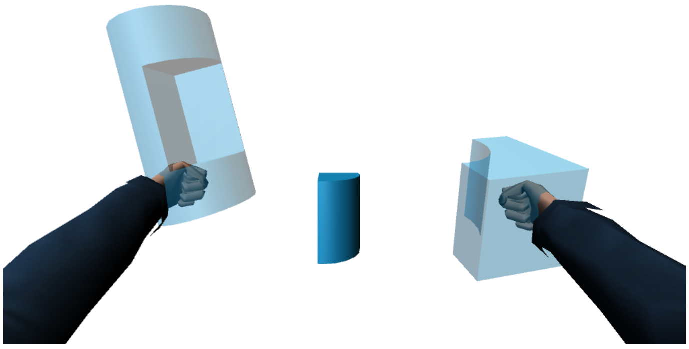


Interaction Techniques for CSG Modeling in VR

Ricardo Jorge da Silva Pinto Ferreira

Thesis to obtain the Master of Science Degree in

Engineering and Computer Science

Supervisor(s): Prof. Alfredo Manuel dos Santos Ferreira Júnior
Prof. Joaquim Armando Pires Jorge

Examination Committee

Chairperson: Prof. José Carlos Martins Delgado
Supervisor: Prof. Joaquim Armando Pires Jorge
Member of the Committee: Prof. Maria Beatriz Alves de Sousa Santos

October 2016

Agradecimentos

Este do documento representa o *terminus* de uma importante fase da minha vida. Uma fase que começa por aprender a escrever e termina com a escrita do que aprendi. Uma fase que não seria de todo possível sem o auxílio de algumas pessoas que merecem o devido reconhecimento.

Primeiramente gostaria de agradecer ao Instituto Superior Técnico, em particular ao grupo de VIMMI, pela oportunidade e meios disponibilizados, e também aos meus orientadores por todo o apoio que me foi dado ao longo desta dissertacao e a oportunidade de trabalhar numa área pela qual tenho bastante gosto e interesse. Gostaria ainda de agradecer ao Daniel Mendes por ter assumido também o papel de orientador e por me ter guiado nesta etapa montanhosa com curvas e contra curvas. Gostaria ainda de agradecer a todos os participantes dos meus testes o tempo disponibilizado.

Gostaria ainda de referir o apoio financeiro da Fundação para a Ciência e Tecnologia, através dos projectos TECTON3D (ref. PTDC/EEI-SII/3154/2012) e IT-MEDEX (ref. PTDC/EEISII/6038/2014).

A todos os amigos que criei ao longo destes anos, principalmente aqueles a quem vou mostrar estas palavras, obrigado por me terem acompanhado nesta aventura e por me terem ensinado uma boa parte das aprendizagens que fiz.

Aos meus pais Manuel e Margarida e à minha irmã Marta, que ainda hoje me servem de modelo, agradeço toda a paciência e o apoio que sempre me foi dado. A vocês dedico esta dissertação e o que ela representa, pois o que sou e consegui, a vocês o devo.

A todos vocês, o meu agradecimento sem cera.

Resumo

Várias áreas usam objectos e ambientes virtuais 3D que são modelados usando diferentes técnicas: sketching, primitivas, manipulação de malhas ou através CSG, uma ferramenta poderosa que permite a geração de modelos mais complexos através da combinação de outros mais simples. Este conteúdo é principalmente produzido com input e output tradicional em 2D. Esta diferença nos graus de liberdade para com interações físicas é contra produtiva. Avanços no hardware de Realidade Virtual tornou o tracking e os HMD mais precisos, confortáveis e económicos. RV permite aos utilizadores operar com os mesmos graus de liberdade que usariam no mundo físico, o que pode acelerar tarefas de manipulações e modelação 3D. Em RV, os utilizadores não conseguem ver o seu corpo, sendo preciso encontrar estratégias para dar aos utilizadores noção da posição do corpo, ou pelo menos das partes que usam para desempenhar as tarefas. Nós propomos duas técnicas para realizar operações booleanas entre dois objectos em RV, baseada em gestos 3D e em menus. Para comparar estas abordagens com uma abordagem tipo, que usa comandos físicos, desenvolvemos um protótipo que implementa essas três técnicas e que utiliza duas formas de representação do corpo: avatar de corpo completo ou duas esferas representando as mãos. Realizamos testes com utilizadores e os resultados mostram-nos que não existe diferença significativa na eficiência de CSG entre técnicas. Os utilizadores preferiram a abordagem dos menus devido ao seu feedback imediato. Descobrimos também que o avatar de corpo completo teve impacto negativo na eficiência neste tipo de tarefas.

Palavras-chave: CSG, Operações booleanas, Modelação 3D, Realidade Virtual, Avatar, Técnicas de interação

Abstract

Many fields use 3D virtual objects and environments which can be modeled using different techniques: sketching, primitives, mesh manipulation, or using CSG, a powerful tool that allows the generation of more complex models using boolean operations on simpler ones. This content is created mainly with traditional 2D input and output. This difference in degrees of freedom to everyday physical interactions is counterproductive. Advances in VR hardware made user tracking and HMD more accurate, comfortable and affordable. VR lets users operate with the same degrees of freedom as they would in the physical world, which has the power to accelerate 3D manipulation and modelling tasks. Moreover, in VR, users can't see their physical body, making it necessary to find strategies to give users the notion of their body, or at least of the parts of their bodies that matter for the tasks they are developing. We proposed two techniques to perform boolean operations between two objects in Virtual Reality based on 3D gestures and menus. To compare those approaches with a baseline approach, with physical controllers, we developed a prototype that implemented all three alternatives and had two types of body representation: a full body avatar or two oval objects for hands. We conducted a user evaluation and results showed that there was no significant difference in efficiency between the three techniques. Users preferred the menu based approach due to its immediate feedback. We also found that having a full body avatar had negatively impacted the efficiency on this type of tasks.

Keywords: CSG, Boolean operations, 3D modeling, Virtual Reality, Self-avatar, Interaction Techniques

Contents

List of Tables

List of Figures

Nomenclature

2D	Two dimensional
3D	Three dimensional
API	Application programming interface
CG	Computer generated
CSG	Constuctive Solid Geometry
DK	Development kit
DoF	Degrees of freedom
HMD	Head mounted display
HUD	Head-up display
IST	Instituto Superior Técnico
IVE	Immersive virtual environment
IVR	Immersive virtual reality
OBJ	Object file
PC	Personal Computer
SDK	Software development kit
SoE	Sense of Embodiment
THI	Two Handed Interface
TRS	Translation, rotation, scale
TV	Television
VR	Virtual Reality
WIMP	Windows, Icons, Menu, Pointer

Chapter 1

Introduction

We live in an age where there is many content that is purely digital and created only with computers. This kind of content is present in various fields: building mock-ups in architecture, human models in medicine, virtual worlds in media like movies, tv-shows, video-games, advertisement, and others. In addition to the wide variety of uses, many of this domains are growing in number of created works which means that the need for those digital models is also increasing.

In the last couple of years the interest in Virtual Reality has seen a big growth with the appearance of commercial products like Oculus Rift, HTC Vive and Playstation VR. These products are the result of the advance in technology of displays and sensors. Mobile displays now have enough resolution and refresh rates to be comfortable to use them in headsets. Sensors can now track the orientation of the head with more precision and are small enough to be integrated on the headset itself. These advancements made possible to use Virtual Reality headsets in a wide variety of applications including entertainment, like movies and games, which generated interest in a wide variety of users. With this new headsets, a new way to interact with virtual environments has arrived as well. Most common interfaces need to be adapted or even rethought to be more efficient and more useful in this new paradigm.

1.1 Motivation

Digital content is used in every field to represent an object or an environment and create new worlds. Virtual objects and environments can be modeled using different techniques: sketching, using primitives, creating the mesh by defining each vertex one by one, or by combining other objects. Boolean operations allow users to generate new and more complex models by combining two simpler ones, a powerful tool to accomplish more complex models faster.

Artists that create those objects and environments usually work with traditional computers with 2D displays and 2D input, might it be a traditional mouse or a 2D surface. Traditional used input only has two degrees of freedom. That is a limitation when using it to operate 3D manipulations because there is no trivial way to approximate those 2-DoF in movement or rotation with precision. For that, most programs started using different modes or 3D widgets. Those are precise but limits user interaction with objects to a degree of freedom at a time and adds time of changing modes to the operations. Immersive virtual environments and 3D inputs could improve those times by allowing input interfaces with the same degrees of freedom as the operation that we want to execute.

Virtual Reality technology lately advanced to the point that is now confortable to use, by increading display's refresh rate and low latency of movement detection. This made it more accessible and more appealing to more people, allowing a growth in the number of users that start to interact with this new virtual worlds. With the release of accessible Head Mounted Displays like Rift, Vive and PS VR, and the wide use of those products in more entertaining experiences like video games and other media makes Virtual Reality more appealing to everybody and not only to professionals.

Because there is no defined standards in 3D interactions, many of those experiences have different approaches and interfaces that may confuse the user and could potentially interfere with the efficiency and fun while using those experiences. Therefore, there is the need to study new ways to interact with virtual environments and to find new interaction techniques that make possible the creation of tools that can empower the user and be a catalyst of productivity.

The growth of use of smart devices like smart phones and tablet computers shows us that interfaces with gestures that are more natural are preferred and more efficient than the common WIMP (WIMP: Windows, Icons, Menus, Pointer) approaches. Consequently we need to find the equivalent natural interfaces to 3D environments that allows users to interact with objects in more natural ways, being more efficient to the tasks performed.

1.2 Challenge

Immersive virtual environments could potentially improve the experience of 3D object modeling and environment creation but we don't have a standard technique to perform those kinds of tasks in these environments. To start using Virtual Reality in CAD like applications we first need to find the proper way to model objects and manipulate them in mid-air. There are some works related to object manipulation in mid-air, some of them presented in this document, but there are less studies related to object modeling and interfaces to perform these operations in immersive virtual environments, especially to perform boolean operations in mid-air. Furthermore, existing approaches don't make use of gesture tracking for more natural ways of interaction. Instead, they still resort to traditional interfaces like 2D menus and physical buttons.

For those reasons, our goal with this work is to study new techniques for modeling 3D objects in immersive virtual environments, with focus on constructive solid geometry, using gesture interactions instead of traditional interfaces while taking advantage of the hardware advancements in head mounted displays and sensors.

While in Virtual Reality, users can't see their own physical bodies because the headsets blocks their vision. Therefore there is the need to give users a good virtual representation of their position, being that through full body representation with avatars or by representing the parts of the body user needs to perform the tasks. With our work, we intend to address the following research questions:

1. Are techniques based on natural mid-air gestures more efficient and appealing to users than approaches based on menu or buttons to perform CSG operations in VR?

2. Can self-avatar help users to be more efficient when performing 3D object modeling tasks?

1.3 Approach

Acknowledged the lack of studies in the field of CSG modeling in immersive virtual environments, we decided to tackle this challenge. We developed two new ways of performing CSG operations in mid-air for Virtual Reality. The first one is based on the natural gestures that people use to interact with real objects, and the second one is based on traditional menus with the addition of a real time feedback of the boolean operations that are available to choose. To validate those techniques, we performed a comparative test with users where we confronted the techniques with each other and with a third technique, our baseline. This third technique is a solution already implemented in a studied work and uses controllers as input, with a button mapped for each of the different operations.

In immersive virtual environments, virtual avatars give users a greater self-report of presence and therefore more immersion. This is known for having a positive influence on the performance of tasks in virtual environments. With our work we want to test if this is also true for the task users have to undertake in our technique comparison. Half of the users performed the test with a full body avatar that mimics all their movements, the rest of the participants carried out the tests with only a virtual representation of their hands. Both portrayals give feedback on position, orientation and gesture of the hands, but only the avatar gives feedback about full body position.

1.4 Contributions

The goal of this work is to develop a new way of performing boolean operations in mid-air for virtual reality environments. We developed and studied two approaches that were evaluated against a baseline in user tests. We also wanted to find out if self-avatars improves the performance of executing our task, which we tested in the same user tests. The main contributions originated by the work here described are:

- **Two new techniques to perform Boolean operations in VR**

  We developed two new techniques to carrying out CSG operations in immersive virtual environments. The first technique is based on the way humans interact with objects in the real world, dragging them to change the position of the objects. This approach implements those interactions to perform the boolean operations. The second technique is based on traditional menus and represents them in the virtual world by a 3D object with buttons that give visual feedback to the users. To validate our techniques we carried out user tests to compare both techniques against each other and against a baseline. The baseline technique, used in a studied work, uses controllers with physical buttons to perform the boolean operations. With our tests we found that there were no significant differences between the three techniques in the efficiency of performing boolean operations. In user inquiries we found differences in easiness, fun, boolean operation recall and boolean operation execution. When asked which technique they prefered as a work tool, 50% of the users chose the menu approach and 45% chose the gestures approach.

- **Assessement of self-avatar influence in task performance test** Avatars can be responsible for user behavior change in virtual environments [1]. With this test we studied the influences of having a virtual body on the performance of a 3D modelling task. We found out that for this kind of tasks, the avatar has a negative effect on user performance.

1.5 Thesis Outline

This thesis is divided in 6 different sections and is organized as follows: Chapter 2 presents related work associated with 3D object manipulation in mid-air, modeling of 3D objects with traditional mouse input, using interactive surfaces and in mid-air and studies about embodiment and the report of self in VR. In the third chapter we present our two techniques to perform CSG operations in mid-air and the baseline technique. In Chapter 4 we describe our prototype's architecture and all the hardware necessary to use it. Within Chapter 5 it is shown the evaluation used to test our hypotheses. We present our environment, our methodology, the attained results and their respective analysis. In the final chapter we have some concluding remarks about our work and deliver some proposals for future work.

Chapter 2

Related Work

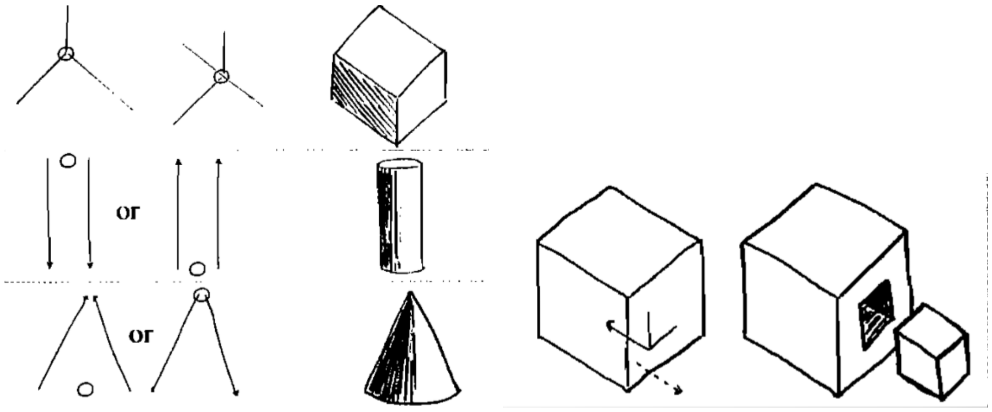
In this Chapter we will study previous research regarding 3D object modeling, 3D object manipulations and embodiment in VR. Because our focus is 3D object modeling with CSG in immersive virtual environments, we will give more emphasis to work related to modeling. We'll start by giving some pratical examples of software used professionally to model 3D objects that have the feature of executing boolean operations on 3D objects. Then, we'll study different approaches of modeling objects: using tradional 2D mouse input, using interactive surfaces and using mid-air gestures. We will also analyze some mid-air manipulation techniques since we also need to implement that type of manipulations in our work. Finally, we also investigate some works about embodiment in Virtual Reality because we want to find the impact of self-avatars in the performance of modeling techniques.

2.1 3D Modelling

Three dimensional objects modelling is the act of creating a new object from scratch or even modifying a previous object to obtain a new one. Instead of constructing very complex models from scratch, boolean operations allows the creation of objects out of simpler objects or primitives. Most objects can be obtained by combining simple parts, such as cylinders, cubes and spheres. An object can therefore be defined by a tree of boolean operations. This approach is known as constructive solid geometry [2]. The basic shapes used to create those new objects don't have to be regular known shapes, like spheres or cubes, they can have any shape. These objects are very useful for creating virtual environments for movies, games, simulations, advertisements and many others. There are several ways to create new objects and in this section we will study some software used professionally to that end and other different approaches to the 3D modeling.

2.1.1 Traditional mouse-based

In this section we are going to cover some techniques that use tradition input, might it be research approaches or common software that are used today by professionals to create virtual environments for



(a) Some primitive creation gestures.

(b) Performing difference on an object.

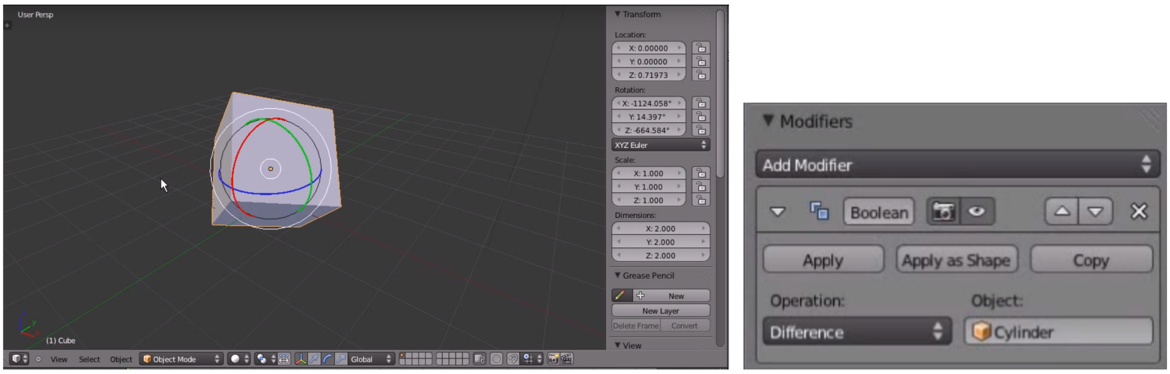
Figure 2.1: Some of the Sketch primitives and a CSG operation being performed [3].

movies, games, advertisement and other design works. All the software have something in common, the fact that everyone of them uses traditional WIMP interface and keyboard and mouse as input.

In Sketch [3] we are presented with an interface for sketching and rapidly create and edit 3D environments that uses a traditional three button mouse as input. Although it uses a mouse, its interface is purely based on gestures and doesn't use the traditional menus and buttons that WIMP interfaces usually offer. Operations are inferred by Sketch based on the gesture performed, and tools are selected with the press of one of the three mouse buttons while performing the gestures: left button to stroke, middle button for interactors and the right one for camera manipulations. Each primitive is created by drawing different sketches, or gestures, with some examples presented in Figure 2.1(a). The editing operations that are possible with Sketch are resizing, drawing shadows, transforms and constraint transformations. CSG opetator Difference is possible by creating a smaller object inside of the one we want to perform the operation to (Figure 2.1(b)).

Besides research work, is very important to study software applications used by professionals to create 3D virtual content. This type of applications are used by a big number of professionals, which makes them a good case study. They have very similar type of interactions but the actions to perform the boolean operations change from application to application.

Blender is a free piece of software that is both an object modelling software and a scene editor. It also has a limited physics engine to create simulations or games with the 3D objects created in or imported to blender. There is an Object Mode (Figure 2.2(a)) to manipulate objects either by moving the 3D widget that appears on them changing either the translation, rotation or scale of the object depending on the selected toggle, or by changing the values by hand on the Transform menu, where there is a text box for each of those parameters. This manipulation allows the users to create a scene by placing an object and move it around to the desired place on the scene. Changing to Edit Mode it is possible to either edit an object or create one from scratch. There is the possibility to create/edit a single vertex, groups of vertices, edges or even faces. CSG operations are also permitted by applying a modifier to an object. With this modifier you can choose which operation you want to do by selecting a drop down menu and the second object that is going to be used in those operations as can be seen in Figure 2.2(b).



