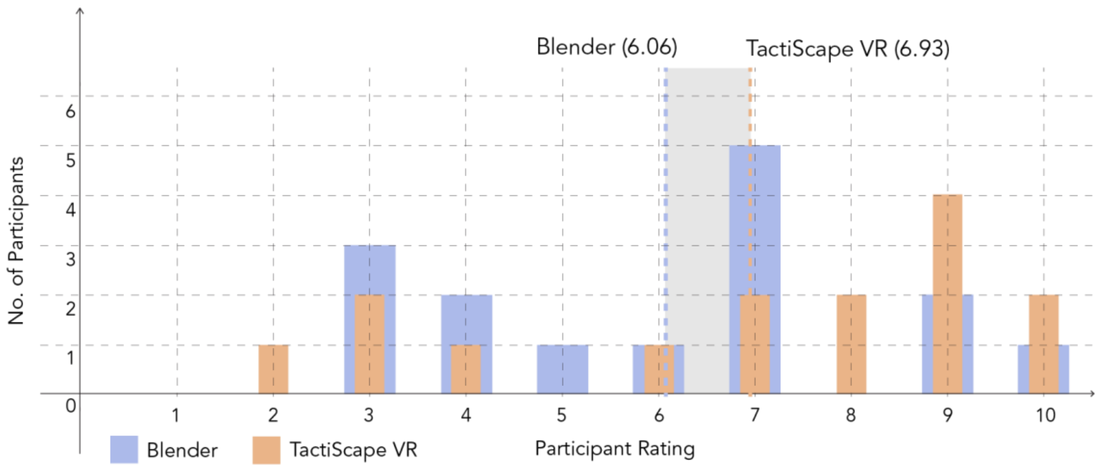
- When asked to rate their efficiency on a scale of 1-10, participants, on an average, marked Blender (8.6) far higher than TactiScape (6.0).

5.2.5 Impact on Creativity

While participants expressed mostly negative sentiments towards their efficiency, they often did so by adding that they felt more creative while using TactiScape.

- Opinions ranged from 'equally creative' to 'more creative' when participants were quizzed on TactiScape's impact.
- 100% of the participants with no previous CAD experience felt more creative using TactiScape. One participant noted *"It's [Blender] very specific about the things you can do, while this is more of a - here you go, do whatever you want to do."*
- One participant said *"I felt like a child playing with something new and I could have gone on. The physicality of it is fun, and when the colours started showing up it was fun."*, while another expressed *"[I] felt like God. Isn't that the definition of creativity? The movement of using your hands is so much better. Clicking with a mouse [in Blender] doesn't feel like you're doing anything."*
- Amongst participants with previous CAD experience, 40% claimed they felt more creative in TactiScape and 20% claimed they were equally creative.

- When asked to quantify the same, TactiScape (6.93) was rated higher than Blender (6.06).



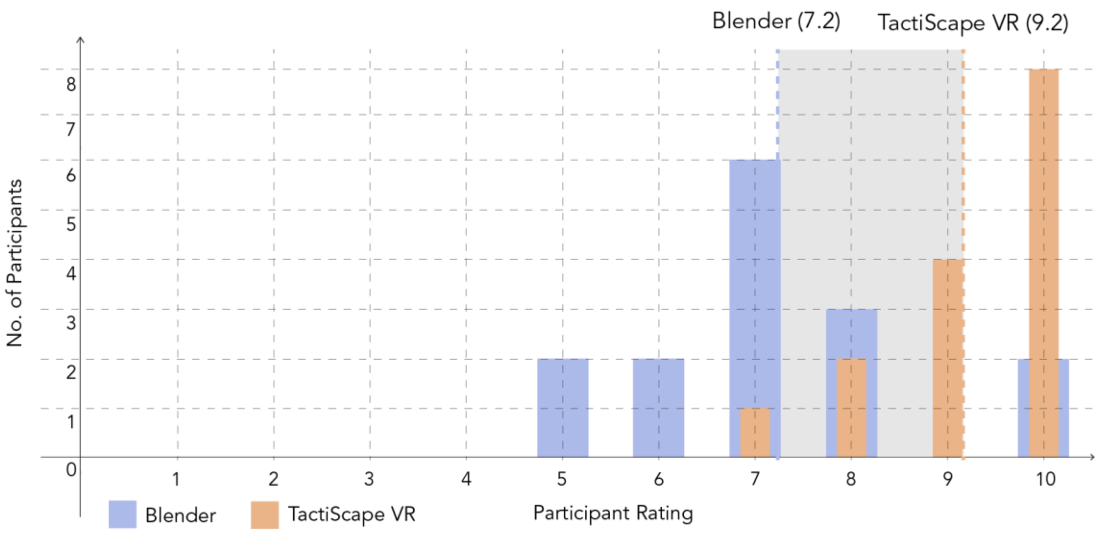
Figures 5.10: Participant answers to how creative they felt while sculpting in each tool.

