UNIVERSITY OF ANTIOQUIA

Remote Touch Interaction with Stereoscopic Visualizations

by

David López Betancur

A work submitted in partial fulfillment for the degree of
Master of Science in Engineering

in the
Faculty of Engineering
Department of Computer Sciences

November 2014

UNIVERSITY OF ANTIOQUIA

Abstract

Faculty of Engineering
Department of Computer Sciences

Master of Science in Engineering

by David López Betancur

Tons of data is gathered in the science fields as the result of tests, simulations or just regular observations. Analyzing this data is a crucial for producing new knowledge, inspect theories or emit a clinical diagnosis. In many cases, the data is multidimensional, being three dimensional representations a better choice when compared with their two dimensional counterpart. It is those three dimensional cases where stereoscopic displays prove their value, providing more perceptual cues that help the viewers making sense of the data. In many cases a static view of the system from one point of view is not enough to scrutinize the data, making the interaction with it an important point to consider. Different approaches use joysticks, gloves, time of flight cameras, among many other different hardware, to enable the interactivity of the visualization system. Among them, the touch-sensitive displays appear as an attractive interaction option providing a simple yet immersive user experience. However, when touch input is directly applied on stereoscopic displays some perception issues appear, diminishing the immersion and effectiveness of the interaction. This work presents a way for interacting with stereoscopic visualizations using remote touch input with tablets, overcoming the perception issues. The analysis and design of a prototype is presented, adapting a metaphor for guiding the interaction. Finally, the prototype is evaluated in a study conducted with experts, looking to gather information about the prototype reception and use.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In their day-to-day, scientists are confronted with data coming from experiments, simulations, observations, and other sources. Understanding this data is crucial for the progress of science. It is used to confirm or reject theories, conclude on experiments, forecast situations (events) and propose new hypotheses. However, the volume of scientific data produced is enormous. At the end of the 1980's, the scientific data sources were considered "fire hoses of information", supplying such amount of numerical data that the human brain could not possible interpret it at the rate it was being produced [1]. This situation promoted the formation of *scientific visualization* (SciViz) as a research field, aiming to aid scientists with the understanding of data, by presenting it in a visual way. Several approaches have appeared since, fusing computer graphics and image processing techniques, UI design, and the use of specialized devices.

1.1 Motivation

The nature of the data handled in sciences is multidimensional in the most general case. In certain scenarios, the use of 3D visualizations is preferred instead of their 2D counterpart, in order to provide more insight on the studied phenomena by presenting more variables in the same view. *Stereoscopic displays* have proven to be of great value for 3D visualizations, mainly because of the additional depth information they can provide. This makes them a good choice for representing complex multidimensional phenomena in a myriad of scientific fields and end user applications.

Moreover, the stereo-imaging is living what appears to be a period of renaissance. After the first attempts to use it in the cinemas during the beginning of the past century, it was ignored by the mainstream film makers, but now it has made a comeback. The entertainment industry is employing stereo displays for films, domestic television, and video-games, impulsing the advance of this technology. This situation entices the search for and development of new applications of these devices.

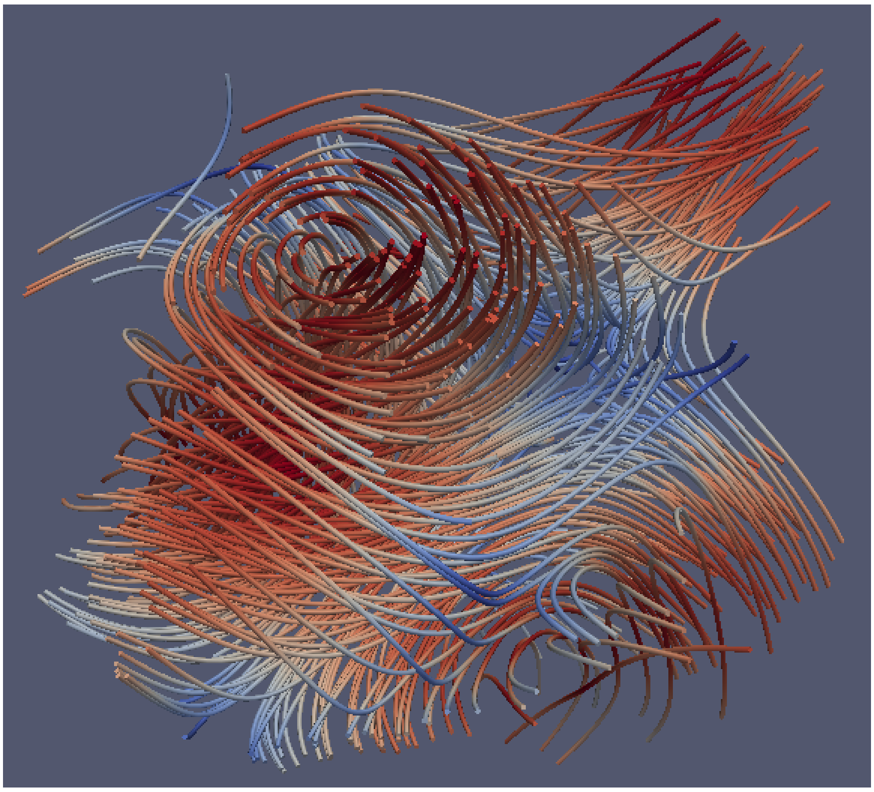


FIGURE 1.1: Stereoscopic displays improve the visualization of complex three dimensional datasets.

Along with the mere representation, SciViz systems should also offer possibilities of interaction. Understanding the data, and not just analyzing it, is easier when viewers are able to manipulate it by changing parameters, point of views, colors, representation modes, etc. In 3D environments, the "universal 3D tasks" categorize all the operations that can be performed in 3D visualizations, used as a base for the construction of *3D User Interfaces* (3DUIs). The 3DUIs allow viewers to invoke 3D tasks by means of physical actions that are captured by input devices connected to the visualization system.

The procedures available to the viewers for executing a task in the visualization are known as *Interaction Techniques*. Making them simple and easy to learn is a challenge for UI designers, who often employ metaphors for this purpose. A *metaphor*, in the UI context, is a comparison made between a real world concept, possibly already familiar to the viewer, and an interaction technique in order to ease its comprehension. Interaction techniques employ mice, pointers, gloves and many other gadgets for receiving input in the form of clicking, pointing, pinching, or more gestures from the viewers. These devices present different degrees of freedom and are specialized for certain types of interactions.

Among these input interfaces, touching input presents interesting advantages for interacting with 3D data. First, it provides a natural interface for executing the universal 3D tasks by means of gestural interaction. The extensive set of gestures that can be detected gives flexibility to the 3D UI designers for choosing mappings with commands in order to execute actions [2]. Second, the ubiquity of tactile devices (such as smartphones and tablets) makes users already familiar with touching interaction, easing the learning curve of the systems based on it. And third, touching devices make feel viewers as if they were in control of the data [3], thanks to the straight conversion of their gestures into actions right on the intended spot.

A good solution for 3D visualization would be a combination of stereoscopic devices with touching interfaces, exploiting the advantages of both. The visual immersion granted by the stereo view plus the sense of control and natural essence of touching interaction make an interesting combination for SciViz applications.

1.2 Problem glimpse

Nonetheless, a series of visual artifacts emerge when touching interaction is applied directly in stereo devices. The properties of a touching screen forbids to touch the projected objects exactly where they are perceived. Likewise, focusing the eyes in a body part and an object at the same time is only possible in a few cases, producing double vision and visual unsettle in the rest. Furthermore, large screens could project a point out of the reach of the viewers, making them translate and stretch in order to touch it. These nuances hinder a fluid interaction and reduce the level of immersion in the visualization.

The potential benefits of combining touch interaction and stereoscopic devices, motivates the search for a solution of its difficulties. The present work aims to investigate how to tackle these problems and propose a solution for them.

1.3 Contributions

The main contribution of this work is the design and implementation of a visualization system combining stereoscopic view and touching interaction while overcoming the problems caused by the simultaneous application of these two, by making use of Tablet-PCs (or tablets) for providing remote multi-touch input to the system. Tablets seem to address all of the detected problems previously considered, introducing a few minor drawbacks.

This work provides a contextualization in stereo visualization and the previous work. Furthermore, it incorporates the "interactive digital photo" metaphor as a guide for the design of the interaction between tablets and stereoscopic screens. Finally, this work presents a user study for the evaluation of the proposed system in a scientific visualization scenario.

In the study it is found that the system was well received by the control group and the drawbacks brought by the use of tablets are not of big concern.

1.4 Goals

The main goal of the project is to propose and implement a system for stereoscopic scientific data visualization manipulated via touching interaction.

In order to accomplish this, it is required

- To survey the already proposed approaches for interaction with stereoscopic displays, analyzing the advantages and weakness that could improve the proposed approach.
- To select an interaction metaphor, suitable for the proposed system.
- To design and implement a prototypical instance of the system.
- And finally to validate the system, evaluating it with the participation of domain experts.

1.5 Scope

The current project focuses on the navigation task (which is arguably the base task for the interaction), leaving for future work and out of the prototype, the tasks related to selection, control, etc.

1.6 Work Structure

The current document is divided as follows. **Chapter 2** introduces the key elements of stereoscopic visualization and SciViz interaction studied as a support for the proposal and identification of the problem. **Chapter 3** describes the identified problems and the proposed solution, revisiting its design and implementation details. **Chapter 4** presents the study design, execution and results. Finally, **Chapter 5** contains the final remarks, conclusions and future work of the project.

Chapter 2

Previous Work and Background

2.1 Scientific Visualization

Scientific visualization is a field of computer graphics dealing with the visual representation of multidimensional data used or produced in scientific research, such as experiment results, simulations, observations, etc with an inherent mapping to 3D spatial space. As with regular visualization, the main goal is to give a better understanding of the data, which in most cases are in numerical form.

The use of scientific visualization extends across several areas of the natural sciences: in Structural Biology to visualize molecules and proteins [4]; in astronomy, it is used to visualize simulations of galaxy formations [5] and 3D representations of astronomical objects such as planetary nebulae [6]; in geology it is used for the visualization of terrains, volcanic eruptions, sea currents, among others; in meteorology it is used to analyze the atmosphere and different climate observations; in physics it is used to visualize several phenomena involving vector quantities, such as velocities and forces; in medicine it is used for the evaluation of different parts of the human body, obtaining the data from different sources such as MRIs, ultrasounds and X-ray tests.