Location:
X: 0.00000
Y: 0.00000
Z: 0.71973
Rotation:
X: -1124.058°
Y: 14.397°
Z: -664.584°
XYZ Euler
Scale:
X: 1.000
Y: 1.000
Z: 1.000
Dimensions:
X: 2.000
Y: 2.000
Z: 2.000

Operation: Difference
Object: Cylinder

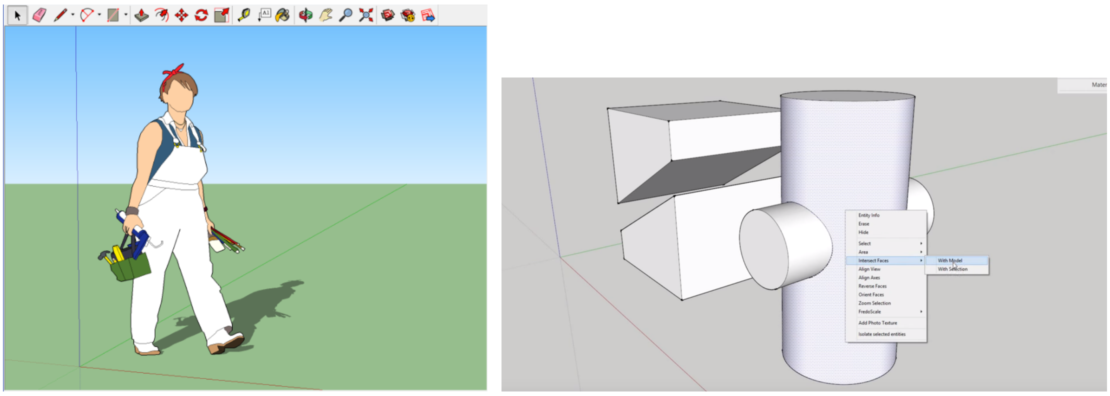
(a) Object manipulations interface.

(b) Boolean operations modifier menu.

Figure 2.2: Some details of Blender's interface.

SketchUp is a 3D modeling program that offers both a freeware version and a more complete and paid version. This software is used to model 3D objects and its advantage next to the competitors is how easy to use it is (Figure 2.3(a)). Simple object manipulation can be perform by toggling the wanted operations on the main menu on the top. After that, by clicking and dragging the operation is applied into that object. To model new objects users have the ability to create shapes by placing vertices with the pencil tool and create a face. Faces can only be created with a square and circle tool that automatically draws a face shaped like those figures with the desired size. Faces can be extruded to give objects volume. Boolean operations difference and intersection can be done between two objects by selecting two intersected objects and clicking with the right button of the mouse and choose Intersect Faces (Figure 2.3(b)). The result intersection can be used in later edits or deleted to reveal the difference.

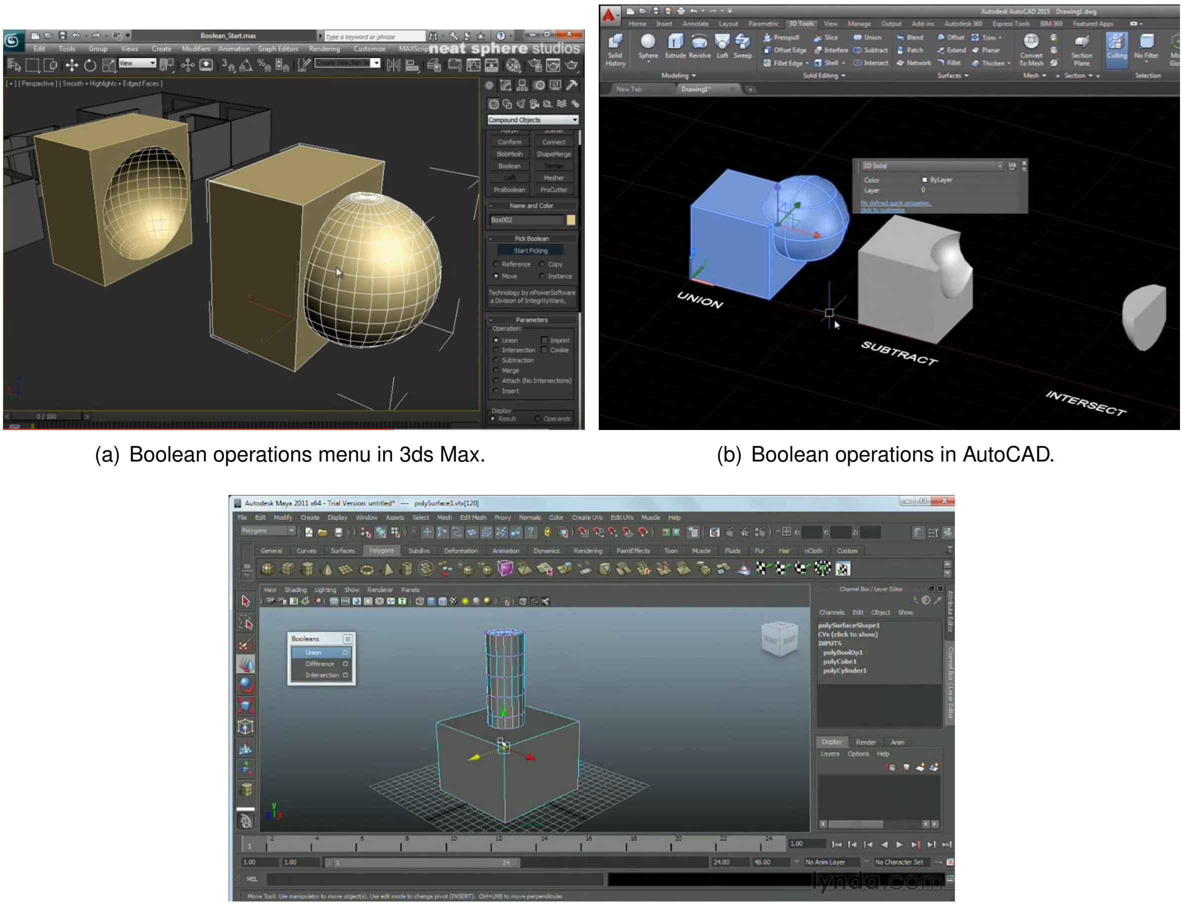
Autodesk distributes many applications where it is possible to create new 3D objects. Exemples of those are Maya, 3ds Max and AutoCAD. Maya is a software designed to create interactive 3D applications like video games and also movies. It can be also used to edit scenes and create objects. Maya supports both modeling and editing by using a mouse on a WIMP interface or via programming through scripts. CSG operators can be applied to two selected objects by picking the right operation on the main menu at the top. This menu can be detached and placed anywere on the screen as seen on Figure 2.4(c).



(a) Main menu at the top.

(b) Boolean operations being selected.

Figure 2.3: Some details of SketchUp's interface.



(a) Boolean operations menu in 3ds Max.

(b) Boolean operations in AutoCAD.

(c) Boolean operations menu in Maya.

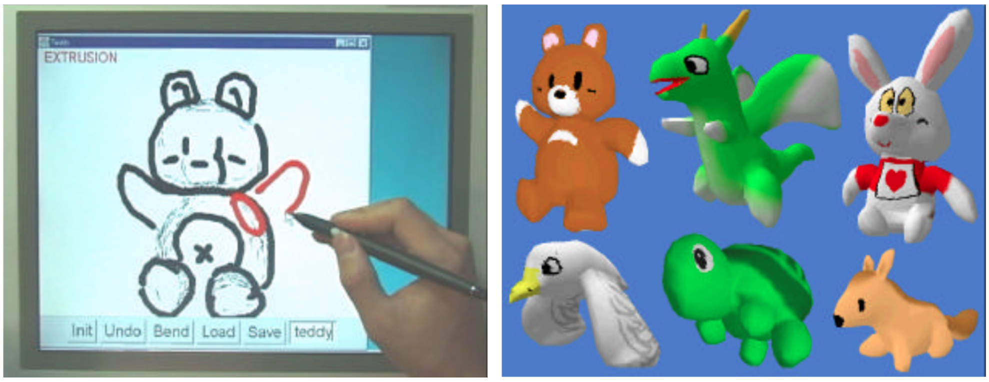
Figure 2.4: Applications developed by Autodesk.

The program 3ds Max can be used to create 3D models, games, animations and more. This tool is widely used by film and game animators because of its powerful engine and the capability of running shaders which lets artists preview their creations as it was running on the target engine. As in previous solutions, 3D object manipulations are done with the mouse and with widgets that appear on the center of the object. Objects can be turned in editable meshes allowing users to create any model. Boolean operations are also done in a similar manor, by choosing an operation on the inspector menu on the right, illustrated on Figure 2.4(a).

AutoCAD was the first product of Autodesk and is used in 2D and 3D design. With similar interface, AutoCAD lets users create new shapes and models by placing vertices. Those objects can be moved or rotated in 3D space using widgets that appear over and around the object. To perform boolean operations there are toogles on the top bar to choose either union, difference or intersection. When chosen, by clicking on two objects and pressing enter, the operation is applied (Figure 2.4(b)).

2.1.2 Interactive Surfaces

Teddy [4] presents an interface to model 3D rotund objects based on 2D sketching, using algorithms to generate objects from strokes. Their interface, which authors claim that can be mastered in 10 minutes,



(a) Interface.

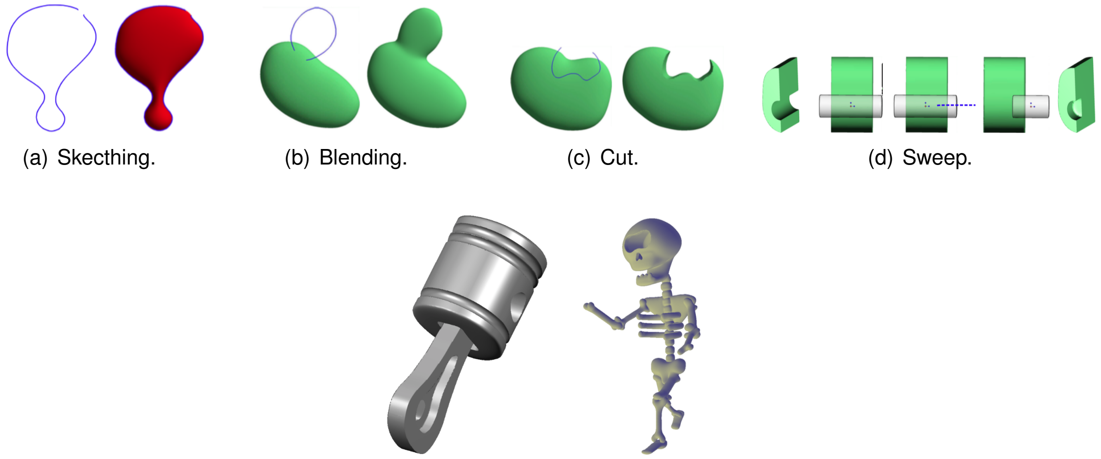
(b) Results.

Figure 2.5: Teddy interface and models done in teddy [4].

uses the 2D input of a mouse or digital pen to generate strokes and then create models proximate to those drawings and is presented in Figure 2.5(a). Operations can be done to the object and they are mostly selected by strokes, allowing users to create objects, draw on them and modify their shape by extrusion or cuts. Users can start drawing from scratch or to load primitives like a cube or a sphere. Objects are created by drawing a closed stroke. Drawings are done by open strokes on volumes, and closed strokes start extrusion mode. A stroke that crosses the borders of a volume cuts the object. Scribbling can be used to delete drawings or, if in extrusion mode, to smooth the shape of the volumes. A transformation mode where objects can be distorted was developed but its interface still required some work. Some works done with this prototype are represented in Figure 2.5(b).

Shapeshop [5] is a sketch based solid modeling system with hierarchical implicit volumes or Blob-Trees. In the style of Teddy [4], Shapeshop offers sketch based operations like hole cutting, oversketch blending and drawing surface detail and adds some extras like smoothing 2D curves and other gestural interface tools. Blobtrees allows a 3D object to be defined as a tree composed of a primitive and a set of operations. Those trees can both represent a scene graph or a history of operations that can be deleted or edited individually at any time via gestures and 3D widgets. There are three different type of surfaces in Shapeshop to create the primitives: blobby inflation (Figure 2.6(a)), linear sweeps and surfaces of revolution. Any of the objects generated with those techniques can be used as primitives to perform the existing operations. The operations that are available to apply to those surfaces are cutting (Figure 2.6(c)), blending (Figure 2.6(b)), surface drawing to add detail, sketch based sweep that is equivalent to the CSG operator difference (Figure 2.6(d)), dynamic 3D clipping and a sketch based resize. Sketches can also be smoothed and erased with scrible like gestures. All these set of operations allows the creation of characters designs or technical drawings of mechanical parts as can be seen in Figure 2.6(e).

Shapeshop was later extended by Lopes et al. [6] combining bimanual touch manipulation to the pre-existant pen based input. This results in a multi-touch version of Shapeshop that uses pen input for more precise operations like sketching and touch input to secondary operations like toolbar selection and camera manipulations. The resulting gesture interface can be seen in Figure 2.7. They also provide a comparison of their approach to the pen-only interface and concluded that combining both interfaces



(a) Skecthing. (b) Blending. (c) Cut. (d) Sweep.

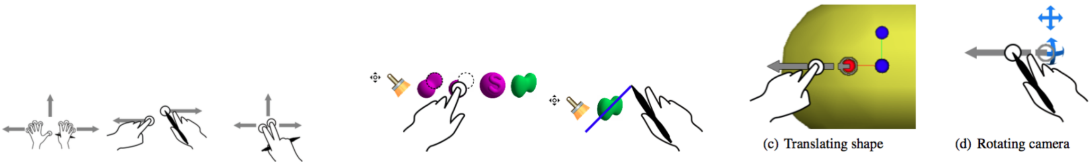
(e) Results.

Figure 2.6: Shapeshop operation examples and results [5].

results in a simplified workflow and lowers task times when compared to the traditional interface present in most sketching apps.

ErgoSketch [7] is a system that joins the Sketch interface to the ErgoDesk framework to support 2D pen-based gestures and also 3D interactions. They replaced Sketch's mouse with a lightpen that still has 3 buttons to use the same operations. Three-dimensional input is done by a 6-Dof Hand tracker which directly applys translations and rotation if an object is selected. Like in the previous work, this approach gives secondary operations to the non-dominant hand to let users use both hands at the same time. While drawing, it is possible to perform camera movements with a trackball as seen in Figure 2.8, and while manipulating objects in 3D space, it is possible to do annotations in 3D space.

Arwand by Ha et al. [8] is a system to build virtual environments in Augmented Reality. The system tracks the physical environment to generate a virtual representation where objects will be placed. With the help of the HMD, user sees the composition of those two worlds, the real world with the virtual objects the user created or imported, since ARWand lets you import any 3D model and use it in the virtual world. Their solution also allows the users to create new objects using the smart phone. For that the user use the gyroscope capabilities of the device to define a plane in 3D space equivalent to the inclination of the phone and then with the finger draws a line that will define the object (Figure 2.9). By repeating those two steps, users can create an object using only one hand. Those objects are then usable in the augmented experience. Their interface to manipulate the world is a circular 2D HUD menu



(a) Camera manipulation. (b) Toolbar selection. (c) Widgets.

Figure 2.7: Multi-touch enabled ShapeShop [6].

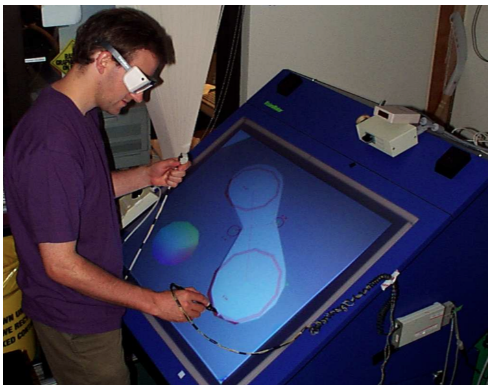


Figure 2.8: Ergosketch user creating an 3D object with right hand and manipulating the camera with left hand[7].

appearing on the screen.

2.1.3 Modelling in mid-air

A 6-Dof Hand mouse is used in 3DM [10], to move a cursor in the 3D virtual space and to interact with the objects in an immersive virtual environment. Their interface uses a 3D cursor, that follows the position and orientation of the mouse, and a toolbox (Figure 2.10(b)), that is located near the user's waist and contains selectable 3D icons with tools, commands and toggles. Their tools consist in Surface creation, Editing and Hierarchy. When in Surface Creation users can create vertices in 3D space on the position of the cursor (Figure 2.10(a)). With those vertices it is possible to create triangles or triangle strips to make the objects. There is an extrusion tool to create more faces and there is the option to create some primitives like boxes, spheres and cylinders that can be edited later. Selected objects can be manipulated in Editing mode. Selection is done by capturing the object with an area created by the mouse (Figure 2.10(c)). There is the option to translate the entire object or a group of vertices of that object. Scale manipulation can also be done to an entire body or to a smaller group of vertices. Undo and redo operations are also available. Hierarchy or grouping allows to apply the same manipulations to a group of pre-selected objects. This solution lacks some constraints that are present in common CAD tools but compensates by allowing to snap the cursor to a 3D grid or to a 2D plane. They concluded

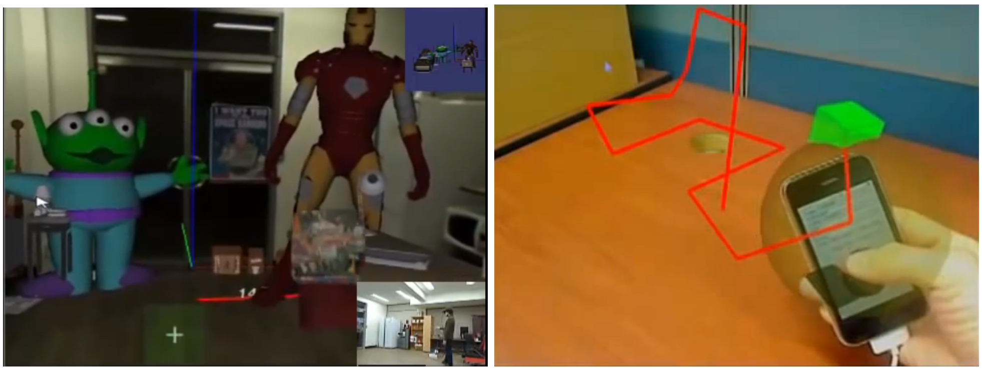
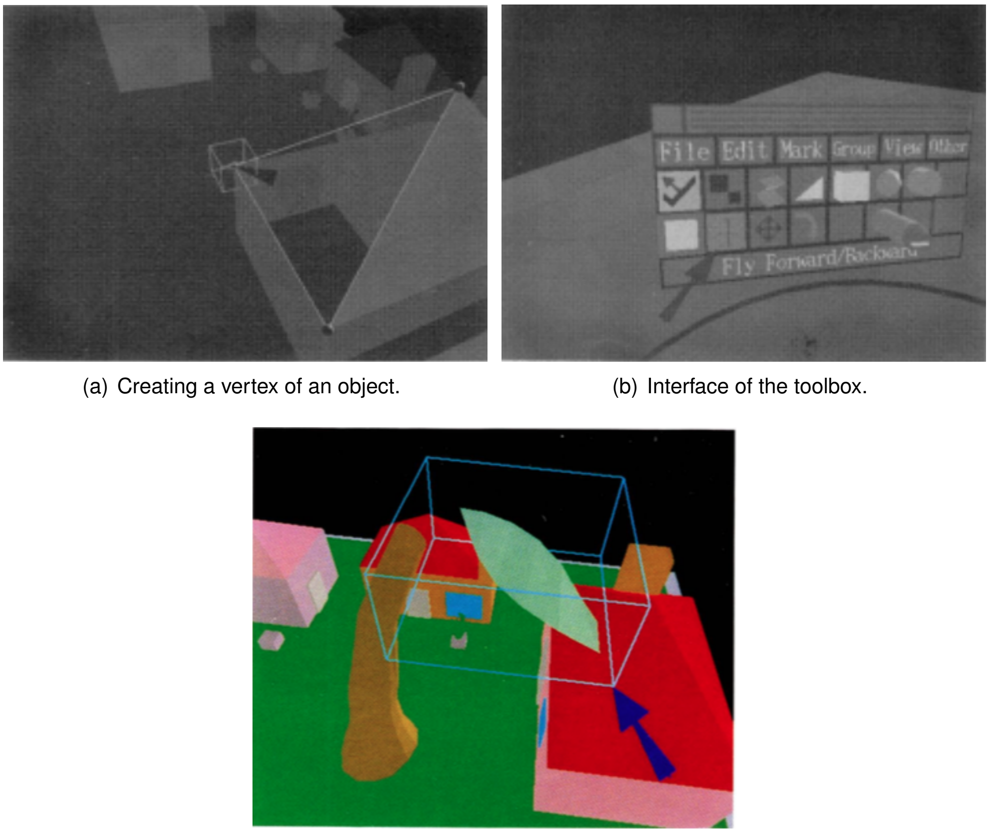


Figure 2.9: Arwand result (left). Creating objects with the smart phone (right)[9].