5.2.6 Feelings and Enjoyability

Another common epilogue to questions about efficiency were participants claiming that although TactiScape was less efficient, it made the process of sculpting more fun.

- Around 90% of the participants pointed to TactiScape as the more fun tool to use to create landscapes.
- One participant noted *"It's more fun. It feels less like something you do for an office job. This would make it (my design process process) more playful."* Another said *"I felt powerful. I felt like I was exploring worlds, like I was God. Very weird, cool though."*
- Participants attribute this to factors such as the method of interaction where hands are moved to raise and lower landscapes, gradient shader, the haptic feedback, etc.
- When asked explicitly about any negative feelings they associated with the experience, participants spoke of their discomfort with VR, frustrations with not being able to move themselves quicker, and their dislike of continuous haptic feedback.

- Participants were also asked how fun it was to sculpt in each tool on a scale of 1-10. The results showed TactiScape with an average rating of 9.2 with 80% of participants rating it either 9 or 10.
- In contrast, Blender was rated at 7.2 on an average, a whole 2 points below TactiScape.



Figures 5.11: Participant answers to fun it was to sculpt in each tool.

5.2.7 Feasibility in Individual Design Processes

Towards the end of the interviews, participants were asked if they could see TactiScape fitting into their own design processes and what scope it would fit in if yes.

- 100% of the participants with no CAD experience and 40% with CAD experience (for a total of 60%) said they could find a place for TactiScape in their design process contingent on them needing to sculpt landscapes.
- Participants with CAD experience tended towards using TactiScape as a small part of a larger sculpting process which also involved their CAD tool of choice.
- They were in favour of using it as a tool for the initial broad scope conceptualisation and latter visualisation. One participant said *"[I] can imagine it as an add-on/plug-in to Blender as it is more fun to do the specific thing it does in this right now but [it] needs a lot more functionality to replace Blender."* Another noted *"For getting abstract landscape, it is fun and very easy, I can use it as a rough ideation part because I think its built for that."*
- Participants who found no use for TactiScape in their process noted it was due to a number of reasons including (a) Discomfort with VR, (b) Lack of fidelity in the current software, and (c) Inertia to switch between applications during their design process.

5.2.8 Ways to Make TactiScape Better

Finally, the participants were asked to explicit the things that would discourage them from using TactiScape and aspects of the experience that they dislike. These indicated areas to improve the tool in and were as follows:

- Multiple views as overlays/switchable.
- Ability to undo/redo.
- Brush shapes including quadrilateral, cylindrical, etc.
- Projection based brushes that allow for manipulation at a distance.
- Smoothening Brush and Higher Density Meshes
- Ability to add more than one mesh.
- Ability to rotate the landscape and the player controller.
- 'Feather' functionality as in Blender
- Better sound design

6. Discussion

The results of this research can be addressed on two separate fronts - its technical contributions in creating TactiScape VR and the results and implications of testing the created tool.

6.1 Technical Contributions

The research proposed a tool for simulating tactility in 3D Sculpting. In this regard, it was successful in conceptualising and creating TactiScape VR - which can be used to create semi-detailed landscapes using the corporeality used for physical sculpting.

TactiScape uses a VR Headset and controllers, and a novel system of mesh manipulation called Free Form Vertex Displacement. Users are allowed to move their hands freely to push, pull, pinch, and pat virtual clay. The tool suffers from some technical compromises including standalone performance and lacks essential tools such as undo, orthographic views, and mesh density and smoothness - that would allow it to compete with its non-tactile alternatives directly - owed largely to the scope and time permitted to its development.