The Scientific visualization uses different techniques from the computer graphics world in order to obtain a good render of the data. For instance, the visualization of medical datasets often employs volume rendering and raytracing, or for improving the view of molecules, engineers use impostors and ambient occlusion. Rasterization and shading are employed in almost everything else, most of the time obtaining interactive speeds.

2.2 Stereoscopic Visualization

Inside of the scientific visualization setups, it is not strange to find the use of stereoscopic screens. The extra information of the depth offered by those devices facilitates the interpretation of multidimensional data, improving the overall immersion in the visualization, the spatial location [7] and giving a better understanding of highly complex 3D phenomena [8].

Stereo Vision

The stereographic screens implement the principle known as *stereoscopy*. That term comes from the greek *stereos* which means "solid", and *skopeo* which means "to see". It is the name given to the phenomenon of perceiving solids in 3 dimensions. In that process, the brain analyzes the image presented by the eyes and determine the 3D position and size of the object. In particular, the determination of the depth, understood as the perpendicular distance of the object to the viewer's view plane, involves the use of several perceptual hints known as *visual cues*, coming from the observed scene. These cues can be *monocular* if are received from each eye independently, or *binocular* if the two eyes are involved.

Among the monocular cues, the most important for judging 3D position in real scenes are *perspective* and *occlusion* [9]. Perspective comprehends the size transformations suffered because of the depth. According to this cue, the same horizontal or vertical distances situated at two different depths will be perceived with different sizes. This causes that the same object is perceived as bigger when it is closer, and smaller when it is farther, giving the brain a clue of the possible depth of the object when the real size is known. The occlusion cue helps to perceive relative depths between objects. When an object A is partially occluded by an object B, B is be perceived as closer than A. If the farther object is fully occluded, then there would be no clue to judge the relative depth between them.

The most important binocular effect is the *stereopsis*, which is required for noticing the actual depth of an object. The stereopsis is caused by the difference between the two images reaching each eye, also known as *binocular disparity*, which the brain merges into a model in three dimensions.

An important detail about the stereopsis effect is that it can be also created artificially. There are two tasks involved in producing an artificial stereopsis effect: **stereo image generation** and **stereo displaying**.

2.2.1 Stereo Image generation

Generating a stereo image involves rendering two images, one for each eye, that can be merged later in the brain by stereopsis giving the illusion of seeing a 3D object. Each image is the result of rendering the visualized scene from a *virtual camera* that emulates a viewer's eye.

Virtual Camera

A virtual camera is a mathematical model used in the rendering of 3D scenes for emulating a real camera located within the scene and pointing in some direction. The virtual camera also contains a perspective model which is used to determine how perspective is applied for rendering 3D objects into a 2D image.

The standard 3D virtual camera is represented by two matrices, commonly known as the *view matrix* and the *perspective matrix*. The view matrix is an orthonormal basis using as its axes three vectors determined by the position of the camera, where the camera is located in the scene; the focal point, where the camera is looking at; and the up vector, indicating where is up.

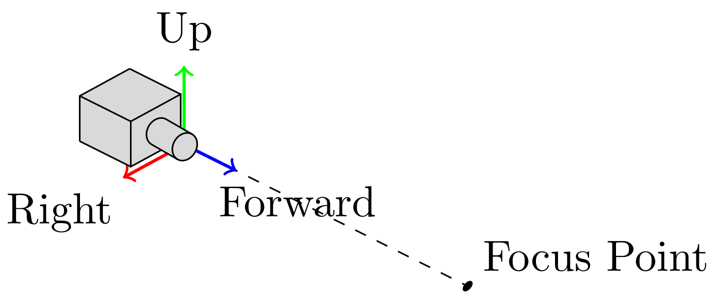


FIGURE 2.1: Classical abstraction of a virtual camera.

The traditional perspective matrix is defined by the aspect ratio, the field of view angle, and the values for the near and far planes, representing the limits of the closest and farthest depth that can be sought. The perspective matrix can be visually represented by a frustum, which is a squared pyramid cut by a plane after its apex. This can be seen in fig. 2.2.

For stereoscopic rendering, there are two common approaches for modeling the virtual cameras: The crossed optical axes model and the off-axis projection model.

Crossed optical axes model (Toed-in Camera)

The crossed optical axes model, also known as the toe-in model, replicates the vergence/convergence eye motion. The *vergence* is the independent rotation of the eyes that allows them to focus a point. Focusing a closer object requires to rotate the eyes

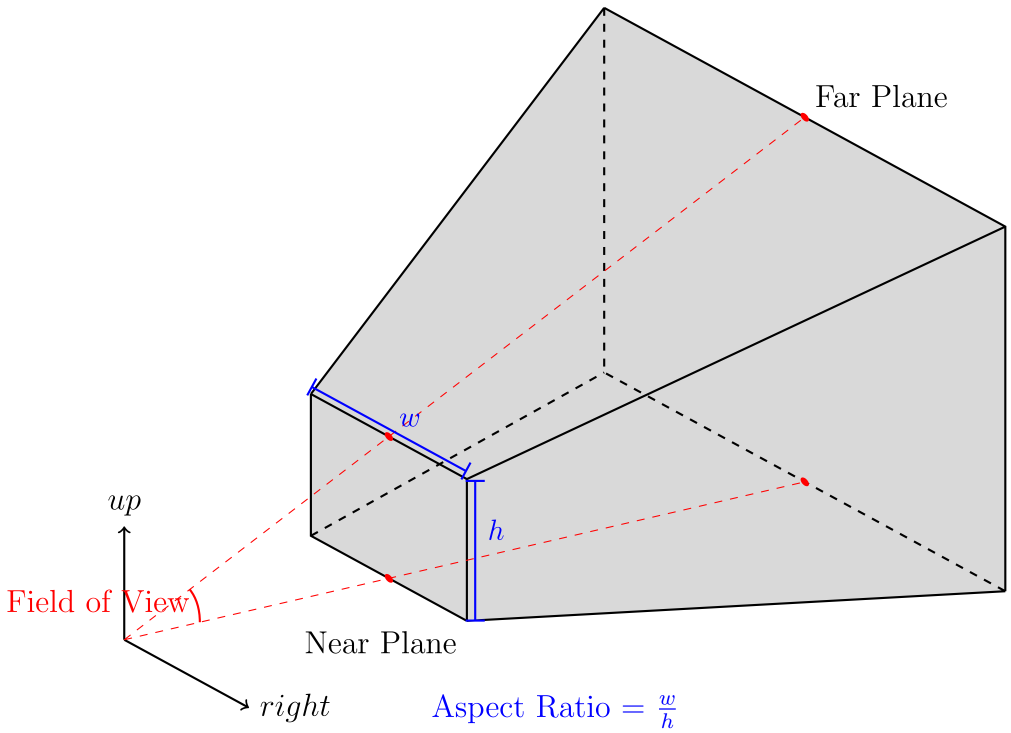
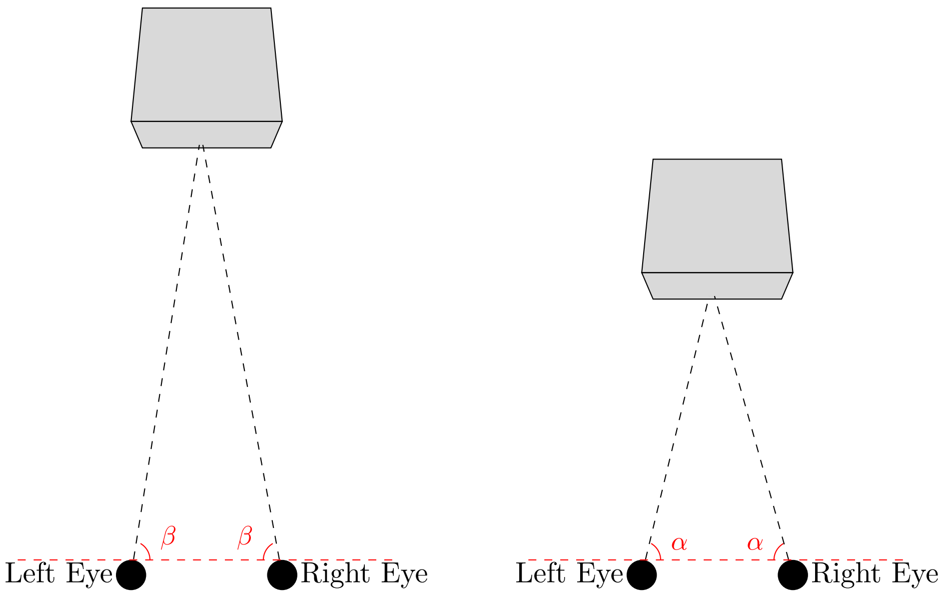


FIGURE 2.2: Frustum representing the perspective of a virtual camera. The visible area is painted in gray.

towards each other (convergence); a farther object is focused by rotating the eyes apart (vergence), going until an almost parallel line of sight. Fig. 2.3 represents this concept.



(a) Object Far.

(b) Object Close.

FIGURE 2.3: Convergence. $\beta > \alpha$.

For implementing this model, the left and right virtual cameras are rotated with respect to the vertical axis making their line of sight to coincide on a point of the focused plane as depicted in fig. 2.4. With this model, the amount of "stereoscopic effect", understood as the perception of an object coming out from the screen or going in, is easily controlled by the rotation of the cameras, making the toe-in model a common choice for television production [10].