(a) Creating a vertex of an object.

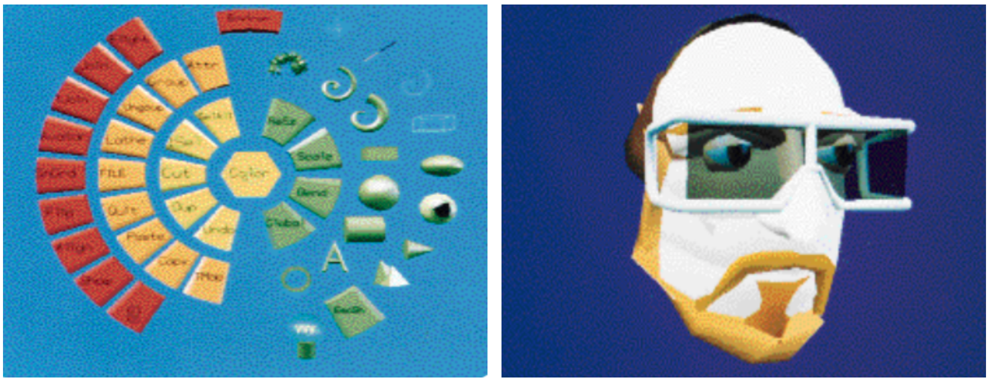
(b) Interface of the toolbox.

(c) Selecting an object.

Figure 2.10: 3DM interface. [10]

that advanced users are empowered because of the use of an HMD and because of some complex operations like translations and rotations being done at the same time with the 6-Dof Hand mouse in contrast to what we find in common CAD tools.

The HoloSketch [11] is a VR Sketching System that was developed based on the supposition that VR-based editors would be easier to use and more productive than standard 2D modeling and sketching interfaces. For interaction, they used a 3D mouse extended with a digitizer rod tip for direct manipulation cursor and a keyboard for translation and rotation operations. To alternate between the available modes(drawing and edit) and to choose operations they build a menu that is cast by holding a mouse button. That menu(Figure 2.11(a)) is based on common 2D menus with adaptations to surpass some limitations of traditional menus. Standard interfaces occupy too much screen space and steal too much cpu cycles and causes confusion when used with head tracking. Their menu is cast with a button press fading out everything behind it and is centered on the position of the tip on that moment. It is a 3D pie menu to decrease the distance that user needs to move to choose a button, which could potentially be greater on 3D environment than what it would be on 2D counterpart. Buttons are selected by poking them with the tip of the mouse. The system offers two modes which can't be used simultaneously, draw and edit. Drawing allows to generate primitives like cubes or spheres in the environment or to draw freely in 3D using a button. Edit allows to change the scale/form and color of the objects created in draw mode. Objects can only be edited when selected, by pressing a button when touching them with the tip.



(a) Menu.

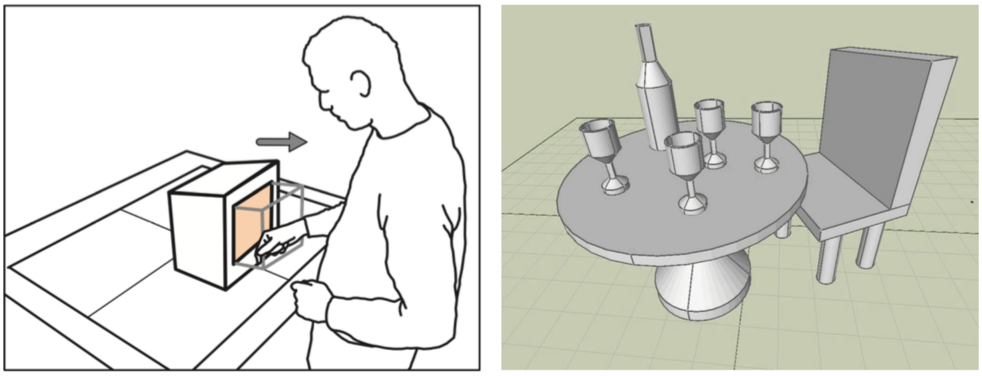
(b) Result.

Figure 2.11: HoloSketch interface and model done with the system[11].

Translation and rotation operations are executed with the mouse and the help of the keyboard. A model done with Holosketch can be observed in Figure 2.11(b).

Mockup Builder [12] shows us a direct modeling approach that mixes 2D interface on a multi-touch gesture surface and a three-dimensional interface with gestures in space above the surface. The surface has a stereoscopic capable screen to visualize objects in 3D through an above the table experience. Objects can be created by sketching a face on the surface and using push and pull 3D gestures to give it depth or create other faces. Push/pull gestures offer five diferent operations: change geometry of an object withouth changing its topology, extruding a face using its normal as a direction vector(Figure 2.12(a)), extruding a face through a free path drawn with 3D gestures or 2D sketch, split faces by drawing strokes on them and finally to switch between surface and space editing modes. These gestures are similar to push/pull gestures present in Google Sketch-Up or Sesame [13]. Sketch-up also offers the possibility to edit edges of polygons individually. To scale an object you pick it with both hands in the 3D space and move them far away from each other. An example of a result done with Mockup Builder is represented in Figure 2.12(b).

In the work by Wang et al. [14] users can create and interact with environments with a magic wand metaphor, by pointing at the objects. Users can move in a limited area and receive haptic feedback



(a) Extruding with pull gesture.

(b) Result.

Figure 2.12: Using Mockup Builter and an example of a result of using the system [12].

when they reach the limits. Travel feedback is given by fans, simulating the movement wind. Although users perceive the virtual world through an head mounted display, tools are chosen in a 2D interface in a smartphone placed on the non-dominant wrist (Figure 2.13). Users can manipulate one object at a time or use brushes that create multiple objects, like leaves of grass, and also the option to elevate or depress terrains. Objects are created in a position where user's are pointing to and their materials can be changed on the 2D menus. There is also an option to create text labels on objects. Text labels are typed using the smart phone interface for easy access to a keyboard. On the smart phone users can also change the lightning scheme and choose a position for light origin.

In the work by Barot et al. [15] users can manipulate an environment in Virtual Reality. It offers a multimodal interface where users can interact using the magic wand metaphor or by voice commands. Users create objects and change modes by interacting with a menu where each option is represented by 3D bubbles (Figure 2.14) that can be selected with the wand or by saying the label of the option. Selecting some of this options brings another set of choices and users have to navigate this menus to select primitives or operations. It is possible to change the scale of the virtual environment to get both the first person perspective or the mock up perspective.

MakeVR by Jerald et al. [16] has a Two-Handed Interface(THI) engine as described by Schultheis, U. et al. [17], where hands are represented by two needle-like pointers that mimic user's movements. Interactions with objects are made by touching and dragging them. This works in a very similar way that gestures do on today's smart-phone operative systems. It is possible to grab the environment and spread the hands to zoom or drag to move around. Those operations are also applied to the objects, dragging the objects to move and rotate them simultaneously. Their approach also allows users to do the following Boolean operators on objects: unions, intersection and difference (Figure 2.15). Boolean operations can be performed between two objects by joining two objects until they intercept each other. By pressing a button in the controllers you choose which one you want to apply to your objects. It is possible to apply unions, difference or intersection. The operation result is applied to the first selected object. There is a 3D menu that lets users change the colors and properties of the objects.

In Takala et al. [18] it is possible to manipulate objects and scenes in 3D. Their system was built on top of Blender which is free and open source. Their solution can be separated in two different modes: a 3D Model builder and a Scene Builder (Figure 2.16). On the first one user has the ability to create primitives and to extrude new vertexes and faces from an existing one, and it also offers the ability of

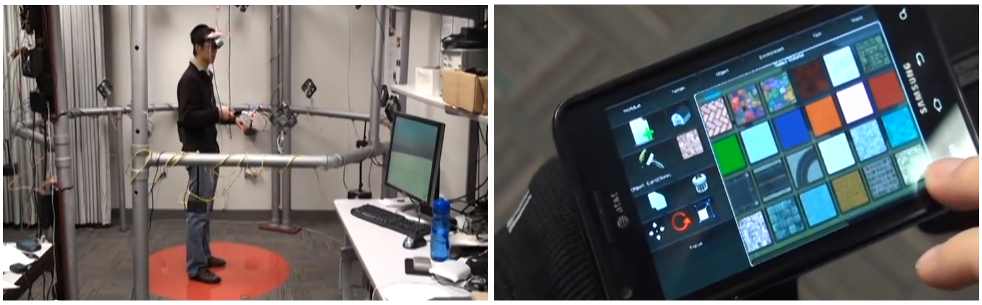


Figure 2.13: User in DIY (left). Smart phone on the wrist(right) [14].

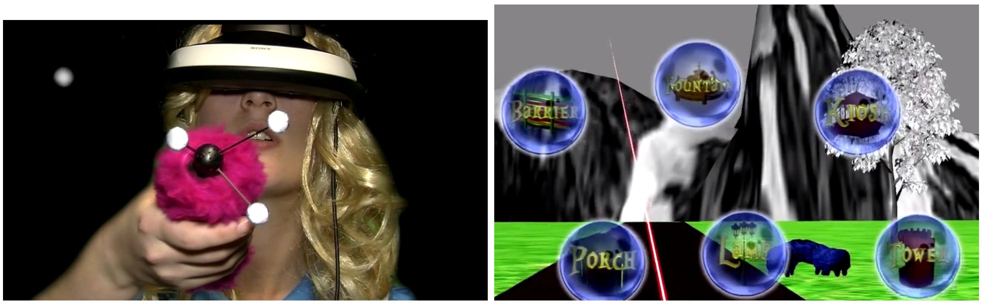


Figure 2.14: User in Wonderland Builder (left). Menu options interface(right) [15].

drawing a 3D object using a 3D meta-ball as a brush. Combining those two possibilities you can sculpt any object and then paint it with colors or textures and use the final creation on your virtual world. The second mode, Scene Builder, allows you to import any 3D object and use it to create virtual worlds. Users interact with the worlds with a controller in each hand. The non-dominant hand is responsible for manipulate the camera and change tools while the dominant hand performs the main operations. Every option and tool is explained in a static 2D HUD that is always visible on the screen.

In Cochard et al. [19] objects are created and manipulated by stroking with the point of the finger in the 3D space. The non-dominant hand is responsible for rotations of objects or menus, which are performed by rotating a physical ball (Figure 2.17). The menus are a sphere with the options on its surface, as the user moves the ball with his hand. The options are selected by rotating the ball until the wanted option is in front of the user, and then clicking a button in the hand of the user. It is possible to perform difference operation by cutting it with strokes.

In Hald, Kasper [20] user interact and create environments with a controller in each hand. These controlleres have one joystick and two buttons and they are used to move the user through the virtual world. Hands are tracked in 3D space and represented in virtual space as two spheres. Users interact with objects through 2D menus that appear on the screen. As seen in the Figure 2.18, objects are selected by creating an area with the position of the hands. Every object inside that position is selected and target of the following operations.

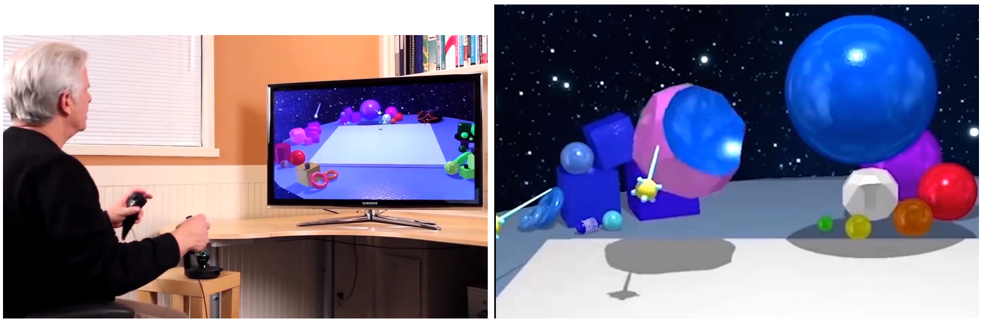


Figure 2.15: User in MakeVR (left). Boolean operator not (right)[16].

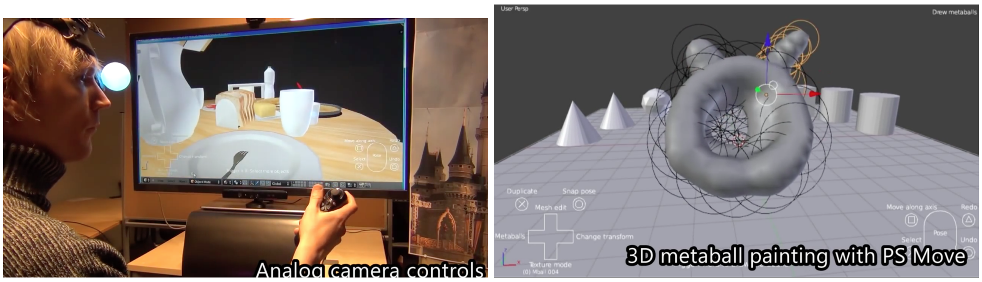


Figure 2.16: Head Tracking (left). Meta ball object modeling (right) [18].

2.2 Object Manipulation in mid-air

Handlebar [21] is a technique to manipulate objects in mid-air. In this approach, users manipulate objects as they are skewered in a handlebar. Like depicted in Figure 2.19, users move both hands in the same direction to translate the object, rotate them around the object to rotate the object and move the hands away from each other to scale it.

The Go-Go [22] is a technique to manipulate out of reach objects that uses the metaphor of stretching the arm of the user. This technique is inspired by the Go Go Gadget cartoon, which had the ability to stretch its arms. Go-Go offers a seamless way to interact with close objects and out of reach objects (Figure 2.20) and mimics what humans do to reach objects, they stretch their arm.

Doug Bowman and Larry Hodges [23] studied the difference of using arm-extension techniques and ray-casting techniques to grab and manipulate objects in VR. They found that although ray-casting was better for grabbing objects due to its unlimited range, object manipulation like the Go-Go technique [22] were preferred because were closer to natural manipulation of objects. This lead them to build an hybrid technique that used ray-casting for selecting objects and hand-centered object manipulations, the HOMER.

MuJoCo HAPTIX [25] presents a system to track hand gestures using a CyberGlove. This system gets very close approximations of virtual hand gestures which enables an object manipulation based on simulated physics, emulating the real world. The downsides of this solutions are the use of an intrusive system, the glove, and its high price.

Five different 3D object manipulation techniques were implemented and compared in Mendes et al. [24]. Four of those techniques were done in mid-air and the other in the surface of stereoscopic

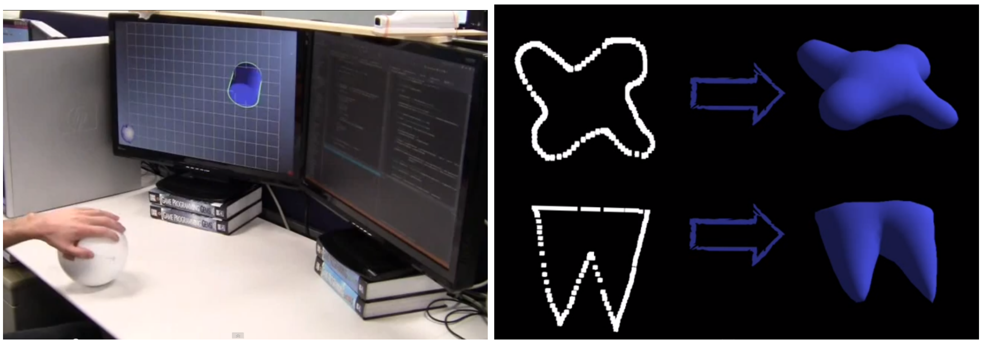


Figure 2.17: Ball interface (left). Creating objects from a drawn line (right)[19].

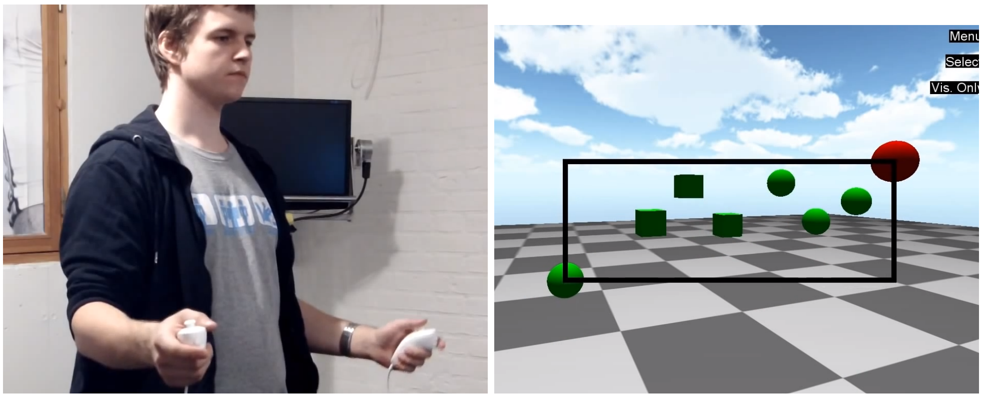


Figure 2.18: Using the project (left). Selecting multiple objects (right) [20].

interactive tables. To interact with objects, hand position and gesture was tracked. Grabbing an object was done by pinching on an object (Figure 2.21(a)). While hovering an object a wire bounding box would appear and turn translucent when the object was grabbed. The four mid-air implemented techniques were 6-Dof Hand, 3-Dof Hand, Handle Bar and Air TRS. In 6-Dof Hand (Figure 2.21(b)) the grabbed object mimics the movement and rotation of the hand, uniform scale was performed by pinching empty space with the second hand and moving hands away from each other. 3-Dof Hand (Figure 2.21(c)) is similar to 6-Dof Hand but the translation operations is performed with the grabbing hand and rotation operations are done by rotating the other hand. Uniform scales are still done with moving hands away from each other. Handle bar (Figure 2.21(d)) mimics a physical bimanual handle bar. This technique uses the middle point of both hands to manipulate virtual objects. By moving both hands in the same direction, translation operations are performed. Moving in opposite directions, a rotation operation is performed. Moving hands away from each other evenly performs an uniform scale. Air TRS is based on Two-Point Rotation and Translation with scale [26] and adapts it to the third dimension. In Air TRS (Figure 2.21(e)) the grabbing hand movement translates the object, second hand rotates the object by moving around it and scale is also done by moving hands away from each other. By comparing these techniques they concluded that 6-Dof Hand was the more natural way to interact with objects in 3D space. Handle bar technique is as fast as 6-Dof Hand to perform the tasks and Mid Air manipulations appeal more to users and are better for new comers than performing these kind of manipulations with 2D inputs.

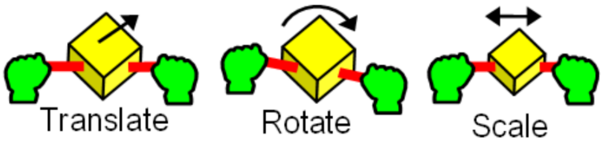


Figure 2.19: Object manipulations with handlebar technique. [21].