Nevertheless, it presents a technical proof-of-concept for the development of a larger system with more fleshed out features using the same underlying methods – which could be used for 3D Sculpting across scopes.

6.2 Evaluation of Testing Results

In order to answer the research questions, we must critically examine the impact of TactiScape VR. Using simulated tactility, TactiScape attempts to create a better system of 3D sculpting with the chief metrics of measurement being intuitiveness, ease of visualisation and navigation, user engagement, user efficiency, overall feel and enjoyability, impact on creativity, and its feasibility in the design process. The results of the preliminary testing done with the product point to the following:

- The intuitiveness of simulated tactility vs. traditional non-tactile CAD is comparable, although people with no previous CAD experience attribute it higher.
- This intuitiveness is derailed by a lack of necessary features and the fidelity of TactiScape in its current form, but testers can speculate a full fleshed out system outperforming traditional means of CAD.
- TactiScape is easier to learn but is held back by a complex UI.
- Users with no expectations from CAD find it easier to use and navigate while users with previous CAD experience face some difficulty in adjusting.
- TactiScape presents a means of visualisation and navigation that is easier to understand and use but takes some getting used to owing to its UI and the cognitive load of VR on new users.
- The tool cannot be considered a more efficient means of 3D Sculpting as old and new CAD users alike tend towards computer-and-mouse 3D Sculptors for efficiency.
- TactiScape also suffers from a lack of tools built for CAD precision in this respect.
- TactiScape is measurably more immersive and inviting.
- Users report feeling more creative within its bounds than traditional CAD tools.
- Probably the biggest impact TactiScape has is in making the process of 3D Sculpting measurably more fun than traditional CAD.

- TactiScape is rated more immersive, and users spent significantly more time in the software. This could be owed to its inherent 'fun' factor or due to curiosity regarding a new environment. Participants with no experience in CAD also tend to spend longer using TactiScape.
- Users tend to find feasibility in TactiScape as a smaller part of their 3D Sculpting process and imagine using it for two main workflows – broad strokes ideation and conceptualisation; and final visualisation of the created model.

The above findings point to a tool that shows potential and ascertains that simulated tactility can go an extent in making the process of sculpting more intuitive for users new to CAD, and more immersive and fun for all kinds of users. It suffers from concerns of lack of efficiency and discomfort in VR than can potentially be overcome with a more developed sculpting tool. This would warrant further development along identified weaknesses, and the yet unfulfilled promise of better XR Headsets. Alternatively, other means of simulating such tactility such as holographic technology can also be explored based on the core findings.

6.3 Limitations

Limitations in the Tool Development

The research faced considerable limitations of time and team leading to the final scope of the project. The ideal version of the tool would involve full 3D sculpting with a plethora of tools, software optimised for performance, and better UX Design in order to examine the differences between the approaches to 3D Sculpting more critically and objectively.

Furthermore, arguments can be made regarding whether TactiScape is the perfect embodiment of pseudo-tactility and whether findings using TactiScape can be used to directly make a case for pseudo-tactility. This research acknowledges that it may not be but assumes it to be, in order to make critical comparisons. Future research can indicate more reliable comparisons.

Limitations in Testing

The research was also limited in both the number and variety of participants in its testing (pertaining to professional and academic backgrounds). Future testing and development would ideally include more participants and a balanced cohort of positionalities.

7. Conclusion and Future Scope

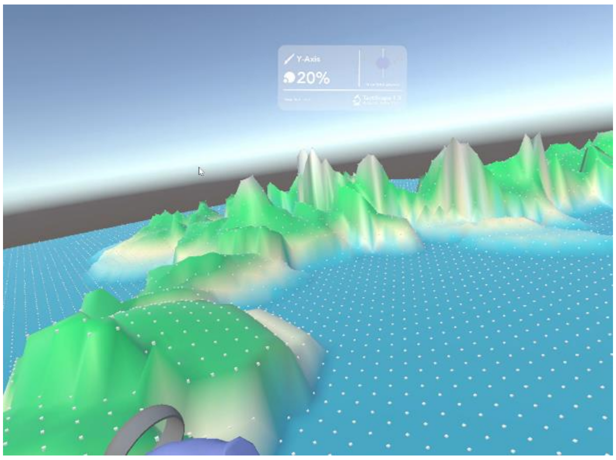
This research attempted to present a novel method (Free Form Vertex Displacement) and means (TactiScape VR) of 3D Sculpting that is built to simulate tactility in order to enhance the intuitiveness and the general experience. It asked the question 'Does simulating tactility in 3D Sculpting software offer any benefits towards the tangibility and overall experience?'