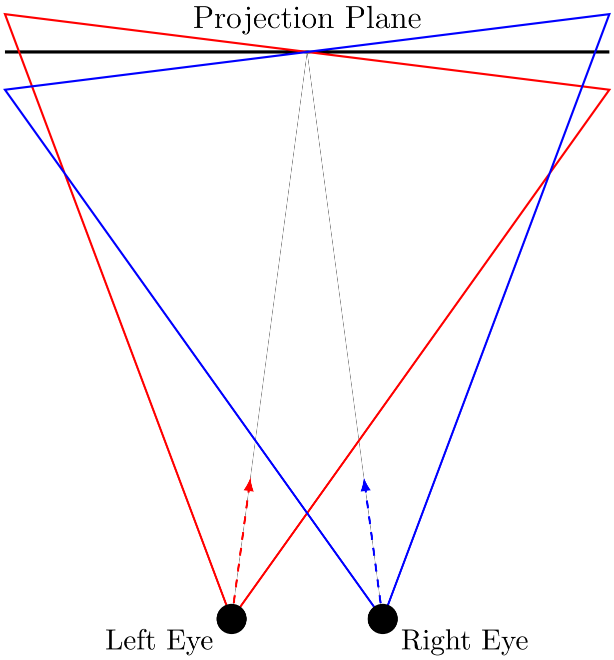
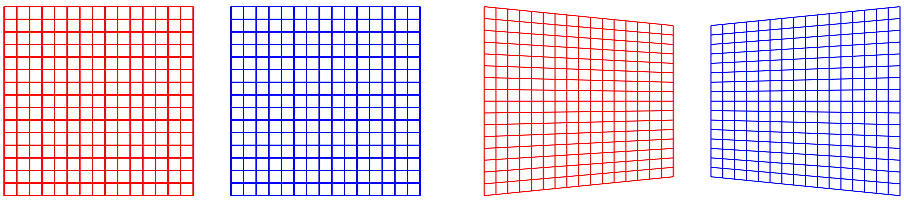


FIGURE 2.4: The cameras for each eye are rotated towards a focused point, simulating the eye convergence.

Nonetheless, this model favors the appearing of visual distortions as a consequence of the *Keystone effect*. This effect is caused by the difference in the planes where reside the images of each eye, curving the visual space depending on the depth as shown in fig. 2.5. The distortions introduced by the Keystone effect impede the estimation of sizes and distances in the image, reducing the effectiveness of the visualization.



(a) Image for the left and right eye. (b) Toe-In projection.

FIGURE 2.5: KeyStone effect caused by the Toe-In method. It distorts the sizes of the real data (a) by presenting a different rotation for each eye (b).

Off-Axis projection model

Inversely to the Toed-in camera, this model assumes that both sight lines are parallel. The projection of each eye is changed in order to make the screen parallel to the projection plane of the images. This in turn changes the shape of the projection frustum to a skewed pyramid as seen in fig. 2.6. Different to the standard model, the off-axis projection is based on the geometry of the room containing the screen and the eye position. The virtual camera frustum is formed by putting the eye at the apex and the screen as the base. Deering presents the implementation details of this model [11].

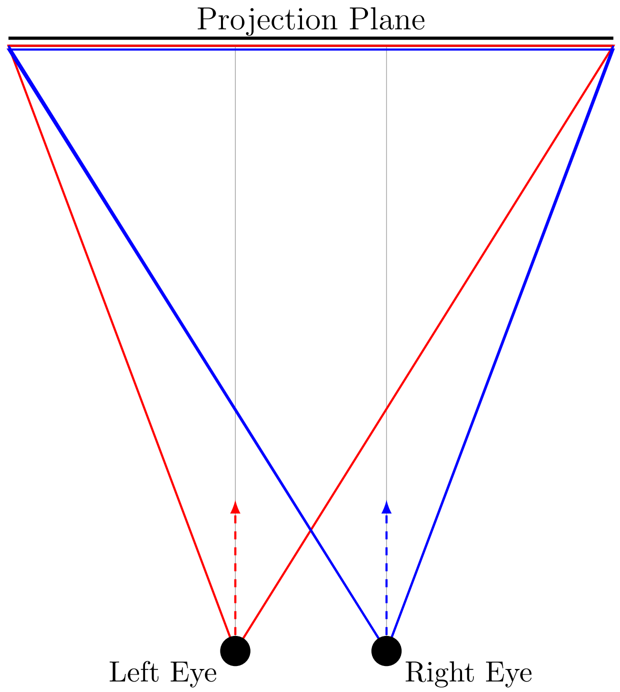


FIGURE 2.6: Both lines of sight are parallel, creating skewed projections on each side.

The main advantage of this projection model is that it neglects the appearing of the Keystone effect, preserving the sizes and lengths of the objects. Also, it is possible to select a new coordinate system based in real world coordinates for specifying the projection, simplifying the setup of the visualization system.

2.2.2 Stereo displaying

After generating the images, it is necessary to deliver them to the respective eye. For doing this, people have employed systems characterized by the use of glasses and special screens or even mounted devices worn by the viewers. Arguably, the most common method for presenting a stereo image has been the use of glasses in combination with a special type of rendering or specialized hardware. According to the level of interaction of the glasses with the screen, the stereo systems are classified as passive or active.

Passive systems

The passive systems rely on physical properties of the light, such as the color and polarization, for blocking the images arriving at the eyes. The most common passive systems employ anaglyph or polarized glasses.

Anaglyph Glasses

Anaglyph glasses are the most basic and inexpensive way of generating stereopsis. The two images are rendered in the same view by using complementary colors. Then, the glasses have lenses with color filters, different for each eye, which separates the encoded images by color. This results in only one image reaching one eye.

Most of the anaglyph glasses employ the colors cyan and red. The red filter allows the red light to go through, and at the same time, blocking the cyan light. Similarly, the cyan filter lets the blue and green light to pass and blocks the red light. In this way, the image for one eye is rendered in red and the other in blue.

The advantages of this technology are the low costs in hardware and glasses, and the broad compatibility with other current existing technologies. Basically all polychromatic screens can display stereoscopic content for viewers with anaglyph glasses. On the other side, the anaglyph 3D imposes a big constraint on the colors, restricting its use to only a few with the respective shading. An explanation of this system can be seen in fig. 2.7

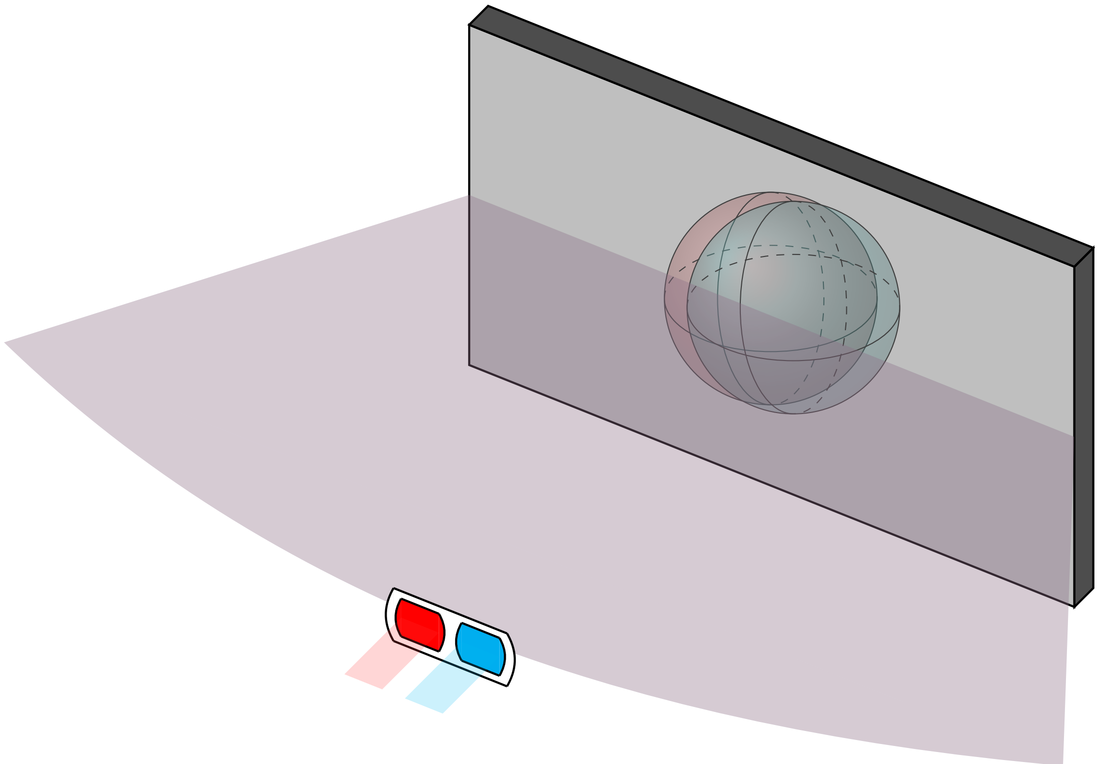


FIGURE 2.7: The lenses filter the light by color for each eye.

Polarized Glasses

Light, as other electromagnetic waves, can oscillate in several directions. The direction can be manipulated with the use of polarization filters which only allow the pass of light going in one predetermined direction. Hence, it is possible to construct stereographic systems by emitting polarized light and filtering it with polarized glasses.

The polarizers are necessary to emit polarized light. An example of a polarizer are the silver screens, which are commonly used in 3D cinemas. Nonetheless, OCR and even LCD monitors in combination with an external polarizer, such as cellophane sheets, can also be used as 3D screens as described by Iikuza et al. [12].

Besides polarizing the light in the screen, the viewers need to wear filter glasses with opposite polarization orientations, for blocking the light in the respective eye and pass the light for the other. Figure 2.8 presents the basics of this system. A drawback of this system is that the polarization makes the image look darker than it really is.