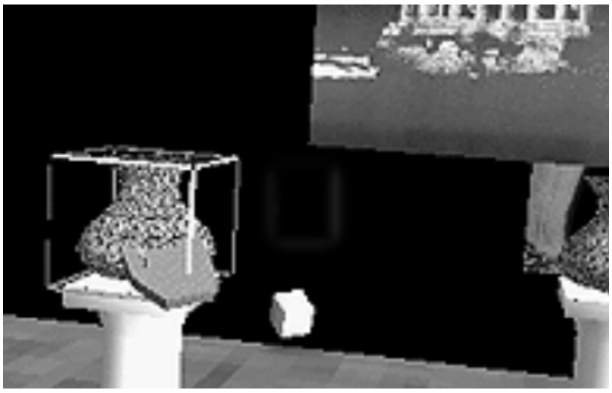
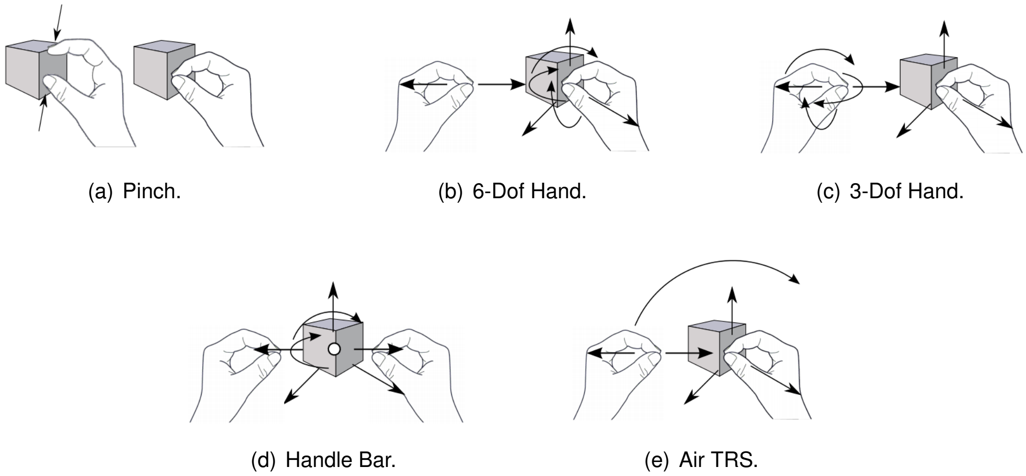


Figure 2.20: Virtual hand touching out of reach object. [22].

2.3 Embodiment in VR

Embodiment has many meanings in different fields like philosophy, neuroscience and psychology, robotics and in relation to presence in virtual environments. To distinguish the last use from the others, Kilteni et al. [28] suggested a definition for the term Sense of Embodiment *as the sense that emerges toward a body B when its properties are processed as if they were the properties of one's own biological body*. As the authors tells us, this definition is vague and the properties of the body need to be specified. Three properties are presented to us in their work: Sense of self location, sense of agency and sense of body ownership. The first property concerns to the volume that the body occupies on the world where we perceive that we are. This property is intensified by watching the world from a first person perspective of the body, by watching the body move accordingly to our movement and by seeing the body respect the same physical laws than the biological body does. Sense of agency exists when there is a feel of control over the actions of the body, which is helped by the perfect synchronism to the biological body. The last property of a body is Sense of ownership and is present when the body is felt as ours and real. For that we need to feel the stimuli which affects the body as we would in our biological body. A way of evaluating this task is proposed by evaluating the performance in motor tasks while in a virtual environment. In this



(a) Pinch.

(b) 6-Dof Hand.

(c) 3-Dof Hand.

(d) Handle Bar.

(e) Air TRS.

Figure 2.21: Mid air techniques [24]

work it is also presented the notion that SoE is not a binary value but a scale with different weights for each of the properties of the body. These contributions and relationships are not defined and still need to be studied.

Steed et al. [27] developed an experience in VR to test embodiment using consumer VR equipment. They invited people that already had devices like Google Cardboard and Samsung's Gear VR to download their application and test it in their own environment giving the testers no control over the experiment, therefore, the experience was 'in the wild' and not expensive to users. The objective of this work was to establish new guidelines in VR consumer application because they find that there are many different interfaces to the same task in VR, which may cause confusion to the users. They had, as well, the goal of testing the feasibility and utility of an 'in to the wild' study and to prove three hypothesis: that self-avatar improves self-report of presence, that other avatars engaging with the participants improved self-report of presence and last but not least that avatar moving itself would improve the self-report of embodiment. Their experience, as depicted in Figure 2.22, was to put the user in a seated position in a bar, listening to a live performance of a singer. In front of the user there was a table with a small box on top of it, another avatar facing the singer and the stage where the live performance happens. The headsets used are very simple and have no controllers or body trackers, so the only input of the users in this experiment was head orientation. There were data collected throughout the experience and there was an inquiry in the end, still using the headsets. The application had different configurations for different users. User avatar had sex variation based on an introductory inquiry. The avatar was present for some users, hidden for others. Singer could look to the user, or look in other direction and for a set of users it would ask for them to tap along the song. One of the reasons that they chose a seated experience was the fact that they had no control over the environment where the experience took place and could be potentially dangerous to participants to do tasks where movement was required. To test the effect of the avatar on self-report of presence the box on the table would fall and touch the users avatar and their reactions were recorded. Self-report of presence was tested by having the singer engaging with participants by looking in their direction. Self-report of embodiment was tested by having the avatar tap along the song when asked by the singer. Singer looking in the direction of the user had no effect on them but could be because of the type of experience, the avatar didn't helped users to feel presence but they did notice the box hitting them. Tapping had a negative effect on embodiment because took control

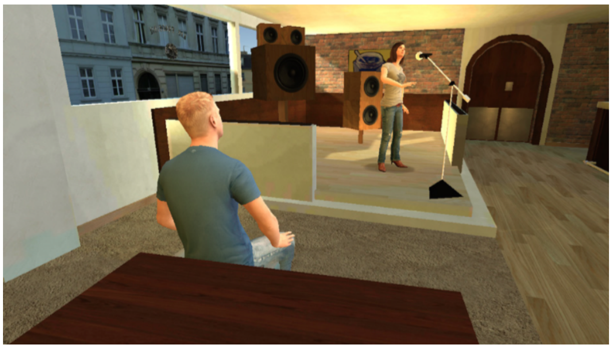


Figure 2.22: Screen shot of the 'in the wild' experience [27].

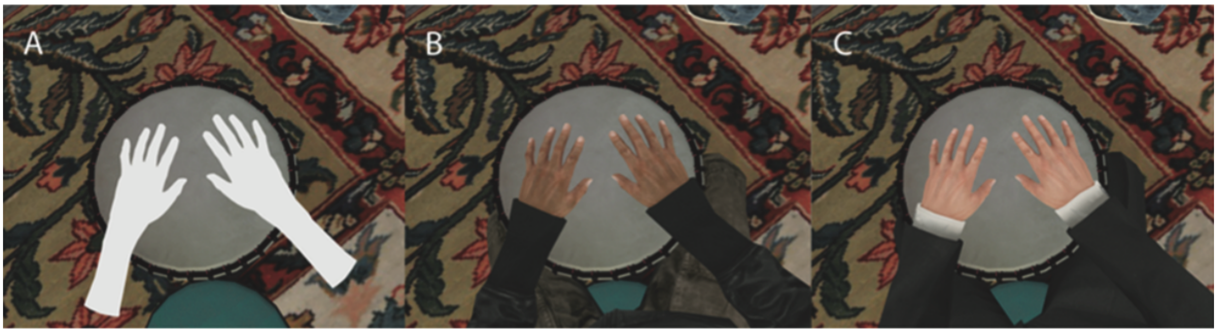


Figure 2.23: Different representations of the participants [1].

over the body like it was explained by Kilteni et al. [28].

In the works of Kilteni et al. [1] it was studied the hypothesis of the form of an avatar influencing behavior of a user in Immersive Virtual Reality. To test this, participants were asked to freely express themselves with an African Djembe in Virtual Reality. The experience was divided in two phases, one with plain white virtual hands and the second with a full body avatar with a more realistic look, as represented in Figure 2.23. Users were separated in two groups, each one with a different avatar: one formal light-skinned and one casual dark-skinned. Participant feedback told them that the casual dark-skinned avatar was more appropriate to perform that task and that only those users felt that they were more expressive than usual. With the logs collected throughout the experience, it was also possible to conclude that when using the dark-skinned avatar, users were more expressive than with the baseline white hands. They conclude that ownership illusion implies behavioral and cognitive change.

A study on the impact of avatar realism on embodiment was done by Lugrin et al. [29]. They gave their users different types of avatars to use in a game like experience where participants had to touch virtual

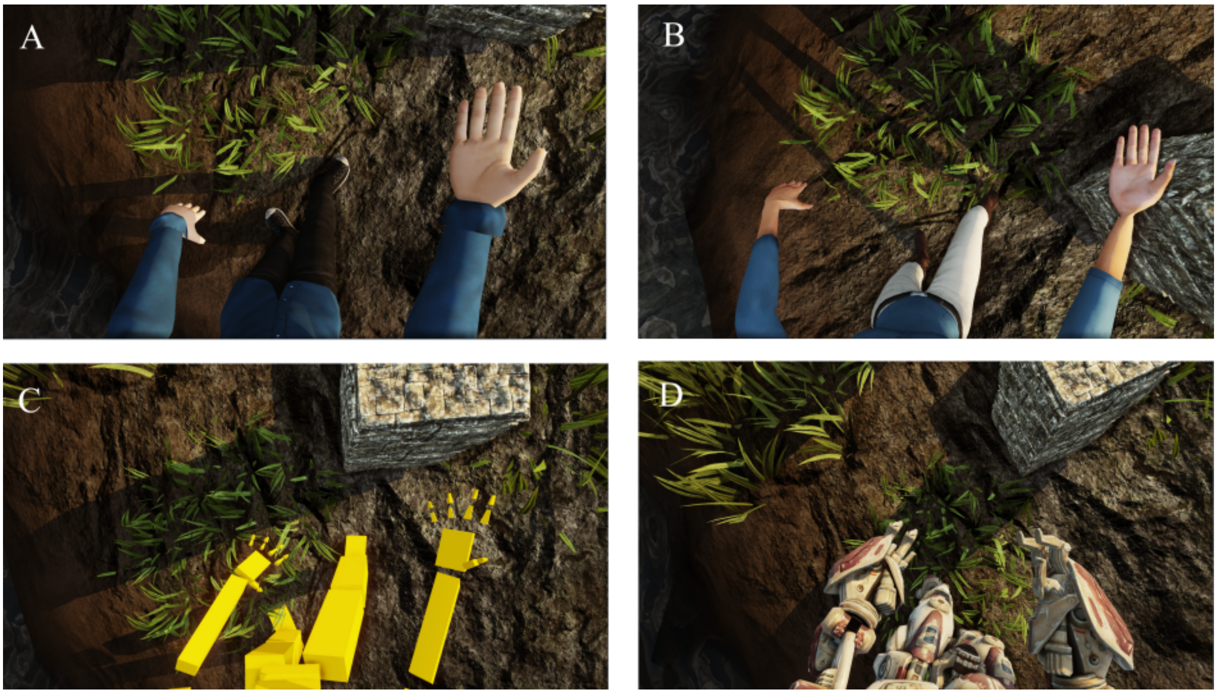


Figure 2.24: Diferent avatars configurations on user's point of view. [29]

targets with two configurable conditions. The first one is the possible presence/absence of a permanent threat like fire torches and the second is the possible existence/absence of a sudden threat represented by an explosion. Each participant had one of the following types of representations: cartoony robot, a stylized human or a photo-realistic human, represented in Figure 2.24. Their results shows that machine-like avatars offer a bigger illusion of virtual body ownership which can be explained by the fact that users noticed little details, like arm length, to be different from the real body more on the photo-realistic avatar than on the others. The conclusion of this work is that is possible to represent virtual bodies with non realistic avatars and still give the sense of embodiment in virtual reality.

2.4 Discussion

In this section we are going to evaluate and compare the works we described throughout this chapter. As can be seen in the Table 2.1, we are going to classify them by their input, output, by their ability of modeling 3D objects, how they do it and finally, by the ability of performing CSG operations and how they do them. Because it is not the focus of our work we will discuss briefly object manipulation. In the end we will also examine the embodiment works that we portrayed before.

The works we studied have different ways of interaction, and different types of visual feedback to the user, illustrated in Figure 2.25. To compare and evaluate them is important to have a proper taxonomy for both the input and output of the solutions[1]. Input can be of two forms, 2D or 3D. Two dimensional input is the most common way to interact with devices nowadays, either it is by using a traditional mouse or by using a touch surface. Three dimensional input requires spacial tracking which can be performed, for example, with magnetic motion sensors or with depth cameras. Visual feedback can be given to the user in both 2D or perceived 3D. Two dimensional output can be given by using a regular display. Three dimensional output is achieved by using a display with stereoscopic capabilities. Examples of stereoscopic displays are 3D televisions and Head Mounted Displays. HMDs provides users a more

[1] This is adapted from the taxonomy proposed by Grossman and Widgor [31]

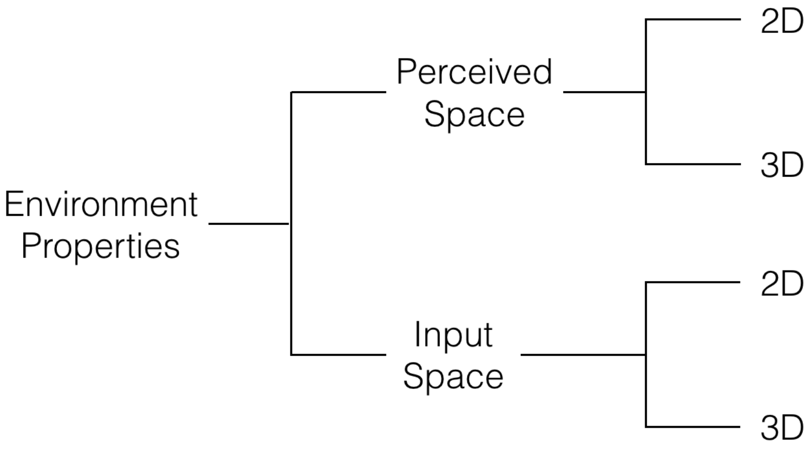


Figure 2.25: Environment properties.

	Input Space		Perceived Space		Volume Creation					
					Primitives	Sketch	Vertex manipulation	Extrusion	CSG	
	2D	3D	2D	3D					Single Object	Two Objects
Sketch [3]	●		●			●			●	
3DM [10]		●		●	●		●	●		
Teddy [4]	●		●			●			●	
Shapeshop [5]	●		●			●			●	
Lopes et al. [6]	●		●			●			●	
ErgoSketch [7]	●	●		●		●			●	
Arwand [8]		●		●	●					
Holosketch [11]		●		●	●	●				
Mockup Builder [12]	●	●		●		●		●		
DIY World Builder [14]	●	●	●	●	●					
Wonderland Builder [15]		●		●	●					
MakeVR [16]		●	●		●					●
Immersive 3D Modeling [30]		●		●		●	●	●		
Building Worlds with Strokes [19]		●	●			●		●		
Low-Cost 3DUI [20]	●	●	●		●					

Table 2.1: Classification of related work.

immersive experience because it gives the illusion that they are in the virtual world, whereas 3D TV's experience is more like a window from where objects protrude.

Many of the works we studied use 2D input, might it be with a traditional mouse, with a drawing pen or even with a touch enabled surface. There are some examples that use 2D input exclusively [3, 4, 5, 6], others that combine 2D input and 3D input [7, 12, 14, 20] and finally works that only have 3D input [10, 9, 11, 15, 16, 30, 19]. Modeling three dimensional objects requires manipulation in 3D to place the vertexes or change their position, to move or rotate objects or parts of objects and even to move the camera around the objects to get a view of every angle of it. These manipulations can be done using 2D input but its conversion is not trivial to implement or to use. An alternative to that conversion is to manipulate an axis at a time, for position, and then again one axis at a time for the rotation or to put the values of position and rotation manually in text boxes. When using this solutions, users take more time to manipulate objects or the camera because they can only manipulate one axis a the time and the change of axis from one to another also costs time. Three dimensional input gives the user the power to place or edit vertices and move objects with the same degrees of freedom as those manipulations require. Using this input makes the process faster because users don't have to switch between axis to manipulate them and with the proper sensors, translations and rotations can be done at the same time. We studied some works that had 3D input to both manipulate and model objects and will be our inspiration to achieve these kinds of manipulations in virtual reality.

We found different types of output as well as we found in input. Output in the studied works uses standard 2D displays [3, 4, 5, 6, 16, 19, 20], 3D capable displays [7, 11, 12, 30] and head mounted displays [10, 9, 14, 15]. In the first case, the 3D model is not displayed like it really is because the displays can not show proper 3D. Instead they display a projection of those models which doesn't give 3D content creators the proper perception of their creations. The second case, 3D TVs can give a better perception of the models but users still need to manipulate the camera to be able to see the object from every angle. Finally with head mounted displays, users can see their objects in 3D and can move the head and even walk around the object, with the proper tracking, to see objects in another perspective. This last example is where users are more immersive and combining that experience with proper 3D input can give the users an optimal modeling experience.

We also defined a taxonomy for volume creation approaches in 3D modeling tasks, illustrated in Figure 2.26. We divided the works we studied in five types of modeling as can be seen in Table 2.1: primitives, sketching, extrusion, vertex manipulation and CSG operations. Modeling with primitives is when you use simple primitives as cubes, spheres or cylinders to achieve more complex objects or environments. Modeling by sketching is when you draw lines and 3D objects are generated based on those drawings. Extrusion is when you select a face, or part of a face, of an object and you drag it to create a new volume for that object. With vertex manipulation you can edit the vertices of a mesh by placing new ones and creating new polygons or editing the placement of existing ones. CSG operations is the ability of performing boolean operations on objects: union, difference and intersections.

Using this taxonomy we divide works in those five types of modeling: using primitives [10, 9, 11, 14, 15, 16, 20], sketching [3, 4, 5, 7, 11, 30, 12, 19], extrusion [10, 12, 30, 19], vertex manipulation [10, 30]

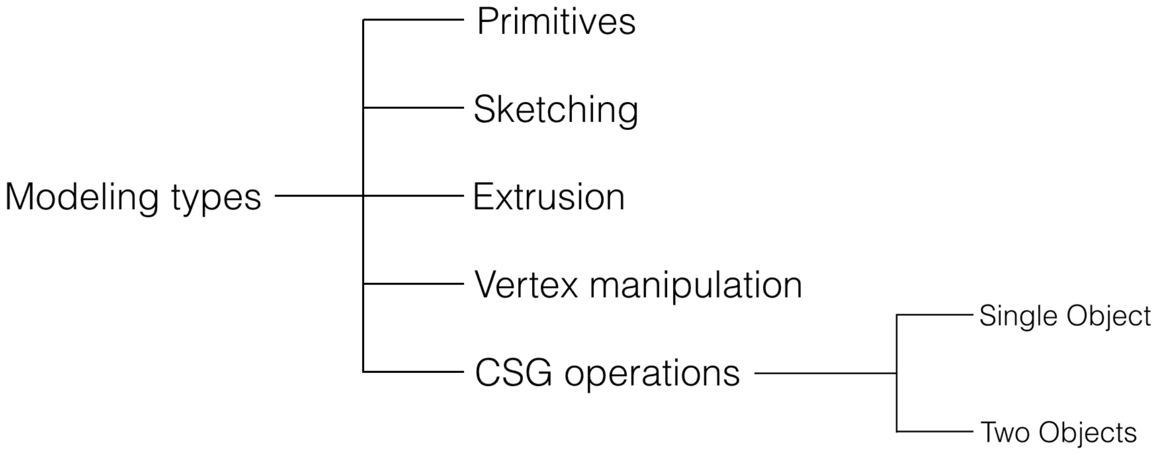


Figure 2.26: Modeling types.

and using CSG operations[3, 4, 5, 7, 16]. Using primitives is limitative because users are restricted to the available basic shapes to make new objects. With sketching users draw the shapes of the new objects but the system has a limited number of shapes that it recognizes from drawings. Inflation is a very useful technique but the result is generated by algorithms and might not be exactly which artists wanted, requiring more manipulation. Although vertex manipulation allows users to define their objects with greater precision, today there are 3D objects with thousands of vertices and would be very time consuming to place them one by one. Boolean operations are a powerful tool to create more complex objects from simpler ones, like primitives. Works like Sketch [3] or Shapeshop [5] had the ability of taking parts of objects, performing a subtraction operation, but those operations were applied using a single object. MakeVR [16] is the only work, from the ones we studied, that was able to perform boolean operations between two existing objects. This manipulation between two objects offers all four CSG operations (union, both subtractions and interception) and gives a more visual feedback of the result because you can clearly see the volumes of both objects you are manipulating. Because we believe CSG operations are very powerful in 3D object modeling and because we didn't find many techniques to perform them between two objects in mid-air, we decided that this would be the focus of our work, techniques to perform boolean operations in immersive virtual environments.