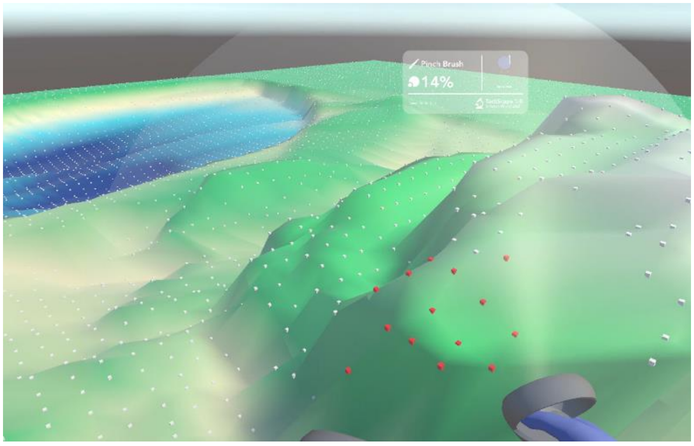
TactiScape VR laid the foundation of how pseudo-tactility can be purposed in 3D sculpting to create better tools. It evaluated the tool against traditional mouse-and-keyboard tools for 3D Sculpting and provided evidence that such tactility can make a tool easier to understand for new CAD users and more immersive and fun for all kinds of users. The method and the means can both be considered limited scope drafts whose user testing highlighted promise in its underlying approach to 3D Sculpting.

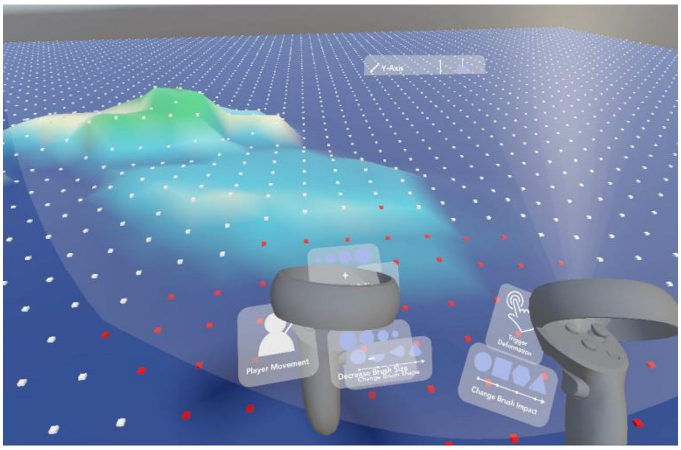
I hope to continue the conceptualisation and development of such pseudo tactile tools, both within and outside the scope of 3D sculpting, in the future based on the learnings from TactiScape VR. I hope to explore the use of simulated tactility, viewing it critically in theory and through practice, beyond the scope of this research.

The research points to a larger question that remains unanswered. Can we create fundamentally intuitive and fun interfaces for all our machines? If so, does simulating the affordances of the physical space we inhabit provide any tangible benefits to the pursuit? If the evidence presented in this research is anything to go by, it is certainly a start.