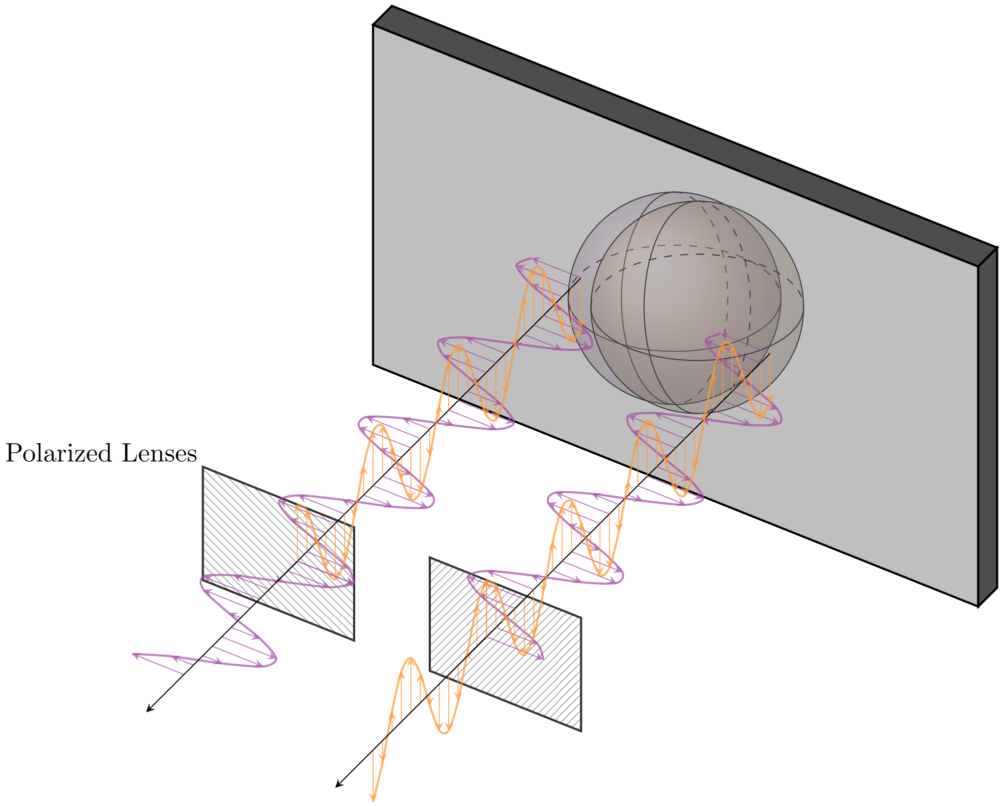


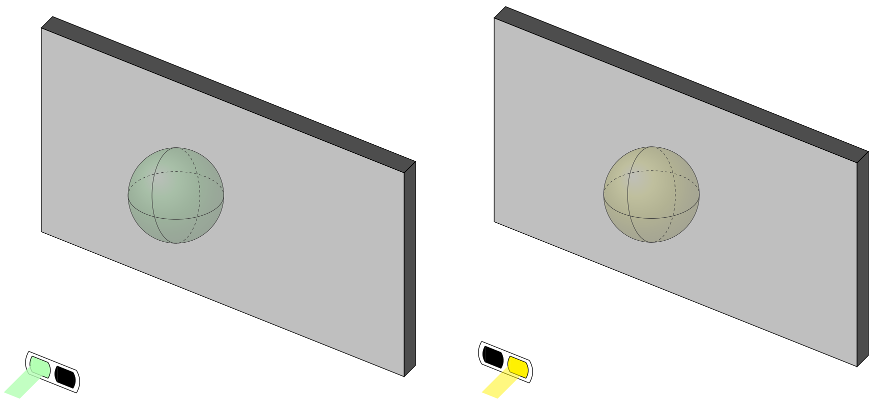
FIGURE 2.8: The lenses block the polarized light according to its phase.

Active systems

Another mechanisms for generating stereopsis are the active systems. They are called active because they require the coordination of screen and glasses for dynamically block the light reaching each eye in every frame. The coordinated blocking can be done thanks to the use of shutter glasses, which can be turned on and off remotely and on demand. Typically, the blocking does not require the alteration of any physical property of the light, conserving the colors and tonalities of the images. Because of this, the active systems are a good alternative for generating high quality stereographic images.

The active systems work in a cycle of two steps. At each frame, the screen displays an image for an eye, and sends an RF signal to the glasses ordering it to block the lens for the other. Next the screen refreshes, shows the image for the other eye, and sends a signal to block the other, ending the cycle. This process is depicted in Figure 2.9

It can be noticed how having a high frequency screen is important for an optimal view in an active system, given that each eye is effectively receiving only half of the images displayed. In the cases where the refresh rate is not high enough, the viewers could perceive some flickering. This fact, in combination with the actual cost of the specialized glasses, make this method more expensive than the passive systems.



(a) Left Eye. (b) Right Eye.

FIGURE 2.9: Basic cycle of an active system. Half of the time, the screen presents an image for the left eye (a), and the other half for the right (b). At the same time the lenses block the light for the other eye.

Head-mounted devices

Another way of getting different images on each eye is by bringing two screens close enough to the viewers' eyes. This is the idea behind the Head-mounted devices (HMD). These devices carry two small screens, one for each of the viewers' eyes, making it possible to present two independent images at the same. Generally, they could provide a wider range of view when compared with traditional screens because of its near distance to the eyes. HMDs have not gained great notoriety when compared with the previous methods based on glasses, but new approaches are appearing, mainly aimed at the videogame industry. An example of this is the OculusRift. [1]

[1]http://www.oculusvr.com/

2.3 Interaction

An intrinsic goal of a visualization system is to enable the viewers to comprehend the multidimensional data they are visualizing in the best way. This often implies changing variables controlling the view, such as the location in the scene, the view direction, colors, positions, and other characteristics of the visualized dataset. For this reason, it is of primary importance for an effective visualization to allow some form of interaction between the viewers and the scene.

Bowman et al. [13] define a set of basic interaction tasks for 3D environments and call them the *3D universal tasks*. These tasks constitute the building blocks of the user interface for an application dealing with 3D data. They are classified as navigation, selection, manipulation, system control and symbolic input.

From these categories, navigation is arguably the most basic and needed task, because it is required when the viewers want to change their point of view or orientation inside the virtual scene. The navigation can be divided in two sub-tasks: travel and way-finding. Travel takes care of the actual displacement of the viewers in the scene, effectively changing their points of view. Way-finding is the cognitive side of the navigation task, which involves asking the questions about the current location, the desired location and how to achieve it. Selection tasks let the viewers specify one or more parts of the data presented in the visualization in order to execute other commands on them. Manipulation tasks enable the viewers to change the properties of the visualized data. It is very likely that these two tasks are used together, because generally the intention after a selection is to perform any kind of manipulation on the properties of the specified data. Nonetheless, there are other commands such as *delete* that instead of modifying properties of the data, removes parts of it. System control tasks are commands that have a repercussion on the visualization system in general, for example, changes in the visualization mode. These type of tasks are commonly started from 2D UI elements, such as buttons in menus, or from the command line. The symbolic input tasks deal with the way of introducing alphanumeric information that is sometimes needed as parameters for other operations.

3D User Interface

The execution of the 3D tasks makes noticeable the need of a mechanism that the viewers can use to specify their intentions. All the elements required to translate a viewer's intention into a 3D task conform what is known as a 3D User Interface (3DUI) [14]. It involves the use of hardware elements, used for input and output, and software elements that are in charge of delivering the action signals and triggering the corresponding task.

Interaction Technique

The actual process of translating a viewer's intention into a 3D task is known as *interaction technique*. This process ties together a subset of the 3DUI elements in a course of one or more steps. The course is regularly initiated by the use of hardware devices, and gives the viewer some form of feedback at the end of each step. For example, an interaction technique for a desktop application could be "Press and hold the left button of the mouse on top of a file to select it, then drag it to the trash bin icon in order to delete it from the desktop". An example in 3D could be "Press and hold the left button of the mouse on the horizontal arrow, then move the mouse to pan horizontally the camera".

Interaction Metaphor

Making the users remember the interaction techniques becomes a key factor for the proper use of a 3DUI. This involves the assimilation of the steps required to execute a task and the feedback obtained from the system. As any other cognitive process, interiorizing the interaction techniques imposes a learning curve on the viewers. Such learning curve can be alleviated if the interface is designed with elements and actions that are already familiar for the viewers. An *interaction metaphor* is an interaction technique that shares similarities with other processes, possibly coming from unrelated fields, that are familiar (or even could be already known) to the viewers. The interaction metaphors constitute the base for the design of natural user interfaces.

An example of a common interaction metaphor is the "desktop metaphor". In this metaphor, the file system in an operative system can be managed as a real world desktop having documents and folders, windows representing pieces of paper in top of the desktop, etc.

2.3.1 Input devices for 3D Interaction

In the context of user interaction, *input* refers to any detected user physical action that provides information for performing a task in the system. Such information can be of continuous or discrete nature. Continuous input provides the values captured at interactive rates of continuous variables, such as positions, rotations, pressure levels, etc. This type of input is generally used as parameters for the different tasks. On the other side, discrete input has a binary connotation, informing the system if a determined event has happened. Examples of discrete input are button presses, touches, hand gestures, etc. They are commonly used as control for launching tasks.

Several devices have been used in 3DUIs for sensing the user input. They differ in the *degrees of freedom* (DOF)— referring to the different independent actions that can be performed by a user — they support, the information nature they provide and the tasks they are specialized for. Nonetheless, "generic" input devices such as Mouse and Keyboard have been proven capable of providing 3D input. A set of this devices for interaction with stereoscopic displays is analyzed next.

General use Devices

Although the *mouse* counts only with 2 DOF, has been used extensively for 3D applications because of its ubiquity. Using it simultaneously with a *keyboard* is also common. The combinations of key-presses with mouse motion and clicks allow the invocation of 3D actions. An advantage of the mouse use is the familiarity that users already have with this device because of its extensive presence in desktop computers. An obvious disadvantage is the reduced degrees of freedom, making the 3DUI designers to look for different ways of mapping the same mouse actions to several commands, increasing the difficulty and steeping the learning curve of the interface. For example, the same mouse motion could mean translation or rotation depending if a key or a mouse button is pressed.

An early system introduced in Yamashita et al. [8] presents an stereoscopic visualization system of magnetic flux lines, Foucault current stream lines, and magnetic flux density. A screenshot of the system running can be seen in fig. 2.10. The system allows different presentation modes by changing the mapping of these variables to visual properties such as color and line shapes. The interaction uses the mouse for navigation by presenting menus for moving to a viewpoint and changing the reference point. Other actions such as configuring variables (cross-sections, streamlines, and colors) can be triggered by clicking buttons with the mouse.

This approach is somehow limited due to the execution of 3D continuous tasks with discrete input interaction, in this case clicking menu items, hindering a fluid exploration. Also, it can be disruptive having the mouse pointer and the 3D data in the same field of view, because the pointer stays in the same parallax (the screen plane) while the data can appear closer or farther. This scenario favors the appearance of some perception problems that disrupt the stereoscopic effect, such as the physiological diplopia, that appears when the viewer tries to focus on two objects with different parallaxes. Moreover, when the eyes focus on the 3D data and the mouse pointer goes over it, a contradiction of visual cues (occlusion and stereopsis) occurs and the stereoscopic effect is disrupted. These problems are explained further in section 3.1.