Manipulating objects is an essential task to model 3D objects and environments. We studied different ways of interacting with the environments. Because finding the better way to manipulate objects in mid-air is a big challenge that is not the focus of our work, we chose to use a well known technique, the 6-Dof Hand [24] which mimics the natural way humans interact with physical objects in the real world. This similarity could potentially make the adaptation of users to the virtual environments easier.

The works studied regarding embodiment showed us that self-avatars change behaviours and the way users interact with virtual environments. In Steed et al. [27] we can see that some users moved their real body to prevent the virtual avatar from being hurt by the falling of a box. Lugrin et al. [29] found that realism can have a negative effect in some task performance and that non realistic avatars can achieve the sense of embodiment. Kilteni et al. [1] showed that the avatar look and clothing style can also have an influence in the way users behave in virtual reality. Acknowledging this behaviour influence, we want to find out if avatar has an effect on efficiency in modeling tasks and will compare the performance between an avatar and a non realistic representation of the user's hands.

Although there are some research works about modeling virtual 3D objects in Immersive Virtual Environments, it does not exist any technique to perform boolean operations in VR that uses gestural interactions instead of traditional input. In addition, there isn't any research work that studies the impact of an avatar in the performance of modeling tasks in Virtual Reality. Therefore, this dissertation's goal is to find an answer to those challenges.

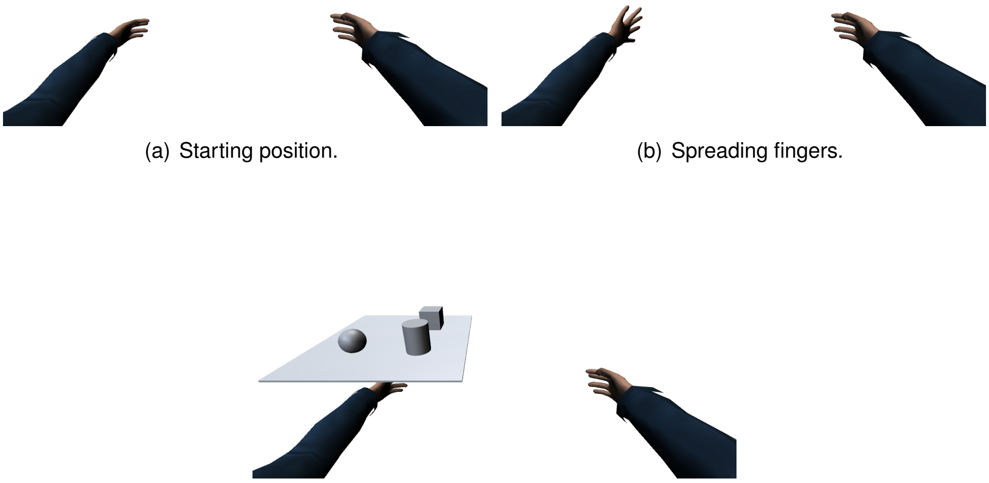
Chapter 3

Interaction Techniques for CSG Modeling in VR

We propose two new techniques to perform boolean operations in immersive virtual environments, the gesture based and the menu based approaches. In this Chapter we present those techniques and also a third one that is going to be used as a baseline later on. Boolean operations between two objects are not possible in our world because our laws of physics don't allow them. We daily interact with real objects, moving them from place to place. This types of object interaction are, from an early age, very natural for humans. We use those gestures as an inspiration for our first technique. In this gesture-based approach it is needed to interact with objects and with part of objects like we would in the real world, by grabbing them and moving them. Our second technique is based on menus with multiple options, which are a very common interface in most tools. We translated the menu to the virtual world by a 3D object divided in sections, which represent the buttons of a traditional menu. This sections offer immediate visual feedback to the user. The third technique is based on MakeVR [16], and uses two controllers with physical buttons as inputs. Although our main focus was to evaluate the different techniques to perform boolean operations, users needed objects in the virtual environment to test them. For that we also needed to have a creation tool and to allow users to manipulate the object's position, rotation and scale. All those available operations are explained on the following sections for each one of the approaches.

3.1 Gesture based approach

The gesture based approach is based on gestures that people do in real life. There is no way to perform boolean operations to two objects in our world, therefore there is no gesture that we can translate directly to the virtual world. We interact daily with objects and change their position and orientation often. To perform CSG operations we need to interact with objects and we based our interactions in those real world gestures which became natural and trivial to us because we use them so often. Users bring objects together as they would in the real world and they take the parts of the objects that are not needed to achieve the final result in the same manor.



(a) Starting position.

(b) Spreading fingers.

(c) Menu appears.

Figure 3.1: Opening creation menu

3.1.1 Creating objects

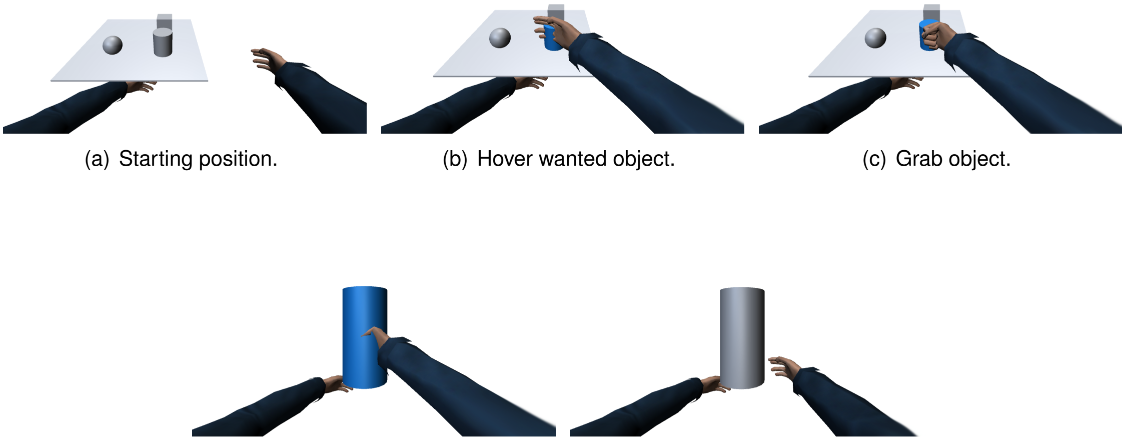
Creating objects is done by grabbing objects from a pallet [1]. This pallet is cast when the user opens its non dominant hand (Figure 3.1) and it follows that hand until it is closed again, with the same gesture. While the pallet is open, the user grabs and drags the object to the virtual environment by moving the hand without releasing it. An example of this can be seen in Figure 3.2.

3.1.2 Object manipulation

The user can manipulate the position and rotation of the object with the drag gesture in a similar manor to the 6-Hand technique [24]. That gesture can be done by any hand on one object at a time. To grab an object users have to close the hand while touching it, and to drag it you move the hand while its closed. The gesture ends when the hand is opened again. While dragging, the object follows the position of user's hand, and rotates in the same way that their arm does. These two manipulations happen simultaneously like they do on the physical world when a real object is moved by an hand [2].

The scale gesture is based on multi-touch screen scale gestures and also inspired by 6-Hand technique [3]. Scale is done by grabbing an object with a hand and then closing the other hand in a free space. With both hands closed, users can move them away from each other to enlarge the object, or move them closely to make the object smaller.

[1] Example video can be seen at: `https://www.youtube.com/watch?v=Bs3LWZjmk_c`
[2] Example video can be seen at: `https://www.youtube.com/watch?v=EblM8kUOpbo`
[3] Example video can be seen at: `https://www.youtube.com/watch?v=sR2dDlpld5w`



(a) Starting position.

(b) Hover wanted object.

(c) Grab object.

(d) Object appears.

(e) Release object.

Figure 3.2: Creating an object

3.1.3 Boolean operations

To execute the Boolean operations the user has to grab the two different objects on which the operations are going to be performed and drag them together until they are intercepting in the desired position and orientation, as depicted in Figure 3.6. The flow to start this operation is represented in Figure 3.3. When the objects are in the wanted position, the decision phase starts. Objects are divided in three parts: left object less right object, interception and right object less left object. The first is held in the left hand, and the last one is held by the right one. Interception stays at the center whichever movement hands make. The position where the user joined the objects is where the final object is going to be constructed. Users have to move away the undesired parts by draging the objects that they are holding. The right operation will be chosen based on the position where objects where released. The chosen operation can be confirmed by releasing the objects, opening the hands [4].

In the decision phase, the user can preview the result of the operation by dragging the objects out of the starting position, being possible to return them to that position while the hands are closed. There are 4 different decisions possible: Union, Intersection and two Differences (left object minus right object and vice versa). To choose the Union operation, both objects held by the hands have to remain at the starting position. To execute any Difference, the object that is going to be deleted has to be moved away, the other object stays at the starting position with the interception removed. The interception is obtained by moving away both hands leaving the interception at the starting position. To confirm the decision the

[4] Example video can be seen at: `https://www.youtube.com/watch?v=_qf0y5FE8FM`

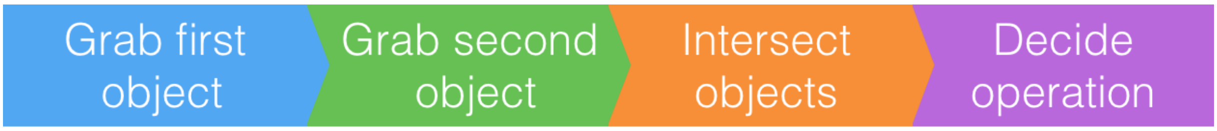
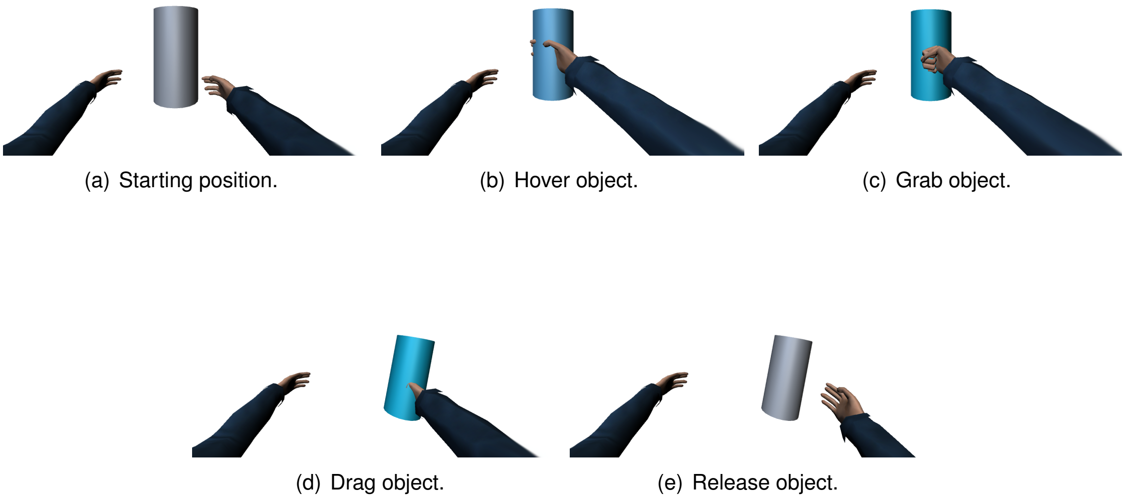


Figure 3.3: Boolean operations flow on gesture based approach.



(a) Starting position.

(b) Hover object.

(c) Grab object.

(d) Drag object.

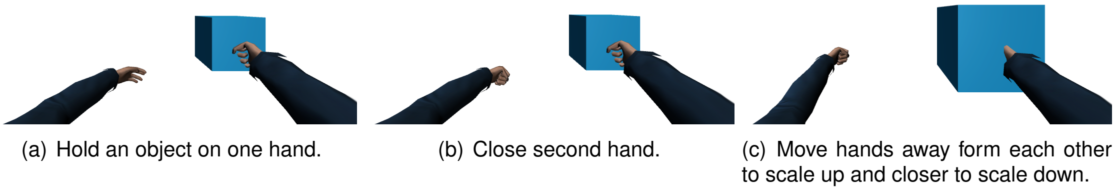
(e) Release object.

Figure 3.4: Moving an object

user opens both hands at the desired position. The hands can be opened individually at any time and they confirm only the final position of the object that they are holding. After opening the first hand, the user can still move the other object with the other hand. This gestures are represented in the Figure 3.7.

3.2 Menu based approach

The menu based approach is inspired by traditional 2D menus present in most WIMP interfaces. We bring this concept to the virtual world representing the menu with a 3D object divided in four sections, one for each selectable options, as seen in Figure 3.8. The menu gives visual feedback to the user by showing previews of the operations in each selectable section, corresponding to the matching operation. Also, by hovering those sections, an extra preview is shown on the original objects and the operations are chosen by grabbing the equivalent section. This menu is cast whenever the user is about to perform boolean operations and follows the non-dominant hand of the user so he can select the options with the dominant hand.

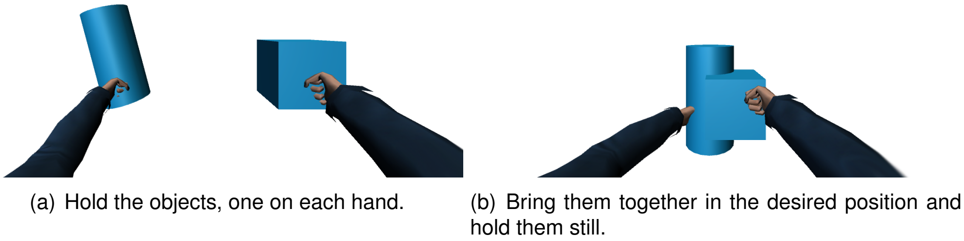


(a) Hold an object on one hand.

(b) Close second hand.

(c) Move hands away form each other to scale up and closer to scale down.

Figure 3.5: Scaling an object



(a) Hold the objects, one on each hand.

(b) Bring them together in the desired position and hold them still.

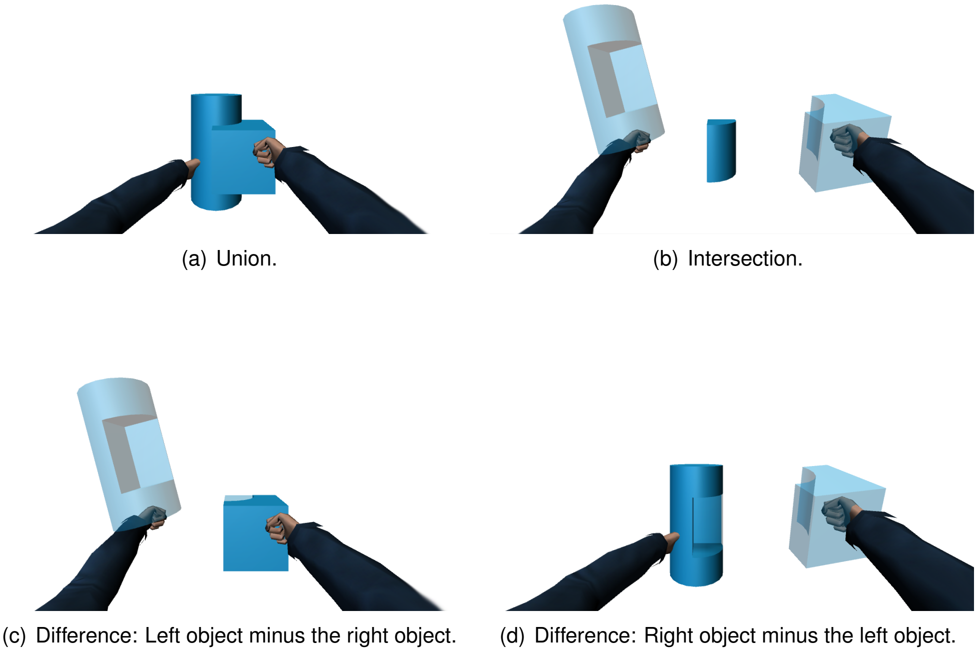
Figure 3.6: Starting Boolean operations

3.2.1 Creating objects

To create objects the user needs to first cast the creation pallet by opening the non-dominant hand. From the pallet the user chooses the wanted object by grabbing it and drags it from the pallet to create a copy. This is done in the same way as in the Gesture approach and an example can be seen in Figure 3.2.

3.2.2 Object Manipulations

Object manipulations in this configuration of the prototype is very similar to the Gestures configuration but has some differences. To move an object the user grabs the object and moves its hand in the physical world, the object follows that movement in the virtual world. While dragging, all the rotations of the arm of the user are applied to the object itself. The difference to the previous approach is that users can



(a) Union.

(b) Intersection.

(c) Difference: Left object minus the right object.

(d) Difference: Right object minus the left object.

Figure 3.7: Possible Boolean operations

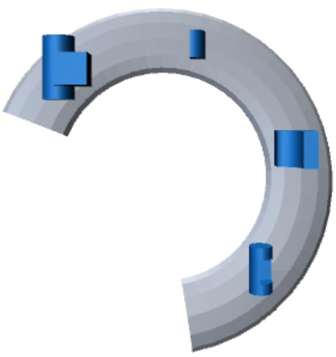


Figure 3.8: Menu with previews for a cube and a cylinder.

move two objects simultaneously, one in each hand, which would have triggered the Boolean operations in the gesture based alternative. Scaling an object is also done by grabbing an object and then closing the second hand and moving them away to scale up the object, and moving them closer to scale down. These manipulations can be seen in Figures 3.4 and 3.5.

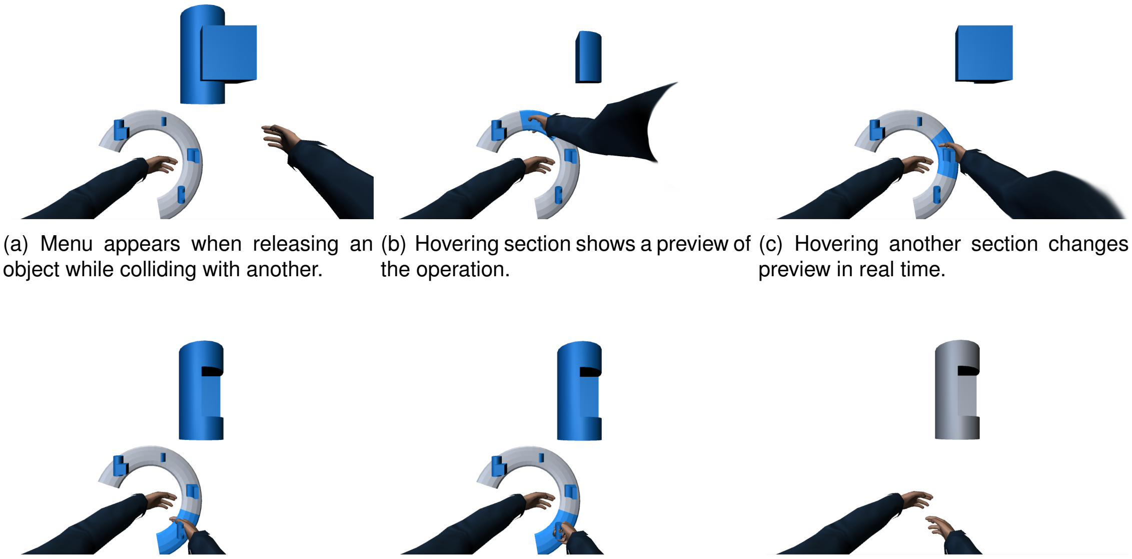
3.2.3 Boolean operations

The main difference from this approach to the gesture one is how the Boolean operations are performed, both the way they are started and the way the decision is made. To start them, the user has to previously have created the two objects between which he wants to operate. Then he grabs one (it's irrelevant which one) with the dominant hand and drags it until it is intercepting the second object. By releasing the object when it is intercepting another one, the Boolean operations menu is cast and shown on the non-dominant hand of the user, following its movement. This menu has four sections, four options, one for each operation available: union, intersection, difference of object a minus b and object b minus a. In each button there is a preview of the operation that it represents, a little object with the result of the operation on the chosen objects. By touching the menu buttons with the dominant hand , the original objects change to a preview of the result in real time giving the user another way of feedback. By showing this previews the user can choose an operation based on the results instead of choosing the operation itself. To choose which one he desires, the user grabs the correspondent section and the original objects give way to the result of the chosen operation [5]. This process is shown in Figure 3.9.