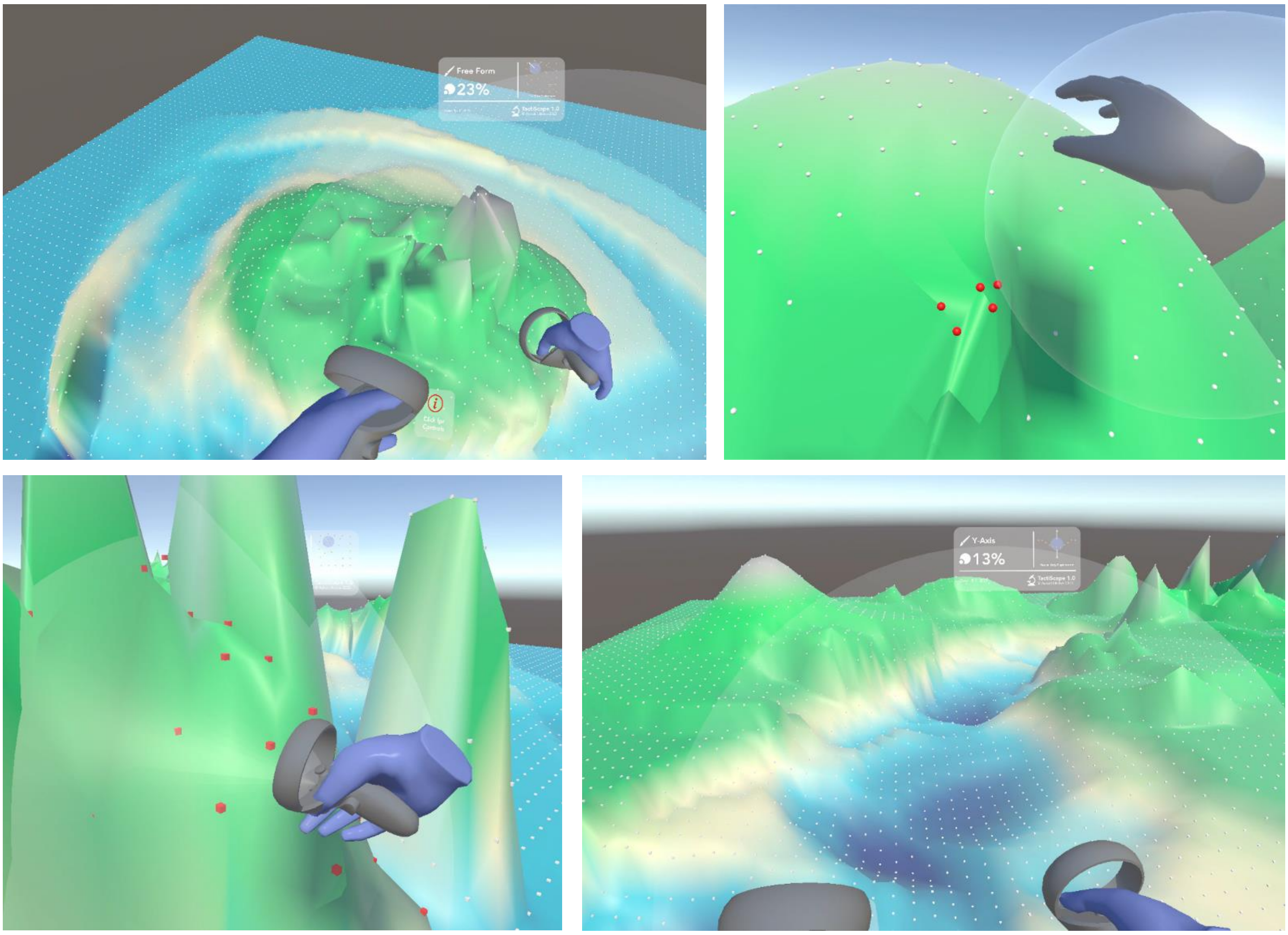
Appendix A - Images from TactiScape and Testing







(Continued on Next Page)



Images showing participants testing TactiScape VR and the landscapes they sculpted using it.

Appendix B - Links to External References

The following is a list of links relevant to the submission of this paper for MSc. Advanced Project:

- **GitHub Repository containing all the code and Readme:** https://github.com/AdvaitU/tactiscape-vr
- **YouTube Link for video presentation:** https://www.youtube.com/watch?v=_FOe3XUQhgI
- **Weekly Logs to show documentation of the process:** https://github.com/AdvaitU/tactiscape-vr/blob/main/Personal_Logs/Personal_Logs.md

Appendix C - Evaluation Questionnaire and Feedback Form

Quantitative Feedback

Online Google Form Document. Can be found at https://forms.gle/YxptzFRuovEXBiD3A

Qualitative Feedback – Interview Questions

The following questions were asked in general order:

1. Could you describe how this tool works in your own words?
2. How successful were you at creating what you'd imagined in both?
3. Would you use TactiSculpt in your design process? If yes, at what stage and how?
4. Did TactiSculpt feel easier to use? If yes, why do you think so?
5. Did you feel more creative while using TactiSculpt? If yes, why?
6. Was using TactiSculpt a more immersive process?
7. Was it easier to visualise what you were working on? If yes, which of the reasons?
8. Describe how using each tool made you feel about what you were working on/while you were working. Was it stressful? Did you feel any significant cognitive load? (For both tools)
9. Was there anything that would discourage you from using TactiSculpt or software of the like in the future? Please explain what all.