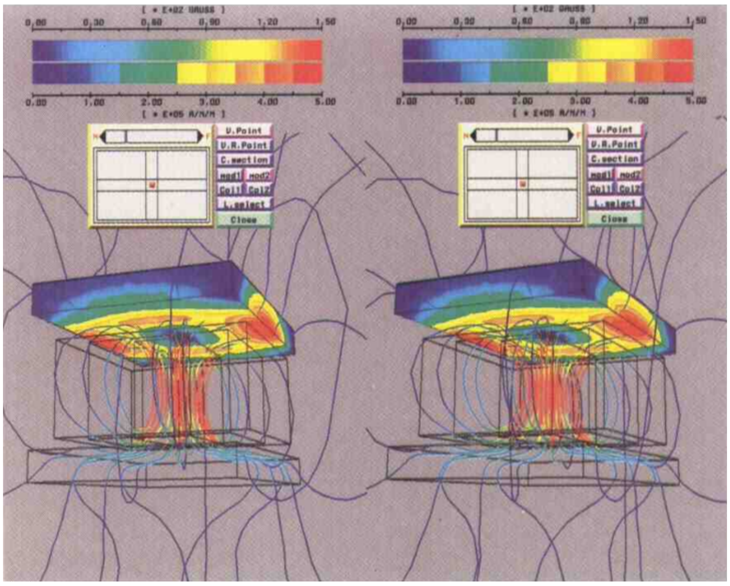


FIGURE 2.10: Snapshot of the user interface used for magnetic fields visualization. Image taken from [8].

An attempt to address those problems was found by Azari et al. [15] who described the implementation of a stereoscopic 3D mouse cursor. Another solution was presented by Stenicke et al. [16] who implemented a 3D depth cursor for interacting with a stereoscopic desktop application that mixes monoscopic (Windows and GUI) and stereoscopic 3D elements. Nonetheless, Bryden et al. [17] analyze a similar setup but find that the participants of a stereographic visualization session spent considerable time of the interaction merely setting a viewpoint with the mouse.

Time-of-Flight Cameras

Another input devices used in interaction are the Time-of-Flight (ToF) cameras or depth cameras. These cameras produce depth images in which each pixel represents the distance of the camera to a respective point in the scene that is being captured. The camera gets a depth image by emitting a phase-modulated ray of light (near infrared) from a light source illuminating the scene. Then, an array of sensors detect the bounce of the initial light, and the depth is computed by comparing the current phase modulation (at the time of reception) with the phase of the received signal. A basic overview of the operation of a ToF is explained in fig. 2.11.

Time-of-Flight cameras are used in interaction as “free hand devices”. Combining them with computer vision techniques it is possible to read input from the viewers in the form of gestures, generally performed with their hands or arms. The Microsoft Kinect [2] is an example of a commonly known ToF.

[2] http://www.xbox.com/en-US/kinect

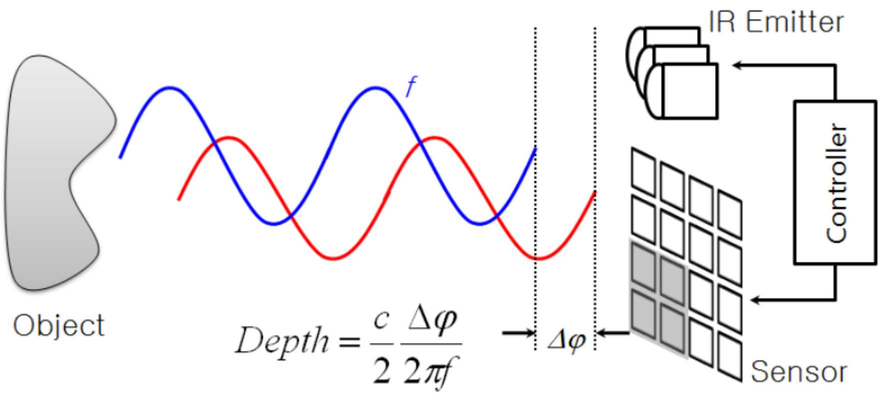


FIGURE 2.11: The difference of the emitted light and the current system phases are used to calculate the depth of an object. Image taken from [18].

Benko et al. [19] present one application of the ToF cameras for the interaction in a stereoscopic augmented reality environment. Viewers wear active shutter glasses to visualize an augmented projection in a curved tabletop that can reproduce real physical objects. The ToF camera captures the geometry of the real object and a regular camera its colors. A geometry mesh is generated for the the viewers' hands each frame. Finally, in this approach the interaction is done through a physical engine that simulates the interactions between the objects and the hand meshes.

Although is not with an stereoscopic visualization, a different interaction approach is followed by Soutschek et al. [20] who use a ToF camera to support free-hand interaction for the navigation in medical imaging applications. A classifier is trained with 3D depth images to recognize 6 different gestures, allowing the viewers to move a cursor across the 3D data, click and select a section of the data, rotate, translate, and resetting the view to a default position.

However, ToF cameras also have some weaknesses. Even though ToF cameras enable the user to interact without any extra equipment, pointing to a particular part of the 3D data on a stereoscopic display using the hands can also cause physical diplopia when the hand or finger interposes between the eye and the stereoscopic image, subsequently losing the visual immersion [21, 22]. Other setups (e.g, [23, 24]) solve these issue by projecting the stereoscopic images into see-through displays and capturing the hand or fingers positions inside constrained areas behind the projections. Nonetheless, in every case, the viewers need to stay in the viewport of the ToF, limiting the range of application to a stationary context [25]. In addition, remembering the gestures needed for the interaction represents a learning curve that might demand a lot of memorization [26]. Similarly, these systems often lack of explicit discrete inputs such as buttons, making selection techniques and clicks non-trivial [27].

Input Gloves

The term *gloves* refers to different gadgets worn by the viewers in their hands or arms that can capture input from the users. These devices can read gestures from the viewers' hands usually with greater accuracy than time-of-flight cameras. A common way of implementing an input glove uses an accelerometer, gyroscope and compass to read the hands' motion, fingertip grip pressure sensors, and finger flex sensors to determine the bending of each finger. Additionally, this type of devices can have haptic capabilities —they can provide tactile feedback to the user—, giving more building elements to the designer of the interface. A classification of the different varieties of input gloves can be found in Sturman et al. [28] or Argelaguet et al. [29]

Among the systems that use input gloves for interaction, Robinson et al. [30] present a CAD system for the design of harness cables. With the software, users can design cabling schemes for different devices using a head-mounted display for 3D stereoscopic view and gloves to provide input. Three modes of interaction are supported in the system: model, which is the main operation where the design activity occurs; menu, which behaves like a typical UI menu but can be accessed through gestures; and text screens, containing helpful instructions in text format about how to complete the tasks. The gestures for triggering the design functions use real world metaphors, for example to move an object it is necessary to move the hand to the object position, pinch the thumb and index finger together to select it, move the hand to the new position while holding the pinching gesture, and then release them to place the object in the new position. However, for several other tasks a set of finger poses need to be memorized as can be seen in fig. 2.12.

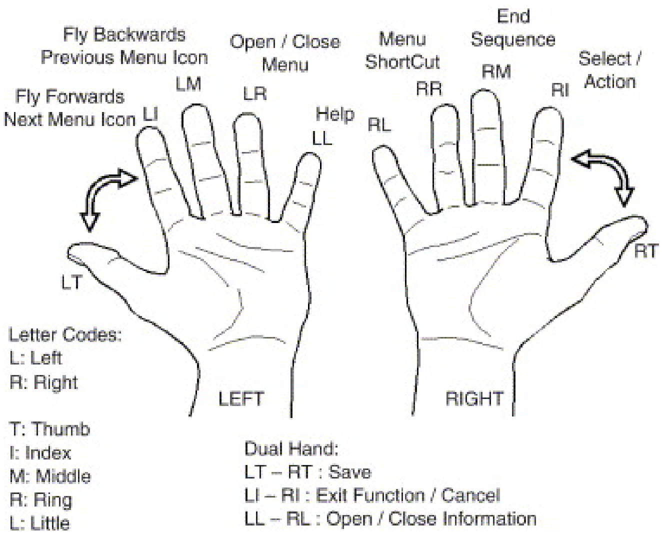


FIGURE 2.12: Description of the gestures implemented in a glove interface. Image taken from [30].

In general, gloves allow *eyes off* or blind interaction because, due to the proprioceptive sense, people can pinch their fingers without any need of looking [31]. Nevertheless, these systems present the same weaknesses of gestural systems, it could be cumbersome to put on the gloves, and using them could restrict the viewer's freedom of movement in some cases [25].

3D Pointing Devices

3D pointing (also known as wsands or 3D mice) devices enable viewers to specify a point in 3D by tracking the device position and orientation, allowing a six degrees of freedom interaction. Most of the time, 3D mice are equipped with buttons for performing discrete input. The tracking modality differs among devices, for example, the use of accelerometers and infrared light for reading the orientation, or the use of tags and external sensors for approximating the position and orientation. An advantage of these devices is their ergonomy and specific design for 3D interaction. Nonetheless, that advantage also plays as a constraint for the type of tasks that can be carried out with them.

Argelaguet et al. [32] present a multi-user stereoscopic system for reviewing the design of a car's engine compartment. Users wear a set of position-tracked shutter glasses and interact with a stereoscopic screen using a wand-like device for pointing individual components on the screen. From a determined point of view, a pointed component could be covered by others, so a special transparency technique is used to remove the occlusions clearing the way for visualizing the intended component.

Dang et al. [33] present a comparison between different input interfaces, including a pen-tablet, voice, and wand. The wand has six degrees of freedom and four buttons. The pen-tablet system provides two modes of interaction: clicking with the pen and sketching different symbols that are translated into actions. The voice interface recognizes a small number of easy to learn commands. In the study is shown how the Wand interface overpasses the others in terms of efficiency and fewer number of errors in a navigation/exploration task of a 3D surface.