3.3 Controller based approach

The controller approach is inspired by the work of Jerald et al. [16], where users interaction is done with handheld controllers with physical buttons. This approach will work as our baseline to compare with the other approaches. In this approach each operation is mapped to a different physical button

[5] Example video can be seen at: `https://www.youtube.com/watch?v=QNUOHNyQaPE`



(a) Menu appears when releasing an object while colliding with another.

(b) Hovering section shows a preview of the operation.

(c) Hovering another section changes preview in real time.

(d) Preparing to choose difference.

(e) Close hand to pick operation.

(f) Operation is applied and menu disappears.

Figure 3.9: Performing boolean operations with menu based approach.

and that correspondence can be consulted in Figure 3.10. That mapping was decided based on how the controllers were designed to be used. The controllers are held vertically with one hand, having the thumb resting near the d-pad and the A button, making them very accessible. For the boolean operations we chose the d-pad because there were four different operations available, the same as the directions of the d-pad, and because the proximity of the buttons makes the difficulty of operation selection equal for all four boolean operations. The entire arm gesture, positioning and rotation still works like in the other two approaches, but the gestures to grab and drag the objects, to cast of the creation menu and to perform boolean operations are done by pressing buttons on the controller.

3.3.1 Creating objects

Objects are created in a similar manner to the other two prototypes but this time around using buttons for input. The creation pallet is toggled on and off with the Menu button. To grab the objects users need to press Grab button, and they have to hold it to drag the object, creating its copy.

3.3.2 Object Manipulations

Object position and rotation are still manipulated on the drag gesture on this variation of the prototype. To hold an object, users press Grab button while touching the object with its digital hand, and hold it while moving the hand to drag. The object, as observed from the other examples, follows the virtual hand and the rotation of the arm that it is holding it. In this variant users can use either hand to manipulate any object but can only use one hand at the same time to avoid confusion between manipulations and Boolean operations.

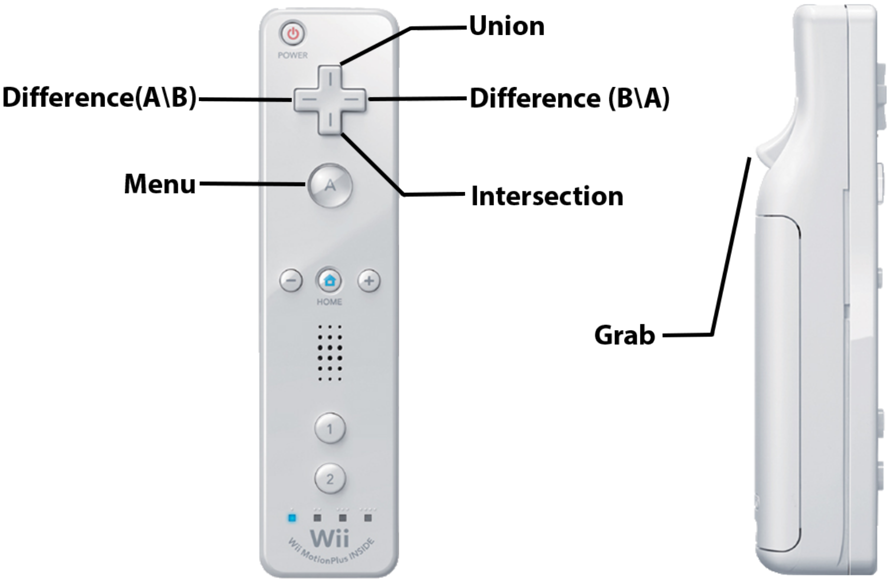


Figure 3.10: Controller key mapping.

3.3.3 Boolean operations

Controller Boolean operations have a similar start as in the gestures Boolean operations but have a different way to decide between the available operations: union, intersection and differences. Identically to the other two configurations users have to previously have created the two objects on which the operations are going to be applied. To start the operations, users need to grab each object with each hand and bring them together until they are touching each other in the desired position. When the hands stop moving, the objects position is locked and the decision phase begins. Users can now release the objects and move the hands freely. On the controller the user has another four different buttons, one corresponding to each of the Boolean operations. To apply any operations to the objects, the user needs to press the corresponding button on the controller on the decision phase. The key mapping for those operations can be seen on Figure 3.10 After pressing, the operation is done and the resulting object appears on the same position of the original objects [6].

3.4 Summary

Throughout this chapter we describe the two proposed techniques for performing boolean operations in Virtual Reality and also the baseline technique. We introduced the gesture based technique and also a technique based on menus that represents them with a 3D object divided in sections for each of the available options. We also present the baseline technique, which uses wireless controllers. We describe the interfaces and how to do all of the possible operations in every approach. We explain how to create an object, how to change the position of objects, how to perform a rotation, how to change the size of the objects and finally, how to execute boolean operations between two objects.

[6] Example video can be seen at: `https://www.youtube.com/watch?v=HsccEjugmVQ`

Chapter 4

Prototype

We built a prototype where we implemented all three techniques explained in the previous Chapter to test and compare them. We wanted to use an engine that was prepared for Virtual Reality and was able to integrate all the different modules and APIs required by the hardware that we were going to use, like Rift, Myo and Kinect, but also were we could try our ideas without having to build everything from scratch. This chapter explains every detail about the implementation, the architecture and all components of the prototype that we used to do a comparison between the techniques.

4.1 Architecture

The architecture of our prototype is represented in Figure 4.1. We chose to use the Unity engine because at the time of starting the implementation, it was the most accessible game engine and because all our hardware had official SDK with Unity integration which made it easier to build a first prototype that used all the models from all the hardware that we needed. Unity brought useful features to our project like the Scene Module and the Render Module but also features like collision detection and physics simulation that we turned off because we needed objects to be able to intersect each other and to float in middle air on our solution. To calculate the new meshes when performing CSG operations we used a library built for Unity by Andrew Perry. We built a Gesture Manager that decided which gesture the user was performing and did the corresponding action and a User Representation Manager that animated the avatar or hand representation accordingly to user movements. The prototype registered all actions performed on tests and also had the ability of saving result meshes to Wavefront's OBJ files. Those modules are explained in detail bellow.

4.1.1 Gesture Manager

The Gesture Manager gathers all the information collected through the sensors to decide which is the gesture that the user is performing at each time. It uses the body tracker position to know when the user is touching objects, and which one it is touching and to know the distance and direction to translate when dragging objects. Gesture Manager also needs the information given by the Myo armbands to know what

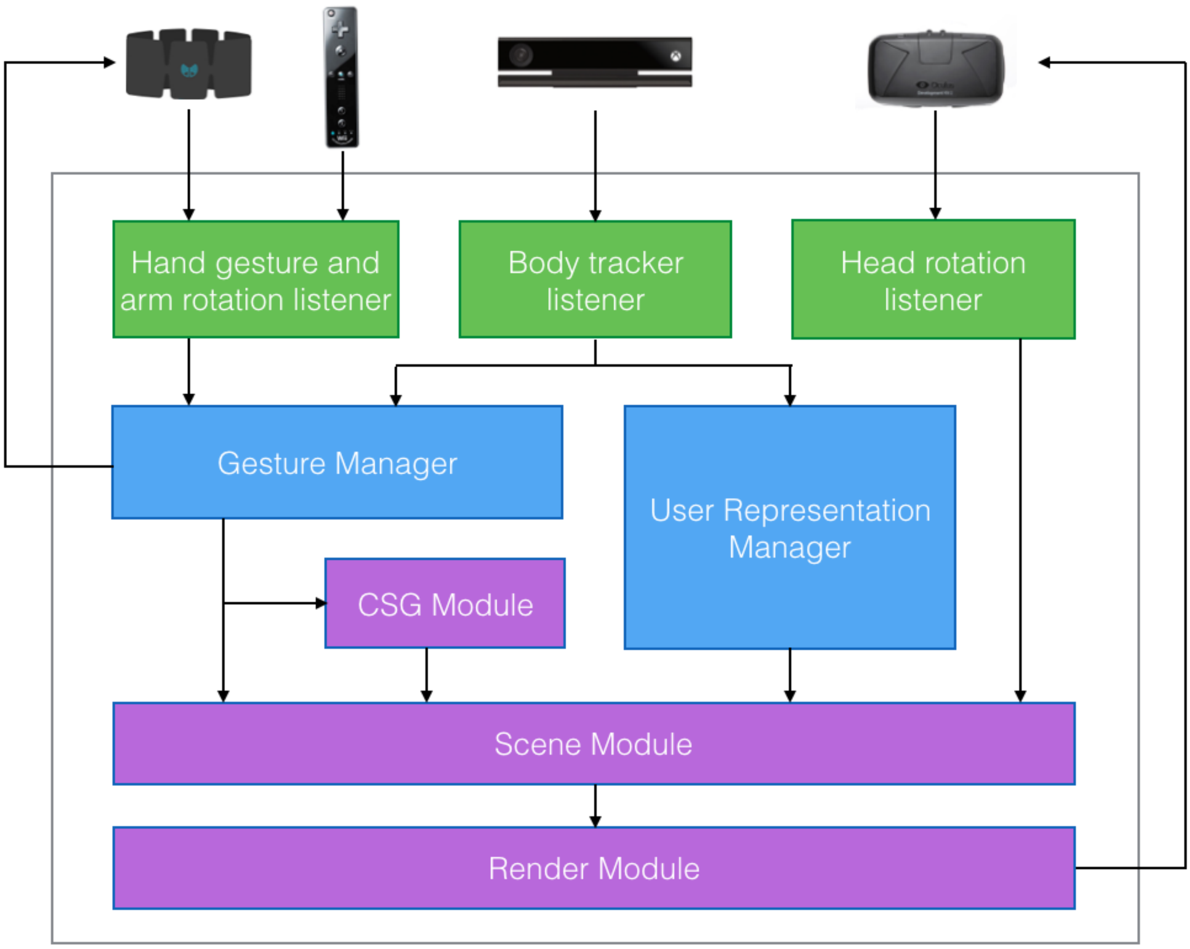


Figure 4.1: Prototype architecture

gestures the user is doing with its hands and the rotations of the arms. That rotation is needed to rotate objects while dragging them. The hand gestures are used to know when user is grabbing a object that is touching and to know when to open/close the creation menu.

This manager behaviour is implemented in a State design pattern, each state representing each gesture: Idle, Dragging, Scaling, Boolean Operation, Toggle menu visibility and Create Object as seen in Figure 4.2. In each frame, the sensor data was sent to the current state and it decides if there is the need to change state or not. As shown, the starting gesture is Idle because users are not doing anything when the prototype starts running. From the Idle gesture, users can open the creation menu by spreading their non-dominant hand fingers, grab an object if there is any already created or create an object from the menu if it's already open. While dragging an object, There is the possibility of starting a Scale operation by closing the secondary hand without touching any object and spreading them. Or, in the case of Gesture based approach and Controller based approach, it is possible to start Boolean operations by grabbing a second object with the second hand. Most of this states return to Idle gestures when finished allowing users to start the following gesture.

The Boolean Operation gesture has slight variations in different approaches but at its base there is the same algorithm, a sequence of three steps: gesture preparation, decision phase, application

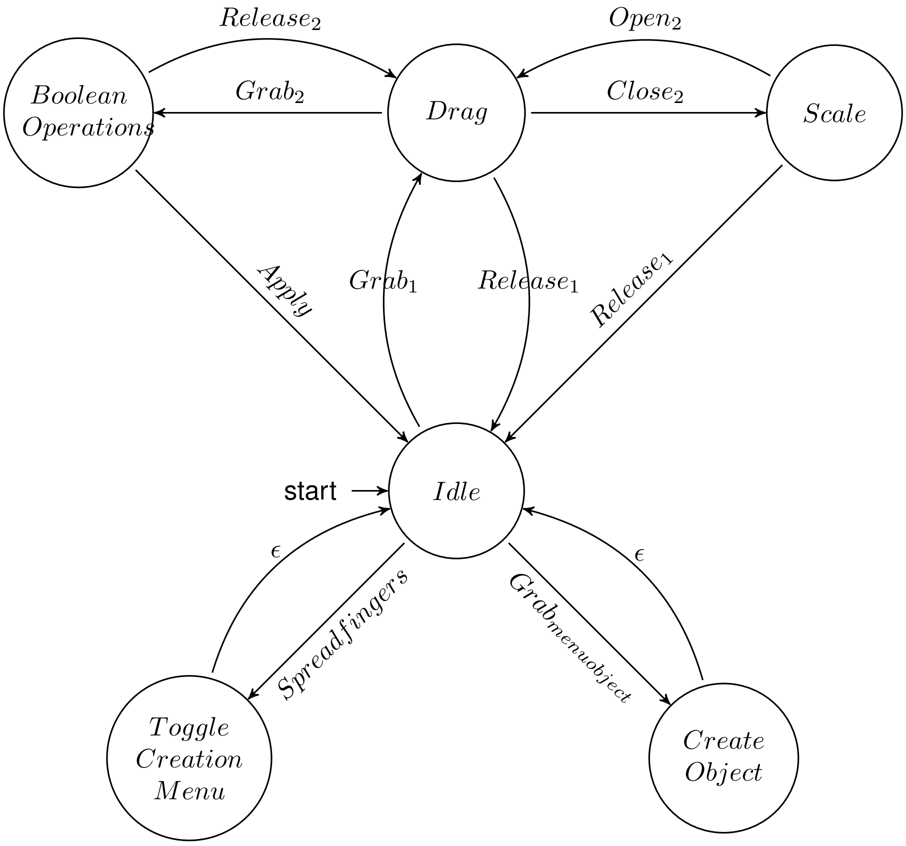


Figure 4.2: Gesture state machine

of the operation. The gesture preparations is where users choose both objects that are going to be affected by this operation by grabbing them with each hand (Gesture based and Controller approaches), or by releasing an object while it touches another (Menu based approach). Decision phase is where you perform the needed steps to choose between the available operations: union, differences or intersection. Application of the operation ends the boolean operations, applies the selected operation on the two objects and generates a new one with the proper result. Although the decision phase of the Menu and Controller versions of the prototype are a little bit basic because they only need to wait for the user to choose a virtual or physical button, in the Gesture version of the prototype, it's a little bit more complex and is also designed like a state machine and it's states can be seen in Table 4.1.

4.1.2 User Representation Manager

There are two possible representations on our prototype to test the influence of a self-avatar on the performance of our task. For that we created two different experiences, one with an avatar, and another without an avatar to test each one with half of the participants (Figure 4.3).

The avatar is scaled to match user's height and we animate it accordingly to the skeleton that the Kinect calculates, which is based on user movement. Those calculations are approximations but still give us a very close movement in the virtual environment to the movement we do in real world. Kinect doesn't provide hand rotation values so we use the rotations of the Myo armband to animate the avatar's hands and also the hands representation of the version of the prototype withouth an avatar. Hand gestures are

State	**Input**						**Feedback**		
	Drag$_l$ away	**Drag$_l$ back**	**Drag$_r$ away**	**Drag$_r$ back**	**Open Left**	**Open Right**	**Left Object**	**Interception**	**Right Object**
E1	E2	-	E3	-	E4	E5	Opaque	Opaque	Opaque
E2	-	E1	E6	-	E7	E8	Translucent	Translucent	Opaque
E3	E6	-	-	E1	E15	E13	Opaque	Translucent	Translucent
E4	-	-	E15	-	-	E16	Opaque	Opaque	Opaque
E5	E8	-	-	-	E16	-	Opaque	Opaque	Opaque
E6	-	E3	-	E2	E9	E12	Translucent	Opaque	Translucent
E7	-	-	E9	-	-	E10	Deleted	Translucent	Opaque
E8	-	E5	-	-	E10	-	Translucent	Translucent	Opaque
E9	-	-	-	E7	-	E11	Deleted	Opaque	Translucent
E10	**Final State : Difference ($r\backslash l$)**						**Deleted**	**Deleted**	**Opaque**
E11	**Final State : Interception**						**Deleted**	**Opaque**	**Deleted**
E12	-	E13	-	-	E11	-	Translucent	Opaque	Deleted
E13	E12	-	-	-	E14	-	Opaque	Translucent	Deleted
E14	**Final State : Difference ($l\backslash r$)**						**Opaque**	**Deleted**	**Deleted**
E15	-	-	-	E4	-	E14	Opaque	Translucent	Translucent
E16	**Final State : Union**						**Opaque**	**Opaque**	**Opaque**

Table 4.1: Decision phase of Boolean Operations in Gesture based approach. l) object held in left hand; r) object held in right hand.

also captured with Myos armbands and that information is used to animate the fingers of the avatar or to change the material of the hand representations like shown in Figures 4.3(d) and 4.3(e).

4.1.3 CSG Module

This module is an implementation of the CSG operations on Unity. Originally this module was built with Javascript by Even Wallace [1] and later ported to Actionscript 3 by Tim Knip [2]. This version is a direct port from the AS3 version to C#/Unity and it was done by Andrew Perry [3]. This algorithms still needs work of optimisation and some bugs to fix. It worked in our case because we use simple primitives with low polygon count. Some errors in mesh calculation occurred but they were negligible because its impact on end results were minimal. We chose this module because it was built for the same engine that we use in our solution which made its integration faster and easier than with other solutions.

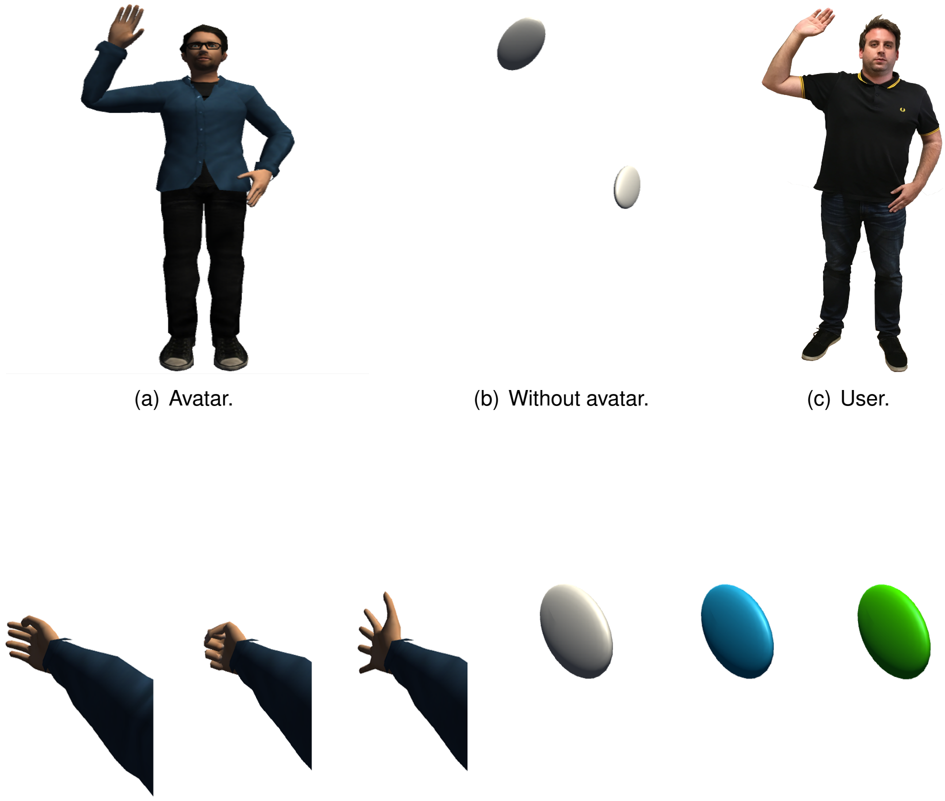
4.2 Setup

Our setup tracks entire body movement and hand gestures while giving an immersive experience and is present in Figure 4.4. Using Microsoft's Kinect V2 we were able to get the skeleton of the user to be represented in the virtual world. Although it captures the position for every member, hand gesture and rotation is not achievable with this camera.