Bibliography

Abdelmohsen, S. and Do, E.Y.-L. (2007) 'TangiCAD: Tangible Interface for Manipulating Architectural 3D Models', in. *CAADRIA 2007: Digitization and Globalization*, Nanjing, China, p. XX–XX. Available at: https://doi.org/10.52842/conf.caadria.2007.x.x8o.

Adobe UK (2016) *Adobe Medium*, *Adobe Medium - Top 3D sculpting tools for virtual reality authoring | Medium by Adobe*. Available at: https://www.adobe.com/uk/products/medium.html (Accessed: 13 October 2023).

Alphabet Inc. (2015) *Tilt Brush by Google*, *Tilt Brush by Google*. Available at: https://www.tiltbrush.com/ (Accessed: 13 October 2023).

Apple Computer Inc. (1987) *Apple human interface guidelines : the Apple desktop interface*. Reading, Mass. : Addison-Wesley. Available at: http://archive.org/details/applehumaninterf00appl (Accessed: 12 September 2023).

Arkio ehf. (2022) *Arkio - Collaborative Spatial Design*. Available at: https://www.arkio.is/ (Accessed: 14 October 2023).

Autodesk UK (1998) *Autodesk Maya*, *Maya Software | Get Prices & Buy Maya 2024 in UK | Autodesk*. Available at: https://www.autodesk.co.uk/products/maya/overview (Accessed: 14 October 2023).

Autodesk UK (2023) *3DS Max Software*, *3ds Max Software | Get Prices & Buy 3ds Max | Autodesk UK<*. Available at: https://www.autodesk.co.uk/products/3ds-max/overview (Accessed: 14 October 2023).

Bahr, D. *et al.* (1996) 'FELIX: a volumetric 3D laser display', in M.H. Wu (ed.). *Electronic Imaging: Science & Technology*, San Jose, CA, pp. 265–273. Available at: https://doi.org/10.1117/12.237011.

Billinghurst, M. and Kato, H. (2002) 'Collaborative augmented reality', *Communications of the ACM*, 45(7), pp. 64–70. Available at: https://doi.org/10.1145/514236.514265.

Blender Foundation (2002) 'Blender - 3D Creation Software', *blender.org - - Home of the Blender project - Free and Open 3D Creation Software*, 13 October. Available at: https://www.blender.org/ (Accessed: 14 October 2023).

Butterworth, J. *et al.* (1992) '3DM: A Three Dimensional Modeler Using a Head-Mounted Display', in *Proceedings of the 1992 symposium on Interactive 3D graphics - SI3D '92. the 1992 symposium*, Cambridge, Massachusetts, United States: ACM Press, pp. 135–138. Available at: https://doi.org/10.1145/147156.147182.

Buxton, B. (2007) *Multi-Touch Systems that I Have Known and Loved*, *Multi-Touch Systems that I Have Known and Loved*. Available at: https://www.billbuxton.com/multitouchOverview.html (Accessed: 14 October 2023).

Daneshmand, M. *et al.* (2018) '3D Scanning: A Comprehensive Survey'. arXiv. Available at: http://arxiv.org/abs/1801.08863 (Accessed: 14 October 2023).

Dani, T.H., Chu, C.-C.P. and Gadh, R. (1997) 'COVIRDS: Shape Modeling in a Virtual Reality Environment', in *IDETC-CIE*. *ASME 1997 Design Engineering Technical Conferences*, Sacramento, California, USA: American Society of Mechanical Engineers. Available at: https://doi.org/10.1115/DETC97/CIE-4302.

Donelan, J. (2010) 'The Actuality Story', *Information Display*, 26(5–6), pp. 24–27. Available at: https://doi.org/10.1002/j.2637-496X.2010.tb00263.x.

Fiala, P. and Adamo-Villani, N. (2005) 'ARpm: an augmented reality interface for polygonal modeling', in *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)*. *Fourth IEEE* *and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05)*, Vienna, Austria: IEEE, pp. 196–197. Available at: https://doi.org/10.1109/ISMAR.2005.13.