An advantage of the wand devices is their number of degrees of freedom, which usually exceed those provided by the typical mouse, and their ergonomic design to cope with some 3D tasks, but at the same time, this makes them context-dependent, hence reducing the benefits when are applied on different visualization scenarios [26].

The WiiMote is a special type of wand extendedly used in 3D interaction almost since its introduction. Lam et al. [34] present the stereoscopic visualization of voxel data coming from a magnetic resonance imaging (MRI) controlled with a WiiMote. The WiiMote allows spatial tracking using a built in camera, making it more suitable for 3D exploration and its buttons can be mapped to control universal tasks. The system allows the viewers to interact with it by using gesture recognition and pointing, enabling them to visualize the rendered MRI from different angles, change the supported rendering modes, control a cursor and zooming. This system is also augmented with sonification, emitting a sound for a selected point in the MRI, giving extra information that can not be seen in the rendering. The approach examines what data variables can be mapped to the sound properties (pitch, volume, timbre, duration ...) with the goal of providing feedback related to the data occluded by other voxels.

Similarly, Bryden et al. [17] evaluate a solution for a collaborative stereographic visualization of molecular shapes using a WiiMote as input, looking to overcome the shortcomings of the mouse. The WiiMote is used as a virtual pointer, trying to get a resemblance to the way PyMOL[3] works. The buttons of the WiiMote are mapped to saving and restoring views and selections, allowing the viewers to quickly going back to previous views or reestablishing an earlier selection of data.

Nonetheless, in the same study, it was found that the WiiMote's camera does not have a sufficient field of view to work with large displays, and the participants showed fatigue due to the need of holding the pointer up to maintain its position. These difficulties ended up precluding the use of the WiiMote for the solution.

Touching

Tactile sensors let viewers to interact with the system using touches. They can register the points pressed by the viewers and some can give an insight on the value for the pressure applied. These devices can process multiple touches at the same time and continuous (or hold) touching, making it possible to interact through *gestures*[4]. Given that people use frequently their hands as a way of interaction with physical objects, the touching metaphor has a low learning curve.

Among the approaches using tablets for the interaction with 3D screens, Bornik et al. [36] present a system for liver surgery planning, using a hybrid system composed by a tablet PC, a large screen projection. In that work, the combination of 2D and 3D interaction proved several advantages, such as relatively high precision in 2D actions and high speed for 3D tasks.

Another system presented by Schönning et al. [37] brings multitouch interaction to an interscopic environment, this is, composed by 2D and stereoscopic 3D elements. The incorporation of 2D elements (such as windows) into a 3D stereoscopic environment is

[3] http://www.pymol.org/ PyMOL is an open-source visualization software specialized in molecules

[4] A gesture is understood as a compounded action that start with an initial touch, and is continued with patterns [35]

done through the introduction of a virtual window analog to a plate. The system is used for urban development and city planning providing polygonal data, and in a medical scenario working with volumetric data. For the city planning activities, it is acclaimed how a multi-touch environment makes possible a seamless collaboration between people of different backgrounds and expertise. For the volumetric medical data, the proposed interaction is inspired on physics, allowing cuts and deformations on the data that can be intuitively triggered by touching and moving points in the screen. The problems of touching directly a stereoscopic screen (See problem description) are taken into account but not addressed in the current solution. The negative parallax of the data is limited to a minimum, with only having a few parts slightly above and below the touch surface.

The positive points make multi-touch an interesting option for interacting with stereoscopic 3D data. The natural essence of its interaction, makes it easily understandable by the users; the interaction through gestures empowers the UI designer with more distinguishable elements mappable to tasks of different scenarios; and the possibilities for multi-user environments promote its adoption in collaborative environments.

A notably disadvantage of the tablets is their lack of fine precision when they are compared with mice, because the fingers occupy more than one pixel at the moment of contact with the screen. It has been determined that interactive elements must be presented in at least 1x1 cm^2 on the touch surface in order to be comfortably picked by an average finger. However, this situation can be circumvented by the application of high-speed selection techniques [2] or the use of pens.

2.4 Summary

Scientific data is produced at high rates by different experiments and simulations, more than the scientists can interpret by just reading the numbers. For this reason the scientific visualization appears with the solely objective of helping with the task of understanding the data. Accomplishing this, requires a good *visual representation* that helps to perceive the data features and a mechanism to *interact* with the data, allowing to visualize different perspectives of it. Among the visual solutions, the stereoscopic displays improve the perception of multidimensional data thanks to their ability to present images in three dimensions. From the interaction point of view, several techniques have emerged to let viewers control the visualization using different methodologies and hardware solutions. Touching appears as a good approach for free-hand interaction with stereoscopic data, which can be done via gestures or physics simulation, allowing the implementation of different metaphors for the interface. Different problems are effectively addressed with the combination of touching interaction and stereoscopic screens.

Chapter 3

Touching remote interaction

The positive points previously found make multi-touch an interesting option for interaction with scientific data [38]. The natural essence of its interaction makes it easily understandable by the users; the interaction through gestures empowers the UI designer with more distinguishable elements for mapping to tasks; and the multi-user possibilities promote its adoption in collaborative environments.

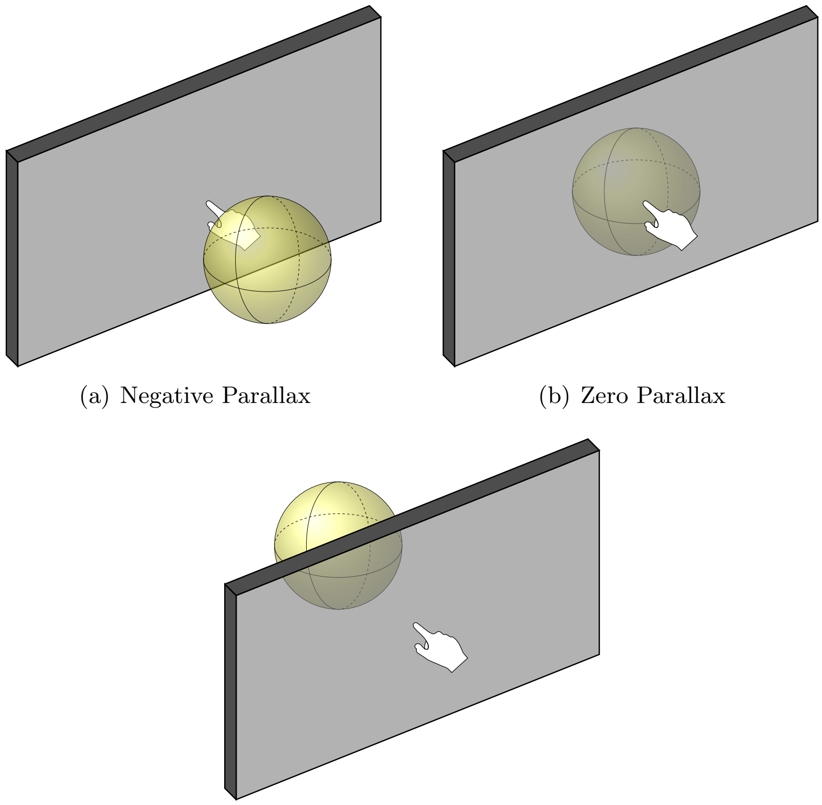
3.1 Problem statement

However, using touch input in stereoscopic environments brings a series of visual artifacts that difficult the interaction and break the immersion in the visualization.

Touching an artificial parallax

The first problem appears when the touching sensors and the perceived object are in different parallaxes. The *parallax*, in the context of stereo-imaging, is the apparent position that a stereo-projected object has with respect to the screen plane. There are three possibilities for a parallax value: positive, when the object is perceived to be behind the screen; negative, when it appears to be in front; and zero, when it seems to lay in the same plane of the screen.

When viewers try to touch an object with negative parallax, their hands will move through the objects. Inversely, if the object has a negative parallax, their hands will meet the screen which will be felt as an invisible barrier. The only possibility where touching and perceived parallaxes coincide is when both are zero [22, 39]. The impossibility of detect touching where the object is perceived disrupts the immersion gained by the sense of control that touching gives. This is depicted in fig. 3.1.



(a) Negative Parallax (b) Zero Parallax

(c) Positive Parallax

FIGURE 3.1: The difference between the screen and the object parallax effects. In (a) the hand goes through the object, in (c) the screen acts as a barrier. Only in (b) is possible to touch the object.

Ambiguity caused by diplopia

Due to the characteristics of the binocular vision, it is only possible to focus in one depth plane at a time, perceiving objects in different planes as double. This phenomenon is called *physiological diplopia* or double vision.

Although physiological diplopia does not present any health threat, it is undesired because the disruption of the stereoscopic effect. But it creates ambiguity when it comes to determine the touched screen coordinate. When the viewers are focusing the object and touching the screen, the body part used to touch (perceived as double) appears to be touching two places. In the other case when the focus is placed on the body part, the object appears double, making it difficult to decide what is the exact point viewer trying to touch. This concept can be seen in fig. 3.2.

Contradiction of visual cues

Another issue found in touching stereoscopic displays is the clashing of visual cues. Putting any body part between the eyes and the screen, occludes parts of the visualization. As mentioned in section 2.2, *occlusion* is another visual cue used by the brain

FIGURE 3.2: Diplopia. When the viewer focuses the finger, they see two objects (left). If they focus the object instead, they see two fingers. Image adapted from [22].

to deduct the depth of an object. When an object is occluding part of another, the former is perceived as being closer. So, having an stereoscopically projected object with negative parallax occluded with a physical object with greater parallax (closer or at the same plane of the screen) causes a contradiction between depth visual cues. This contradiction does not occur with real objects, it only happens with stereoscopic displays disrupting their 3D effect and causing an "unpleasant 3D experience" [40] for the viewers.

Arm reaching

Lastly, an issue presented by large displays with touch screens is the walk-up-and-use [27]. With direct touching the interaction is limited to the objects that are in the viewer's arm reach. For interacting with objects out of it, the viewer is required to move closer, reducing the performance of the basic 3D tasks.