[1] https://github.com/evanw/csg.js

[2] https://github.com/timknip/csg.as

[3] https://github.com/omgwtfgames/csg.cs



(a) Avatar.

(b) Without avatar.

(c) User.

(d) Avatar hand gestures. From left to right: Idle, Closed, Spread

(e) Hands representation gestures. From left to right: Idle, Closed, Spread

Figure 4.3: Difference between having avatar and not having an avatar.

Instead, we used a Myo armband in each arm to perceive gestures of the hand as well as the orientation of each arm. For the controller version of our prototype we used two Nintendo Wii Remote controllers conected to our PC via Bluetooth.

Using a virtual reality headset by Oculus, the Rift DK2, allowed us to give the users a visual immersive experience.

4.3 Summary

This chapter explains the implementation of our prototype. Here we give an explanation on the engine we used and all the modules we either integrated or built ourselves to make our final solution. We explain how the most important modules work, the Gesture Manager, the User Representation Module and CSG Module. We also explain how they work with each other, explaining the overall architecture of the prototype. We also talk about all the hardware required to achieve this solution and the role they have in our implementation.

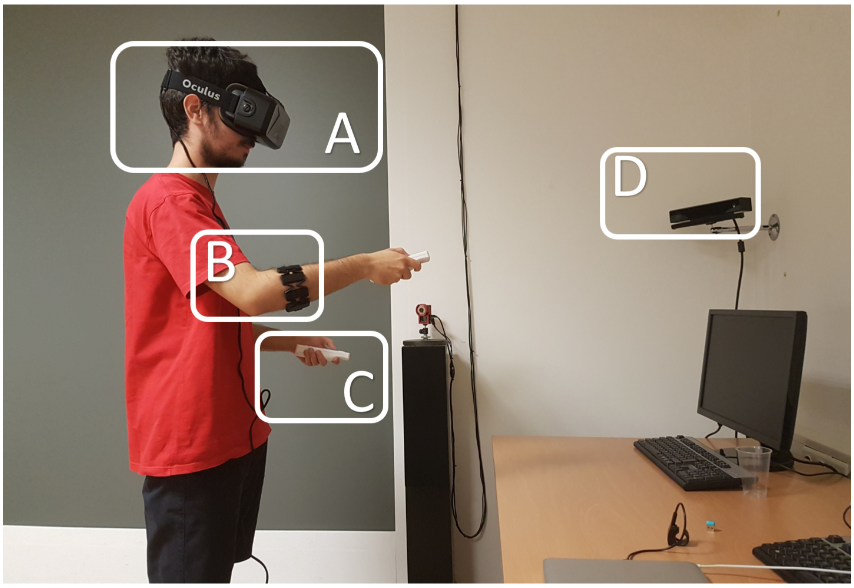


Figure 4.4: Setup of our prototype. A-Rift DK2; B-Myo armband; C-Wii remote controller; D-Kinect v2.

Chapter 5

Evaluation

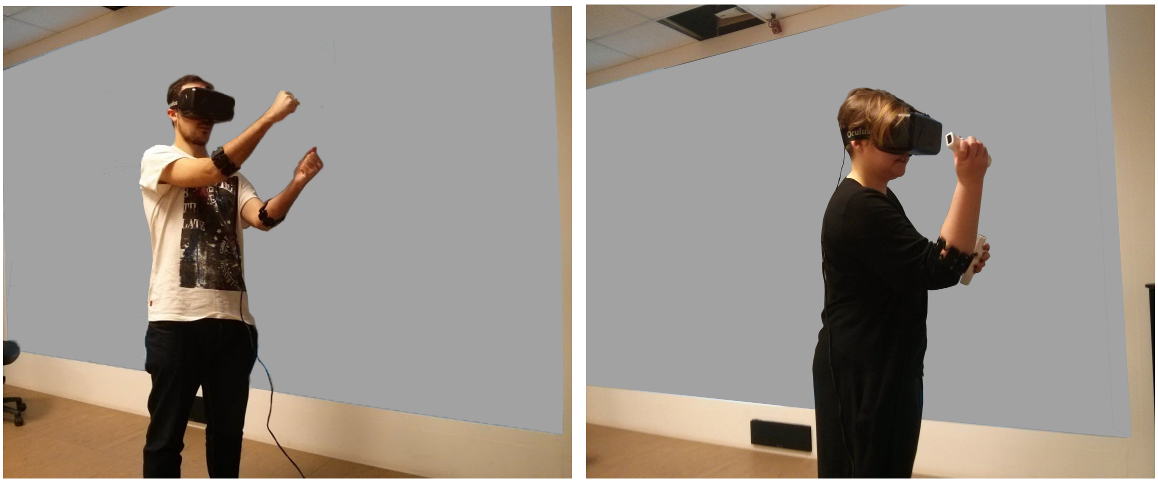
In this Chapter we compare the three techniques presented in section 3, and try to find which one is better suited to use as a 3D modeling tool, focusing on Boolean operations between two virtual 3D objects. This comparison was done by conducting a user evaluation. In that evaluation we asked users to perform a task with each of the different techniques and measured their efficiency and their opinion through an inquiry. We also tested self-avatar influence on user efficiency by executing the tests with half of the subjects with an avatar as a body, and the other half with markers for virtual hands to see how that influenced the results. Throughout this chapter we present the evaluation procedure. We start by describing the user tests, followed by the evaluation parameters and the metrics used to measure them. Finally we present and discuss the results, as well as present some examples of models done by users.

5.1 Participants and apparatus

The tests were performed in our laboratory in the Taguspark campus of Instituto Superior Técnico, also known as Lourenço Fernandes Laboratory. The room has restricted access which produces an isolated, clean, calm, and controlled environment without external disturbance. This room is fully equipped with all the hardware needed to carry out the tests: Kinect cameras, Myo armbands, Oculus Rift DK2, Wii Remotes and a projection screen for the introductory presentations. We performed the tests with 24 people (3 female), with ages from 21 to 30 years old. Most were students in higher education (75%), while the remainder had already achieved a MSc degree. Less than half (42%) had never experienced Virtual Reality before, only 4 people (17%) had never experienced 3D gesture tracking systems like Microsoft's Kinect, Playstation Move or Wii Remote Controllers, and only 3 people (12.5%) had never experienced modeling tools.

5.2 Methodology

Tests were carried out individually. An example of users performing the test can be seen in Figure 5.1. For each one of them, we gave a small presentation explaining how the prototype works, a script with



(a) Participant using the gesture and menu based approaches.

(b) Participant using the controller approach.

Figure 5.1: Users performing the task

instructions and asked them to do a task. Every user performed the task three times, one for each technique: gestures, menu, controller. The order of the techniques they used was changed in every test, going through a total of 6 different possible orders four times. That configuration guaranteed that the times for each approach wasn't influenced by the order of the tests or the knowledge the user had of the task. To test self-avatar efficiency, half of the tests were done with an avatar representing the whole body of the user on the virtual world, and the rest were done with two round objects representing only the hands. The tests had the following procedure: We started by giving a brief presentation explaining the prototypes, boolean operations and the task. To assure that the presentation was equal for every user, a keynote was prepared with example videos demonstrating how to perform the different manipulations. The objective of this presentation was to give the users all the knowledge about the project and the task so they would be able to do it by themselves. After the presentation, the user would perform the task. For each technique the test divided in four different parts: watch a video explaining that specific technique, experiment the prototype with that technique, perform the task, answer a little inquiry about that task. This videos were separated from the main presentation to avoid confusion, we explained each of the interfaces only when the user was about to use them. When testing the prototype, the test supervisor made sure that the user did all the possible object manipulations: translation, rotation, scale and at least one of each boolean operation. That phase had an estimated duration of 3 minutes and the users could ask for help and clarify all the doubts they had. Past experimenting, the users did the task, without any help from the test supervisor. This phase was filmed for evaluation purposes and had the estimated duration of 10 minutes. After finishing the task, users were asked to answer some questions about that specific technique. The estimated time for the test and for each individual part are presented in Table 5.1. In the end, after performing the task with all three techniques the users answered some more global questions, comparing all three techniques and their version of body representation.

Stage	**Duration**
Experiment Introduction	5 min
Technique Description	2 min*
Training Period	2 min*
Task Execution	10 min*
Questionnaire Completion	3 min*
Total	60 min

Table 5.1: Methodology estimated stage duration. (* per technique)

5.3 Task

The task that users were asked to do is to replicate a 3D model with our prototype. The model is represented in Figure 5.2. This model was inspired in a real building, the Casa da Música metro station in Porto, Portugal, which was designed by the Portuguese architect Eduardo Souto de Moura and can be seen in Figure 5.3. The result is achieved by doing Boolean operations between the available primitives. Users could create cubes, cylinders and spheres. The top of the model cold be done both by intercepting two cubes and by doing a difference between two cubes, leaving the choice to users to do as they prefer. The cylinder base is obtainable with the same operations but using the cylinder primitive instead. The hole in the middle has to be done by subtracting a cylinder to the object, but for that to work, the top and bottom parts need to be united first. This sequence obligates the user to use all of the Boolean operations present in the prototype, helping us figuring out how easy and fast it was with each kind. Examples of results done by our test participants by performing this task can be found in Figure 5.4.

5.4 Results and Discussion

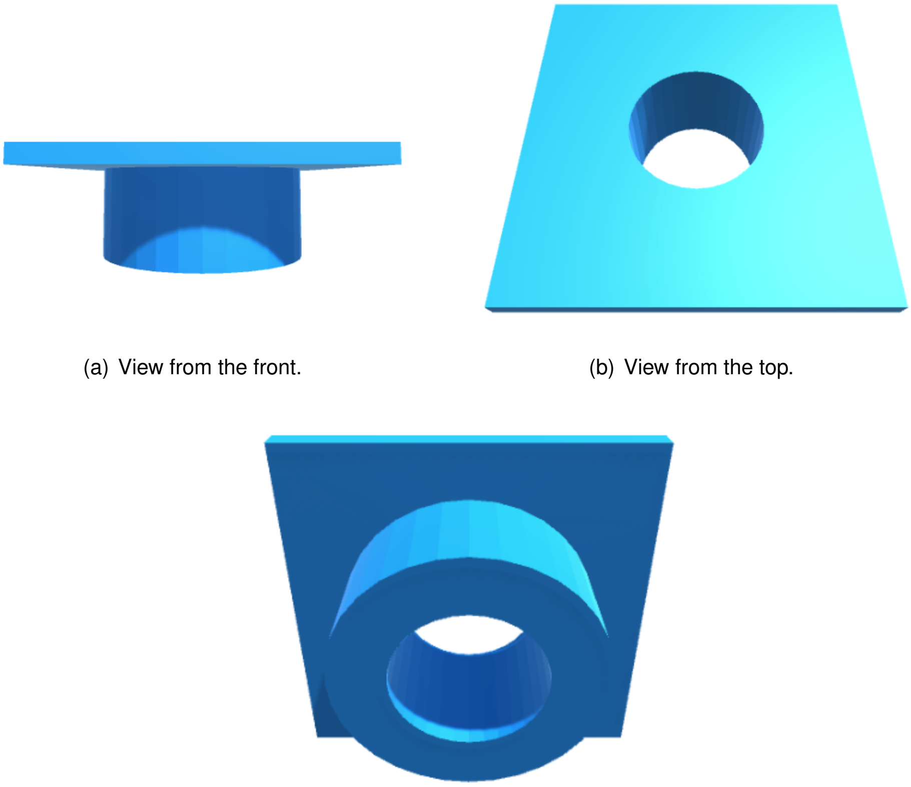
To evaluate the difference between the three techniques to perform CSG operations in Virtual Reality, we collected objective and subjective data in the form of logs and inquiries respectively. To test data normality we used the Saphiro-Wilk test. With the normal distributed data we found if there were significant differences with the ANOVA with repeated measures test with a Greenhouse-Geisser post-hoc test and a Bonferroni correction (corrected sig. = sig. x3), when data wasn't normal distributed we used Friedman non-parametric test with Wilcoxon-Signed Ranks post-hoc test, also with Bonferroni correction. To test the significant differences in embodiment, we used the independent t-test.

5.4.1 Objective Data

With efficiency evaluation of the three techniques in mind, we collected the times (in seconds) of users performing the task with each one. Our logs kept the total time (Figure 5.5) of execution of the task as well as individual logs for each kind of operations like object manipulation, Boolean operations, object creation and idle (Figure 5.6).

We found statistically significant differences between approaches in the total completion time of the task (F(1.874,37.481)=5.101,p=0.012). With the pairwise tests comparing total time, we found that there was statistically different between the Gestures and Menu alternatives (p=0.023). This significant difference in total time lead us to test the time for individual type of manipulations to find where that difference was more prominent. We found with this that there was no significant difference in the execution of Boolean operations. Differences were felt in manipulation operations(F(1.718,36.076)=6.499, p=0.006) with post-hoc test revealing that there was statistically significant difference between Gestures and Menu approaches (p=0.062) and between Gestures and Controller approaches (p=0.018). Idle times had also a significant difference (χ^2(2)=7.128, p = 0.028). We ran Wilcoxon-Signed Ranks to find that the Gesture and Menu approaches are statistically different in Idle time (p=0.057).

Although we found no significant difference in execution of boolean operations, we have to recall that the gesture based and menu based approach used the Myo arm bands as input in contrast to the Wii remote controllers used in the controller based technique. Myo arm bands gave us the enormous vantage of having free hands but also brought inaccuracy. Wii remote controllers worked as expected every time, Myo arm band, on the other hand, detected wrong gestures which introduced errors and lead to user mistakes and ultimately to frustration and worst execution times. This observation leads us to think that this interfaces could have advantages when compared to the controller approach if the hand-gesture detection hardware was better. The differences felt manipulation-wise could be caused by users being faster with the controllers because of its higher accuracy, or because users tend to better



(a) View from the front.

(b) View from the top.

(c) View from the bottom.

Figure 5.2: Objective model

(a) View from the front.

(b) View from the side.

(c) View of the ceiling from the inside.

Figure 5.3: Casa da Música metro station in Porto, Portugal.

prepare the position of objects to execute boolean operations when using myo armbands. To better find out, we did some pairwise tests to find where those differences were felt. Idle times differences could also be result of users having to think more in some approaches that others or even by users having to rest more, or being more frustrated by the errors caused by myo arm bands.

By testing approaches pairwise we can see where there are the differences. Total times between gesture based and menu based alternatives are influenced by the difference in manipulation. In spite of the fact that object manipulations gestures are exactly the same in gesture based and menu based, the differences in the boolean operations are enough to influence that significant difference in times. On the gesture based approach, because users had to grab objects with both hands, one on each hand, and because there was a Myo armband in each arm, users tended to make more mistakes. To compensate that, users carefully positioned objects to better prepare boolean operations, resulting in bigger times of manipulations. Also, on menu based approach, it was easier to cancel boolean operations and re-adjust object position, which also made users more careless on manipulation gestures, and therefore faster. The difference between gesture based and menu based approaches could be justified by the efficacy of the Wii remote controllers when compared to the Myo armband gesture detection. That advantage contributes to more efficiency in executing the operations, allowing users to perform them faster and with less preparation and thought, being that a key factor to the difference in Idle time as discussed before.

Our embodiment test was done with an independent-samples t-test and we found that not only there is a significant difference time between aving and avatar or not in total times (t(22)=-3.003, p=0.007), there is also a difference in each approach of the prototype between having an avatar or only hand

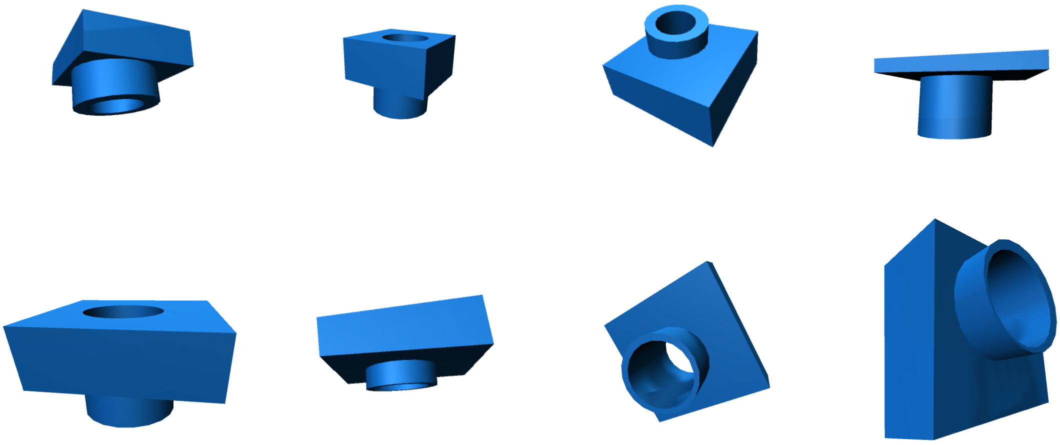


Figure 5.4: Example models created by users on our tests.

representation: gestures (t(22)=-1.792, p=0.087), menu (t(22)=-1.72, p=0.099) and controller (t(21)=-2.927, p=0.008).

These results tell us that the self-avatar had a negative impact on the performance of users executing our task. This conclusion disagrees with the studies and theories about embodiment and embodiment that we referred in Chapter 2. We have some some suggestions to help us understand why we achieved this results.

Avatars are great to help users feel present in a virtual world where they need to see their scale, and to move through the immersive virtual environment. While our work gave the possibility of movement through the room, most users stayed in the same place where they started the test, moving only their arms to interact with the objects. Therefore the advantages of seeing our virtual size and where we are in the world in relationship to the ground were not felt at all.

The only real part of the avatar that was used by every participant was the hands, because they had to use them to interact with the objects. When the avatar was not present, there still was a representation of the hands, by an oval object, which gave the users enough perception of position to operate the objects. As a matter of fact, the change of state of those oval hand representation was more noticeable because they changed colors, than on the avatar counterpart, where the change was only felt by the position of the fingers. Our participants could have perceive the change of color faster than the change of position of the fingers. So, although there was no avatar, there was still the same amount of feedback present.

Avatar arms potentially occluded the vision of users by being in front of objects blocking them and the world, affecting overall perception of the scene. Lastly, avatars can have raised wrong expectations in users. By having arms and hands similar to theirs in the real world, they create the idea that they move the same one that real ones do. Virtual movements were constructed based on the capture of a Kinect v2.0 camera which introduces some imprecision and jitter. This might cause some frustration in users that wanted to move the arms like real ones. Participants with ovals only had approximations of the hand, and did not expected them to move like real hands because they didn't look familiar, adapting faster to the jittering movement.

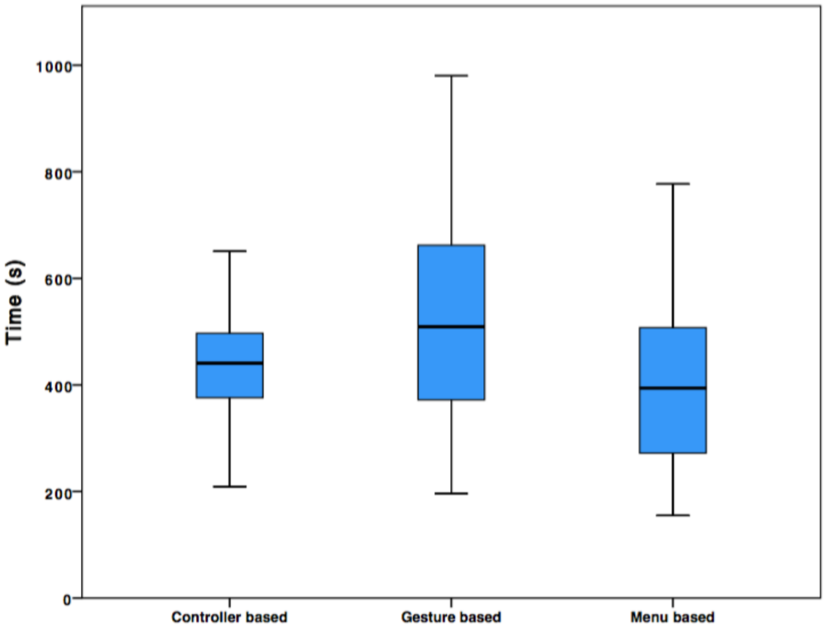


Figure 5.5: Test total times for the different techniques

5.4.2 Subjective Data

We collected subjective data to help us identify the user's opinion, their preferences and other useful information that is not collectible through automatic logs. To compare the different approaches we asked the users how difficult and fun was to perform the task with each of the prototypes. To find how different was to execute the operations we asked them how hard was to recall how to do them and how hard was to actually perform them. Those preferences are represented in Table 5.2.