Fitzmaurice, G. (1996) *Graspable User Interfaces*. Ph.D. Thesis: Doctor of Philosophy. Graduate Department of Computer Science, University of Toronto. Available at: https://www.dgp.toronto.edu/~gf/papers/PhD%20-%20Graspable%20UIs/Thesis.gf.html (Accessed: 14 October 2023).

Fjeld, M., Bichsel, M. and Rauterberg, M. (1999) 'BUILD-IT: A brick-based tool for direct interaction', 4.

Gao, S., Peng, Q. and Wan, H. (2000) 'An Approach to Solid Modeling in a Semi-Immersive Virtual Environment', *Computers & Graphics*, 24(2), pp. 191–202.

Gravity Sketch (2017) *Gravity Sketch*, *Gravity Sketch | 3D sketching and design software*. Available at: https://www.gravitysketch.com/ (Accessed: 13 October 2023).

Holografika (2004) 'Holografika', *Holografika*. Available at: https://holografika.com/ (Accessed: 14 October 2023).

Ishii, H. and Ullmer, B. (1997) 'Tangible bits: towards seamless interfaces between people, bits and atoms', in *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. *CHI97: ACM Conference on Human Factors & Computing Systems*, Atlanta Georgia USA: ACM, pp. 234–241. Available at: https://doi.org/10.1145/258549.258715.

Kim, H. *et al.* (2005) 'Tangible 3D: Hand Gesture Interaction for Immersive 3D Modeling', *IPT/EGVE*, 2005, pp. 191–199.

Kim, M.J. and Maher, M.L. (2005) 'Comparison of Designers Using a Tangible User Interface and a Graphical User Interface and the Impact on Spatial Cognition', *Conference: Proceedings of International Workshop on Human Behaviour in Designing* [Preprint]. Available at: https://www.researchgate.net/publication/231513988_Comparison_Of_Designers_Using_A_Tangible_User_Interface_And_A_Graphical_User_Interface_and_the_Impact_On_Spatial_Cognition.

Leal, A. *et al.* (2011) '3D sketching using interactive fabric for tangible and bimanual input', *Proceedings of Graphics Interface*, 2011, pp. 49–56.

Liang, J. and Green, M. (1994) 'JDCAD: A Highly Interactive 3D Modeling System', *Computers & Graphics*, 18(4), pp. 499–506. Available at: https://doi.org/10.1016/0097-8493(94)90062-0.

MacKay, W.E. (1999) 'Is paper safer? The role of paper flight strips in air traffic control', *ACM Transactions on Computer-Human Interaction*, 6(4), pp. 311–340. Available at: https://doi.org/10.1145/331490.331491.

Malafouris, L. (2013) *How things shape the mind: a theory of material engagement*. Cambridge, Massachusetts: MIT Press.

Maxon (1990) *Cinema 4D*, *Maxon*. Available at: https://www.maxon.net/en/cinema-4d (Accessed: 14 October 2023).

Maxon (1999) *ZBrush*, *ZBrush by Maxon*. Available at: https://www.maxon.net/en/zbrush (Accessed: 14 October 2023).

Meta Platforms (2020) *Meta Quest 2*, *Meta Quest 2: Immersive All-In-One VR Headset | Meta Store*. Available at: https://www.meta.com/quest/products/quest-2/ (Accessed: 23 October 2023).

National Taiwan University of Science and Technology, Taiwan and Lin, M.-H. (2022) 'Examining performance of VR sketch modeling tool in personal sketches', in. *DRS2022: Bilbao*. Available at: https://doi.org/10.21606/drs.2022.200.

Opiyo, E.Z. and Horvath, I. (2010) 'Exploring the viability of holographic displays for product visualisation', *J. of Design Research*, 8(3), p. 169. Available at: https://doi.org/10.1504/JDR.2010.032605.

Opiyo, Eliab Z. and Horvath, I. (2010) 'Towards an interactive spatial product visualisation: a comparative analysis of prevailing 3D visualisation paradigms', *International Journal of Product Development*, 11(1/2), p. 6. Available at: https://doi.org/10.1504/IJPD.2010.032987.