3.2 Solution

The problems exposed in the previous section are the main weaknesses of directly touching a stereoscopic screen. The origin of the visual issues is that the touching surface plane is the same that the projection plane but they are both different to the plane of the perceived object. A common plan for a solution involves the separation of the touching surface and the screen, moving it closer to the plane of the perceived object. This way, the visual problems disappear, provided that now the at the moment of touching, the hand or finger will be located in a parallax that should not affect the visualization perception in any form.

Previous solutions

That idea is implemented in "Toucheo" by Hachet et al. [41]. In this approach, a stereoscopic image is projected through a semi transparent mirror into another screen in top of the touch area. The objects are controlled by interacting with their shadows using widgets. This way, the hands do not occlude the 3D objects because for the interaction they go under the projection area, avoiding the visual issues when the touch is direct.

"TouchMover" is another approach presented in Sinclair et al. [42]. Here, the stereoscopic screen is put in a robotic arm capable of moving in the Z direction (back and forth), allowing to provide haptic feedback to the touch and moving the screen to the corresponding object Z position. Hence, the finger and the object are both in the same plane, avoiding the perception issues caused by positive and negative parallax objects.

Coffey et al. [43] present a touch interactive tabletop used to control a stereoscopic visualization in a bigger screen. The tabletop shows a 2D horizontal slice projection of a World in Miniature 3D map of the data permitting the viewers to interact with it. Therefore, the viewers interacting with what it seems to be the shadow of the data on a horizontal table. Although the projection and touch planes are not the same, the viewers can effectively map their intentions to transform the data in the touch surface.

While those approaches solve the perception and reaching issues presented before, they also introduce in the system a flexibility constraint, making the viewers to remain in a stationary context. Nonetheless, it is possible to remove such constraint by letting the viewers to take with them the touch sensor.

Proposed Solution

The current work proposes the use of tablets as facilitators to the interaction with the system. Tablets have different benefits that make them attractive for being used as an interactive tool. They gather different components for input or feedback that can enrich the interaction, such as tactile sensors, screens, speakers, gyroscopes, and accelerometers.

Tablet screens could display more advanced controls in the form of widgets, giving more space to the data visualization in the main display as shown in previous works [44, 45]. Widgets facilitate higher-level operations such as selection, filtering, seeding, saving view states (bookmarking), taking snapshots of the current view, tagging or adding notes, among others.

Tablets also have the potential to improve collaborative work. They have been used in learning collaborative environments [46] with good results. Similarly, Malcher and Endler [47] concluded that pocket PCs or smart phones are feasible options for classroom usage and Richards and Mantley [48] presented a collaboration system with tablet PCs and a large screen. Moreover, as tablets can be used at distance, in collaboration scenarios the interaction of one user with the display will not cause any occlusion for others.

Finally, tablets can cope with most of the problems presented by other interaction approaches. As the remote communication can be wireless, there are no restrictions related to the location of the viewers, as opposed to ToF cameras based approaches. And because tablets are standard devices with common and known operative systems, they can be adapted to multiple scenarios.

3.3 Solution description

After deciding that a tablet is the main interaction input to the system, it rests to describe the other components that will integrate the system. At the first place lies the use of a stereoscopic screen as the main output visualization display in the system. The use of an eye-tracking method is also required for presenting a correct perspective image for the viewers' position. A computer used as a server, in charge of the visualization rendering, database management and interaction control. And lastly, the employment of a tablet for the interaction with the data. Also as a part of the system, a wireless network acting as a mechanism channel communicating the tablet and the server.

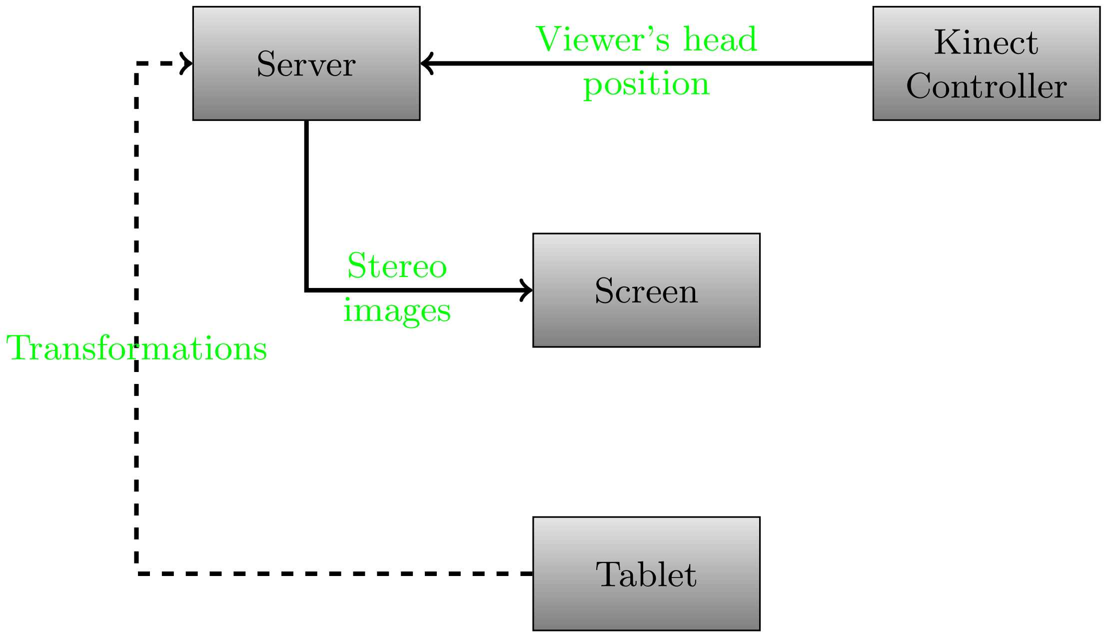


FIGURE 3.3: General overview of the system.

3.4 Visualization System Design

All these components offer several possibilities for the interaction and are oriented to optimize the navigation task, which is the base for other 3D Universal Tasks. This section presents the design of the visualization system from a high level view, indicating the motivations that led the decisions for the design.

Screens distribution

The introduction of the tablet brings a new screen to the visualization environment. This raises the question of what to present on it.

It is clear that the stereo screen is the main output device for the system. The stereo screen can give the best visualization possible to the viewers. The images it generates have the best fidelity and quality as well as reducing all the visual ambiguity as it can. For this reason, this screen should present a view corrected for the viewers position, because when they move around the room, the stereoscopic displays exhibit a perceptual distortion due to the difference of the real viewing position and the one used for rendering the image [49, 50]. This is what motivates the incorporation of an eye-tracking mechanism.

At the same time, the tablet screen is another possibility for presenting data to the viewer. Its smaller size and the fact that it is a monoscopic screen suggest that it should be used for different purposes than data visualization. For instance, it could present an abstracted map of all the data available in the visualization following a world in miniature metaphor [51]. A different option would be to show information augmenting the visualized data [45]. However, presenting the data in the tablet screen improves the interaction, because seeing and touching the data in the same place, gives the users a better sense of control as explained by Hancock et al. [52] with the "Sticky Tools".

After deciding to present the same data in both screens, another question emerges regarding the synchronization between the views. The tablet screen could be synchronized with the main display, showing a monoscopic render of the data from the exact same perspective, or on the contrary, it could be independent and present the data from a different view. In that scenario, the synchronized mode maintains coherence between views, so the viewers are always interacting with what they are seeing in both screens, and more importantly, the gestures they perform are always coherent and expected in both views. Nonetheless, being synced all the time brings some difficulties to the interaction. Continuously updating the virtual camera based on the viewer's eyes position can lead to a "shaky view" in the tablet, decreasing the touch precision specially when interacting with smaller objects [53]. Although filtering can address this problem, it introduces lag which affects the immediateness of the visual feedback for the interaction. Another option would be to utilize more robust tracking hardware, with the negative aspect of increasing the cost of the overall system.

Having the tablet view unsynced by default solves the previously mentioned problems by presenting a frozen/steady view of the data to interact with. This way, the viewers have the possibility of a more relaxed interaction, being able to move freely without having to worry about maintaining a specific posture. However, because both screens might be showing a different view, there is a problem in defining which is the outcome of a gesture on each screen. Whatever be the answer, the tablet view is the one receiving the input directly, so it should react accordingly with the gesture in order to maintain the sense of control. This only leaves to consideration the effect of the gesture in the stereoscopic screen.

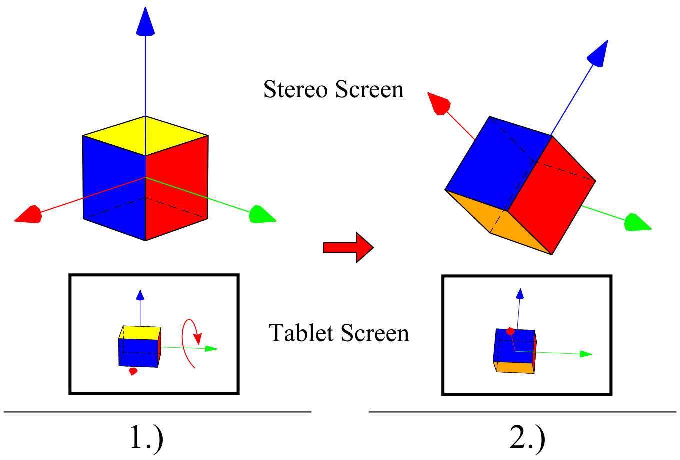
Interacting in unsynchronized scenarios

Two ways of applying the interaction transformation in the data were found for the system configuration: *data-based*, when the intended transformation is applied using the data coordinate system (X, Y and Z axes regardless of the perspective); and *view-based*, when it is performed with respect to the camera axes (Forward, Up and Right).

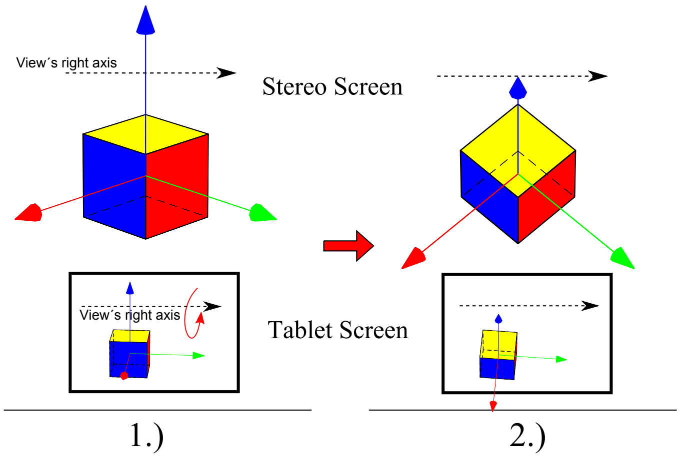
The data-based method leads to unintuitive outcomes of the gesture performed when there is a noticeable transformation between both views, this is, when the angle between the perspectives for the screens is significant. For example, if the viewers intend to rotate the data as seen in the tablet around the x axis, it could result into a rotation around the z axis on the stereoscopic screen depending on their position in the room as seen in fig. 3.4(a).