To test if there was a significant difference we used Friedman non-parametric test with Wilcoxon-Signed Ranks post-hoc test and with Bonferroni correction. We found statistically difference in every parameter: difficulty , fun, recall and execution.

Although these results represent the overall opinion of participants, we do think that their judgement was heavily influenced by the challenges that the low efficacy of the input introduced. Kinect input is an approximation and because of that it has some very noticeable jumps from frame to frame. Myo armbands sometimes failed to proper recognize the right hand gestures that testers were performing even though they were calibrated for each one of them. This jitter and imprecise controls lead to frustration that was more felt in one particular approach than others.

In terms of difficulty, we found significant difference (χ^2 (2)=7.786, p = 0.02) between approaches and with the pairwise tests that there was statistically difference between the gesture based approach and the menu based approach(p = 0.012). This difference was felt because the menu approach gives more immediate feedback. As soon as the boolean menu shows up, users are presented with the results of every boolean operation, making more direct which one they have to choose when compared to the thinking or experimenting that they have to do in the gesture based approach. Also, because they didn't have to hold the objects in the menu approach, they could re-adjust the position of objects instead of choosing an operation. This difference could also be influenced by the performance of the

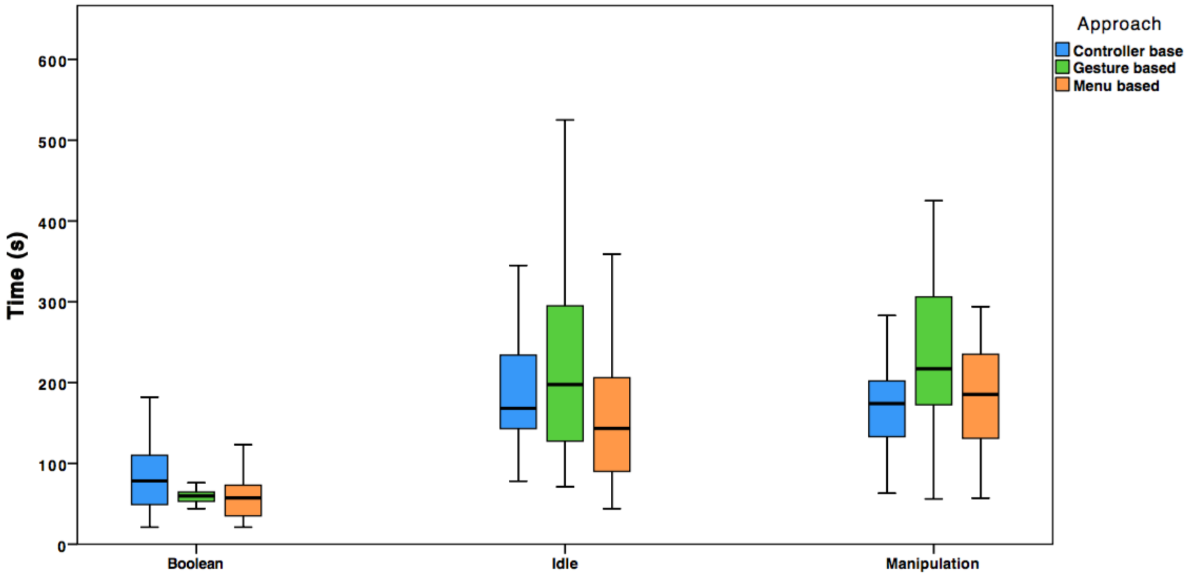


Figure 5.6: Pairwise comparison of times for the different techniques separated by operation.

input hardware. To perform boolean operations in the gestures approach, it was necessary to hold the objects with both hands and for more time than on the menu approach. Myo armbands recognized, often, hands opening when they were in fact still closed which had a big impact on the final result of the operation. The preview was done while holding the objects and because users had fear that the armbands would wrongly detect that they opened their hands, they wouldn't depend on that preview to execute the operation. They often preferred to think before in what they have to do. On the menu based counterpart of the prototype, users only had to hold objects with one hand, therefore less possibility for hardware failure, and preview is done after positioning object, without holding objects which gave testers more confidence to play with. Also, there is the possibility of re-adjust object position in the menu based approach and that fact makes the Myo wrong detections less significant because they could be corrected when it happened. That adjusting component also compensates the jitter introduced by Kinect which is more felt on gesture based approach because it lacks that possibility of readjustment.

Fun factor also had significant difference (χ^2 (2)=6.123, p = 0.047) with pairwise tests reveiling differences between gesture based and menu based approaches (p = 0.42). All those problems introduced by the hardware were very frustrating to the participants when they were performing the task with the gesture based approach. The jitter when positioning the objects and the lack of re-adjusting their position made the errors more impactful on the final result and that had a very negative impression on users enjoyment. For these reasons, the gesture based approach was less fun that the menu based for most of participants.

There were significant differences for Recall(χ^2 (2)=30.104, p $<$0.05) and also differences between every approach: Gestures and Controller (p = 0.006), Gestures and Menu (p = 0.021) and Controller and Menu (p $<$0.05). Recall is the only parameter were we can compare directly all three variations of the boolean operations, because it has the only results where we found statistically difference between all approaches in pairwise tests. Menu based is the easier to recall not only because it showed the result

	Gesture based	**Menu based**	**Controller based**
Easiness*	4.5 (2)	5 (1)	4.5 (1.75)
Fun*	5 (1)	6 (1)	5 (1.5)
Recall*	5 (2)	6 (1)	3 (2.5)
Execution*	4 (2.75)	5.5 (1)	4.5 (2.75)

Table 5.2: Participants opinion for each technique (Median, Inter-quartile range). * means that there were statistically significant differences for that factor.

of each operation in the correspondent button, but it also showed the final result on the actual objects by hovering the buttons with the dominant hand. Users could choose the boolean operation needed to achieve the wanted result without really knowing which operation they were choosing. Gesture based were the second easiest because they also showed the preview of the result of the operation that users are choosing by changing the material of the parts that were going to be deleted to a translucent blue. With the low efficacy of the hardware used, participants were not as comfortable using this preview as they were in the menu based approach. This also helped to the fact on making menu easier than gestures. Controller is the last approach in this recall ranking because it doesn't possesses any kind of preview feature. Participants had to remember which physical key they had to press corresponding to each of the operations. There was an image on the virtual world that they could consult in case they forgot. Therefore we could rank the approaches in terms of recall:

1. Menu based approach
2. Gesture based approach
3. Controller based approach

We found statistically significant difference in terms of execution (χ^2 (2)=11.541, p = 0.003) and the Wilcoxon-Signed Ranks post-hoc test showed us significant between Gestures and Menu for (p = 0.03). With our results we can say that users found executing the boolean operations easier in the menu based approach than on the gesture based approach. This is consequence of the stronger feedback menu gives to the users with two types of previews. Before choosing an operation, users have a preview of all operations on the sections of the menu, and there is also the possibility of hovering those sections to have a real time preview on the actual objects. Users had to think and experiment less with menu based approach than with the gesture based approach and they could also re-adjust the position of objects before making a selection. That difference was also influenced by the hardware as discussed before. The frustration felt by the users made them fear the preview feature on the gesture based approach which made it more difficult than we thought it would be, in theory, if the input hardware was more reliable. Also, on the gesture based technique, even though participants didn't experiment as much with the four different possibilities, they would still have to perform the gesture of that particular operation which was prone to input errors making the task more difficult.

When asked which technique they liked most and would like to see in a CAD like application in VR, 50% of people answered the menu-based approach, 37.5% chose gesture-based approach and the left 12.5% chose the controller based approach.

5.5 Summary

In this section we presented the process that we used to evaluate our implemented techniques. We presented the environment in which we performed the tests and we drawn a profile on the participants of our tests. Next we talked about the methodology we followed to carry out the tests, explaining every step of them. We mentioned the presentation we did to users explaining them the techniques, the task and the time reserved for the inquiry. We presented the task, the objective model we asked the users to reproduce and the inspiration that lead us to that model. In the end we present the gathered results and data and evaluate them to check for statistically difference between the different approaches, based on the time and user's opinion, and we do a brief discussion of the results of those comparisons. We found no statistically significant difference between the three approaches in boolean operations performing time. Users preferred the Menu based approach and found it more easy to use, more fun, easier to recall how to perform boolean operations and easier to execute them. We also found that self-avatar had a negative impact in efficiency while performing the tasks with every approach.

Chapter 6

Conclusions

The use of computer generated three dimensional content has grown in the last years. This growth can be seen in the increase of animated movies being done, movies with CG environments and characters, 3D video-games that require a lot of assets, TV advertisements that are completely made with 3D models, digital mock ups of buildings and vehicles, human models used in medicine and in many other uses. These contents are made by artists, architects and even by self learnt modelers. The available software for the creation of 3D content are designed with 2D input and output in mind. Content creators usually work with a traditional mouse or a drawing tablet which only has two degrees of freedom. This is a limitation to create 3D objects because manipulations require three degrees of freedom, one for each axis. Converting 2D input in 3D manipulation is not trivial neither to implement in those type of software, neither for creators to use on their programs. These applications usually settle for the option of changing between axis and manipulate one at a time or typing the precise values by hand. Despite working, it's a counter productive solution because requires users to waste a lot of time changing between axis and modes, translation and rotation, and precise values sometimes are not the most eye appealing.

Virtual reality has seen a growth in interest because the late advances in display resolutions and refresh rates, as well as the advances in motions sensors made possible to create new headsets that are comfortable for the human eye to use. In the present year of 2016, three headsets were launched: the Oculus Rift, the HTC Vive and the Playstation VR. All three headsets were designed to the common people and not only for professionals. One proof of that is the main tool used for advertising and selling the hardware: videos and games. The possibility to be immersive in new virtual worlds attracted the interest of the masses. This technology can be used to empower professionals of 3D content creation. Users of this headsets perceive 3D models in 3D as they would in the real world which gives them a more accurate feedback about the visuals of their work. Combining virtual reality headsets with the latest motion sensors and cameras like Microsoft's Kinect gives the users both input and output in 3D. This is an advantage face to the traditional input because it has the same degrees of freedom as the manipulations usually performed in 3D modeling. While in Virtual Reality, users can not see their own bodies because the headsets isolate their vision. To give users the notion of position in those immersive environments, it is necessary to find strategies to represent their body, or at least, the part of the body that is needed to perform the task in hands.

Studying the related work in how to perform boolean operations we found that there weren't many techniques designed to execute those type of manipulations in immersive virtual environments. Because we believe that virtual reality can improve the productivity of 3D content creators, our goal was to develop new techniques to perform CSG operations in immersive virtual environments. We also wanted to find if an avatar could improve the efficiency of users in modeling tasks like ours.

We designed two techniques to perform boolean operations between two 3D objects in immersive virtual environments. The first technique is based on gestures humans do to move objects in the real world. To perform boolean operations, users drag two objects together until they are intercepting in the desire position. After that they decide which of the boolean operation they want to do by moving away the parts of the final object that they don't want. The second technique is based on traditional menus with buttons. We represent menus by a three dimensional object separated in sections. Each of those sections represent the selections that users can choose, one for each of boolean operations and has a little preview of the result of that operation. This menu appears when the user intercepts two objects and disappears if user starts any other operation. By touching the sections the intercepted objects change to show a preview of the result of the operation represented by the touched section and grabbing it applies that operation to the objects. We also had a third technique that we used as a baseline. This technique uses handheld controllers with physical buttons as input to grab, opening the creation menu and to choose the boolean operations. Like on the gesture based approach, users have to grab both objects and intercept them to start the operations. After that, by pressing the button corresponding to the desired boolean operation, that is applied to the objects.

We created a prototype where we implemented all three techniques and performed tests with users. We asked them to perform a task of reproducing a 3D model with each of the techniques. This model was based on the Casa da Música Metro Station in Porto, Portugal. We timed tests to compare efficiency of users with each technique. We logged total times as well as individual times for each type of manipulations. In the end we asked users to answer an inquiry to give us subjective feedback about the techniques.

The results have shown us that are no differences in efficiency between the three techniques in boolean operations performance. We also found that users took more time to prepare boolean operations on the gesture based approach because they were afraid of tracking failures. The Menu approach was easier to use, easier to recall and to perform boolean operations and more fun to use than the gesture based technique. With controller based approach it was more difficult for users to remember how to perform boolean operations than with the other two techniques. Most users found that the better technique was the menu based approach.

Our prototype had two modes to represent the body of participants. The first one gave a full body representation with an avatar. The second one only gave the representation of the hand position with two ovaloid shaped objects. We gave each representation to half of the participants and compared the results from each group. We found that self-avatar had a negative impact in the total time users took to perform the task with every approach.

We developed two new techniques to perform boolean operations between two 3D objects in immersive virtual environments. Although no advantages in performance were found, they were as good as the baseline that used traditional buttons to execute the same task even though it had some tracking induced errors. This leads us to believe that our techniques could be superior in terms of efficiency with accurate hand gesture tracking.

6.1 Future Work

In our work there are some aspects that, because they were not the focus of this dissertation or because we found that they didn't work as we desired, can be improved in future or be the focus of future works:

- **Better hand gesture tracking**

  Hand actions were detected using Myo armbands, which tracked the compression of muscles of the forearms and compared that to an initial calibration to decide in which position user's fingers were. This comparison is not perfect and introduces some false positives and also some false negatives, which introduces errors and increases times in task performance. We also tracked rotations using Myo armbands. Although this gave us very good approximations, they were placed in the forearm, near the elbow, while we believe that for object manipulation, tracking the rotation of the hand is more appropriate. This had no effect on our comparison because every technique used the Myo tracking but in future works this should be considered. We believe that a better tracking tool would improve the efficiency of our techniques in performance of our task. To use the gesture based technique with this type of tracking we suggest an alteration to the technique so it doesn't require users to grab the objects for too long.

- **More precise object manipulation**

  By evaluating the quality of the models created by users, it is evident that they are far from perfect. They are not properly aligned and the proportions are not ideal in most cases. This is a consequence of the present object manipulation, not of the boolean operations approaches. To get better results we suggest combining our techniques with object manipulation approaches that enable users to get more precise object placement and rotations.

- **Improve controller based approach**

  Controller based approach had the disadvantage of having no preview of the boolean operations result. This can be changed in future work by implementing previews in this technique. For example, instead of using the any of the controllers to apply the boolean operation, press one controller to preview and both controllers to confirm.

Bibliography

[1] K. Kilteni, I. Bergstrom, and M. Slater. Drumming in immersive virtual reality: the body shapes the way we play. *IEEE transactions on visualization and computer graphics*, 19(4):597–605, 2013.

[2] C. M. Hofmann. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann, 1989.

[3] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. Sketch: an interface for sketching 3d scenes. In *ACM SIGGRAPH 2007 courses*, page 19. ACM, 2007.

[4] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. In *Acm siggraph 2007 courses*, page 21. ACM, 2007.

[5] R. Schmidt, B. Wyvill, M. C. Sousa, and J. A. Jorge. Shapeshop: Sketch-based solid modeling with blobtrees. In *ACM SIGGRAPH 2007 courses*, page 43. ACM, 2007.

[6] P. Lopes, D. Mendes, B. Araújo, and J. A. Jorge. Combining bimanual manipulation and pen-based input for 3d modelling. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pages 15–22. ACM, 2011.

[7] A. S. Forsberg, J. J. LaViola Jr, and R. C. Zeleznik. Ergodesk: a framework for two-and three-dimensional interaction at the activedesk. In *Proceedings of the Second International Immersive Projection Technology Workshop*, pages 11–12. Citeseer, 1998.

[8] T. Ha and W. Woo. Arwand: Phone-based 3d object manipulation in augmented reality environment. In *Ubiquitous Virtual Reality (ISUVR), 2011 International Symposium on*, pages 44–47. IEEE, 2011.

[9] T. Ha and W. Woo. Arwand for an augmuented world builder. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 207–208. IEEE, 2013.

[10] J. Butterworth, A. Davidson, S. Hench, and M. T. Olano. 3dm: A three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 135–138. ACM, 1992.

[11] M. F. Deering. The holosketch vr sketching system. *Communications of the ACM*, 39(5):54–61, 1996.

[12] B. R. De Araùjo, G. Casiez, and J. A. Jorge. Mockup builder: direct 3d modeling on and above the surface in a continuous interaction space. In *Proceedings of Graphics Interface 2012*, pages 173–180. Canadian Information Processing Society, 2012.

[13] J.-Y. Oh, W. Stuerzlinger, and J. Danahy. Sesame: towards better 3d conceptual design systems. In *Proceedings of the 6th conference on Designing Interactive systems*, pages 80–89. ACM, 2006.

[14] J. Wang, O. Leach, and R. W. Lindeman. Diy world builder: An immersive level-editing system. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 195–196. IEEE, 2013.

[15] C. Barot, K. Carpentier, M. Collet, A. Cuella-Martin, V. Lanquepin, M. Muller, E. Pasquier, L. Picavet, A. Van Ceulen, and K. Wagrez. The wonderland builder: Using storytelling to guide dream-like interaction. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 201–202. IEEE, 2013.

[16] J. Jerald, P. Mlyniec, A. Yoganandan, A. Rubin, D. Paullus, and S. Solotko. Makevr: A 3d world-building interface. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 197–198. IEEE, 2013.

[17] U. Schultheis, J. Jerald, F. Toledo, A. Yoganandan, and P. Mlyniec. Comparison of a two-handed interface to a wand interface and a mouse interface for fundamental 3d tasks. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 117–124. IEEE, 2012.

[18] T. M. Takala and M. Matveinen. Full body interaction in virtual reality with affordable hardware. In *2014 IEEE Virtual Reality (VR)*, pages 157–157. IEEE, 2014.

[19] D. Cochard, P. Rahier, S. Saigo, and M. F. Maltouf. Building worlds with strokes. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 203–204. IEEE, 2013.

[20] K. Hald. Low-cost 3dui using hand tracking and controllers. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 205–206. IEEE, 2013.

[21] P. Song, W. B. Goh, W. Hutama, C.-W. Fu, and X. Liu. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1297–1306. ACM, 2012.

[22] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80. ACM, 1996.

[23] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 35–ff. ACM, 1997.

[24] D. Mendes, F. Fonseca, B. Araujo, A. Ferreira, and J. Jorge. Mid-air interactions above stereoscopic interactive tables. In *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, pages 3–10. IEEE, 2014.

[25] V. Kumar and E. Todorov. Mujoco haptix: A virtual reality system for hand manipulation. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 657–663. IEEE, 2015.

[26] M. S. Hancock, F. D. Vernier, D. Wigdor, S. Carpendale, and C. Shen. Rotation and translation mechanisms for tabletop interaction. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'06)*, pages 8–pp. IEEE, 2006.

[27] A. Steed, S. FrIston, M. M. Lopez, J. Drummond, Y. Pan, and D. Swapp. An 'in the wild'experiment on presence and embodiment using consumer virtual reality equipment. *IEEE transactions on visualization and computer graphics*, 22(4):1406–1414, 2016.

[28] K. Kilteni, R. Groten, and M. Slater. The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments*, 21(4):373–387, 2012.

[29] J.-L. Lugrin, J. Latt, and M. E. Latoschik. Avatar anthropomorphism and illusion of body ownership in vr. In *2015 IEEE Virtual Reality (VR)*, pages 229–230. IEEE, 2015.

[30] T. M. Takala, M. Mäkäräinen, and P. Hämäläinen. Immersive 3d modeling with blender and off-the-shelf hardware. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 191–192. IEEE, 2013.

[31] T. Grossman and D. Wigdor. Going deeper: a taxonomy of 3d on the tabletop. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP'07. Second Annual IEEE International Workshop on*, pages 137–144. IEEE, 2007.