Ratti, C. *et al.* (2004) 'Tangible User Interfaces (TUIs): A Novel Paradigm for GIS', *Transactions in GIS*, 8(4), pp. 407–421. Available at: https://doi.org/10.1111/j.1467-9671.2004.00193.x.

Rosales, E. *et al.* (2021) 'AdaptiBrush: adaptive general and predictable VR ribbon brush', *ACM Transactions on Graphics*, 40(6), pp. 1–15. Available at: https://doi.org/10.1145/3478513.3480511.

Schäfer, K., Brauer, V. and Bruns, W. (1997) 'A new approach to human-computer interaction—synchronous modelling in real and virtual spaces', in *Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques*. New York, NY, USA: Association for Computing Machinery (DIS '97), pp. 335–344. Available at: https://doi.org/10.1145/263552.263638.

Sederberg, T.W. (1986) 'Free-form deformation of solid geometric models', *ACM SIGGRAPH Computer Graphics*, 20(4), pp. 151–160. Available at: https://doi.org/10.1145/15886.15903.

Sidharta, R. (2005) *Augmented Reality Tangible Interfaces for CAD Design Review*. Master of Science. Iowa State University. Available at: https://doi.org/10.31274/rtd-20201107-466.

Smith, D.C. *et al.* (1982) 'Designing the Star User Interface', *Byte Magazine*, 7(4), pp. 242–282.

Son, J., Jang, H. and Choi, Y. (2019) 'Tangible Interface for Shape Modeling by Block Assembly of Wirelessly Connected Blocks', *Journal of Computational Design and Engineering*, 6(4), pp. 542–550. Available at: https://doi.org/10.1016/j.jcde.2019.04.006.

Sullivan, A. (2004) 'DepthCube Solid-State 3D Volumetric Display', *Stereoscopic Displays and Virtual Reality Systems XI*, 5291, pp. 279–284. Available at: https://doi.org/10.1117/12.527543.

Suzuki, H. and Kato, H. (1995) 'Interaction-level support for collaborative learning: *AlgoBlock* —an open programming language', in *The first international conference on Computer support for collaborative learning - CSCL '95. The first international conference*, Indiana Univ., Bloomington, Indiana, United States: Association for Computational Linguistics, pp. 349–355. Available at: https://doi.org/10.3115/222020.222828.

Teng, C.-H. and Peng, S.-S. (2017) 'Augmented-Reality-Based 3D Modeling System Using Tangible Interface', *Sensors and Materials*, p. 1545. Available at: https://doi.org/10.18494/SAM.2017.1703.

The Khronos Group (2017) *OpenXR Standard*, *OpenXR - High-performance access to AR and VR —collectively known as XR— platforms and devices*. Available at: https://www.khronos.org/openxr/ (Accessed: 23 October 2023).

Trimble Inc. (2000) *Sketchup Viewer*, *Sketchup Viewer | 3D Modeling Online Free | 3D Warehouse Models*. Available at: http://www.sketchup.com/products/sketchup-viewer (Accessed: 14 October 2023).

Ullmer, B. and Ishii, H. (2000) 'Emerging Frameworks for Tangible User Interfaces', *IBM Systems Journal*, 39(3.4), pp. 915–931. Available at: https://doi.org/10.1147/sj.393.0915.

Unity Technologies (2005) *Unity 3D*, *Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine*. Available at: https://unity.com (Accessed: 23 October 2023).

Weiser, M. (1991) 'The Computer for the 21st Century', *Scientific American*, 265(3), pp. 94–104. Available at: https://doi.org/10.1038/SCIENTIFICAMERICAN0991-94.

Wellner, P., Mackay, W. and Gold, R. (1993) 'Back to the real world', *Communications of the ACM*, 36(7), pp. 24–26. Available at: https://doi.org/10.1145/159544.159555.

Zmoelnig, C. (2000) *The Graphical User Interface - Time for a Paradigm Shift?*, *Sensomatic.net*. Available at: https://sensomatic.net/gui/ (Accessed: 11 September 2023).

Zucker, M. (2017) *The Perlin Noise Math FAQ*, *The Perlin noise math FAQ*. Available at: https://web.archive.org/web/20170201233641/https://mzucker.github.io/html/perlin-noise-math-faq.html (Accessed: 10 November 2023).