On the other hand, applying transformations based on the view, conserves the sense of control of the touch interaction in both screens, because the gesture is applied depending on the presented view's perspective. If the viewers intend to rotate the data as seen in the tablet with respect to the view's right, a rotation will occur in the stereoscopic screen using its view's right direction, as it can be seen in fig. 3.4(b). This brings the disadvantage of having data and view desynchronization, creating difficulties for keeping track of what is happening in both interactions.

Taking all of this in consideration, the design choice that presents more benefits is the one having the views unsynced and the transformations applied in a data-based way. In this configuration the gesture outcome is still coherent and its unintuitive nature can be circumvented with a mechanism for re-synchronizing both views. Table 3.1 summarizes the different configurations analyzed.



(a) Data-Based interaction



(b) View-Based interaction

FIGURE 3.4: When the views are out of sync, in the data-based interaction (a) the gesture performed in 1.) (denoted with a red arrow) is assumed with respect to the data axes (in that case the green axis), obtaining the state in 2.). In the view-based interaction (b) the gesture is applied with respect to the view's axes (in that case the camera x-axis).

ViewLink	Transformation	Ixn View	Gesture Outcome	DataLink
Sync	Data/View-based	Shaky	Intuitive	Sync
Unsync	View-based	Steady	Intuitive	Unsync
Unsync	Data-based	Steady	Unintuitive	Sync

TABLE 3.1: This table presents the configuration modes for the view or data synchronization.

Tablet-to-Screen and Screen-to-Tablet mechanisms

The re-synchronization mechanism is then an important part of the system because it allows a reconciliation of the views and a safe reset point for making the interaction outcomes intuitive again. The synchronization can go in two ways, setting the tablet view in the stereo display (Tablet-to-screen mechanism) or inversely the stereo display view in the tablet (Screen-to-tablet mechanism). The system supports both because they can be used in different scenarios. For example, viewers could want to find an interesting view in their tablet and then share it with others, or maybe another person could want to keep looking in their tablet for something more in an already existing view.

Interactive Photograph Metaphor

Putting to work the components of the system in an organized and understandable way requires the introduction of an interaction metaphor. Different works introduce metaphors for interaction of handheld devices and remote screens via touching. Boring et al. [54] show the Touch Projector metaphor, which allows users to manipulate content displayed in remote screens. Users aim a smartphone towards a display in order to "grab" its contents, making possible a touch interaction with them in the phone screen.

From the augmented reality (AR) field, Lee et al. [53] show the "Freeze-Set-Go" method to interact in "shaky" conditions, this is, where it is not simple to hold a mobile device steady while touching its screen for interaction. They propose a similar setup to the one described by Boring et al. but allowing users to freeze the view, interact with it, and then unfreeze again to continue the work in the AR environment.

A combination of the metaphors found by Boring et al. and Lee et al. seems to be an adequate choice for the proposed system. In the tablet/stereo screen environment, the interaction is explained better thinking of the tablet as an *interactive photograph*, adapting the frozen view from Lee et al.'s metaphor. The interaction starts when the users take a photo of the image they see in the stereo screen with the "screen-to-tablet" mechanism, which is basically the same as the "grabbing" action in Boring et al.'s metaphor. At that moment, the tablet screen presents a static render, analogous to a photo, that is used to control the view of the stereoscopic screen through different input actions, such as physically rotating the tablet or touching its screen. While the viewers are modifying such view, they are not constrained to remain in the initial position occupied when taking the photo, and can freely move to a different place in order to get a different perspective of the data in the stereo screen, while still having the photo on the tablet. Then they can put the photo back to the stereo screen, by means of the "tablet-to-screen" mechanism, or keep going with the interaction taking another picture from that new perspective.

The proposed metaphor interactive photograph metaphor sums up from a high level view the main use case of the system. It rests to describe the design of each particular component before presenting its implementation.

Tablet's touch interaction

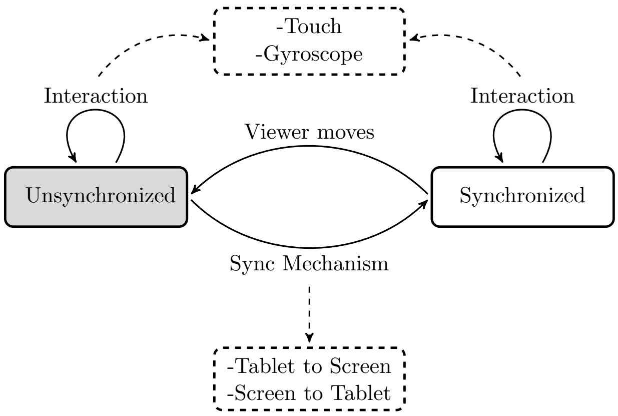


FIGURE 3.5: The metaphor starts at the more general unsynchronized state.

Recognizing the viewers' touch and transforming it into 3D actions is the main functionality of the tablet. Its multi-touch capabilities make possible the interaction through gestures and widgets to effectively capture the viewers' intentions. Also, because the tablet offers a steady view, it is possible to incorporate widgets for the multi-touch interaction. Two already tested techniques are the tBox [55] and the FI3D [56] widgets which have slightly different advantages and could be used for manipulating 3D data.

tBox

The tBox is a 3D widget used for the manipulation of 3D objects in a scene. The original implementation is changed in order to handle transformations to the whole dataset instead of only one object. The users are presented with a 3D cube as soon as they touch the view area in the screen. With the tBox it is possible to perform three types of manipulations: Rotation, translation and zooming, achieving 7 degrees of freedom.

Rotation The rotation starts by touching a face of the tBox with one finger and dragging along an axis forming the face. The touched face is highlighted in order to indicate that it is the active face. The data will rotate around the axis perpendicular to the touched face's normal and the line traced by the finger drag. The rotation axis is selected based on the initial motion of the finger; after the finger moves a few pixels, the axis is calculated and locked for the rest of the dragging gesture.

Translation The translation is invoked when a finger is dragged along an edge of the tBox. When a finger touches an edge, a cylinder is drawn indicating the translation axis. The actual translation is a projection of the dragging on the selected edge.

Zoom The zoom manipulation is a multi-touch gesture that scales the dataset, making it appear larger or smaller. It starts by putting two fingers on any place of the

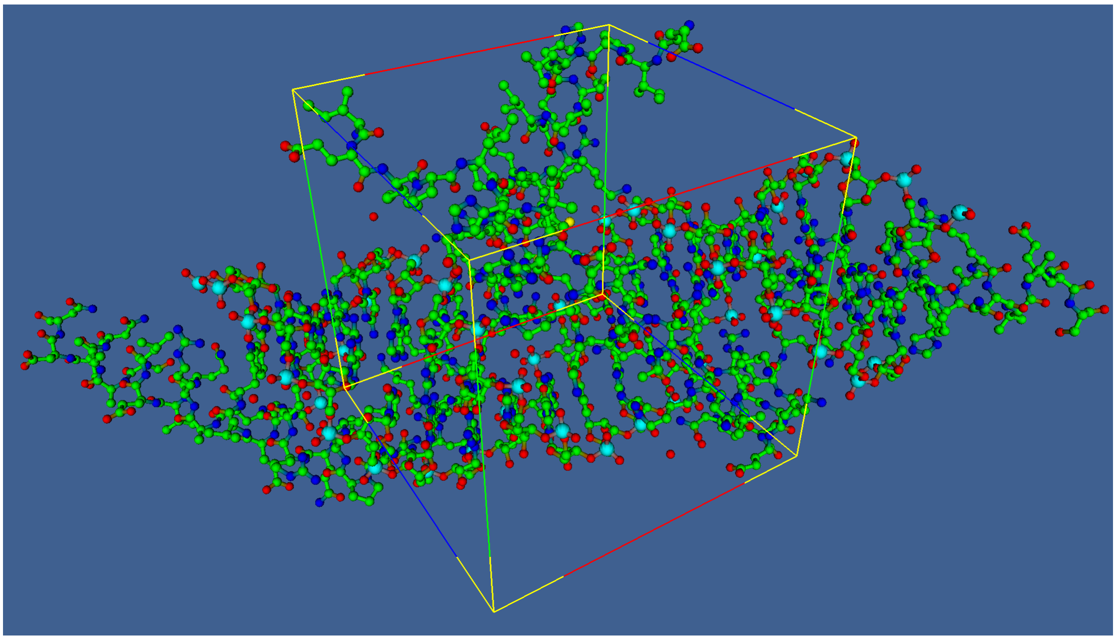


FIGURE 3.6: Tablet interface using the tBox widget.

view, inside or outside of the tBox. When the fingers are pinched away from each other the object will be zoomed in, appearing larger, and pinching the fingers towards will do the opposite.

Because of the constraints imposed on the gestures, the tBox lets viewers a more precise and controlled transformation of the data.

FI3D

The FI3D is a graphical widget designed for explore and navigate 3D data by touching controls in the borders of the screen or by touching directly on the view. The FI3D consists on four bars located in the edges of the view used to start the interaction tasks by dragging and holding them. Having the control widgets at the view borders, gives an unobstructed view of the data in the center. Just as the tBox, the FI3D can rotate, translate and zoom in the data also achieving seven degrees of freedom.

Rotation It is possible to rotate around different axis with the FI3D. Dragging from the bars (located) at the borders towards the center starts the rotation mode, which is maintained while the finger is pressed. While the FI3D is in the rotation mode, dragging horizontally rotates the data around the vertical axis of the view and doing it vertically rotates the data around the horizontal axis. The rotation can also be restricted to only one axis at a time by pressing one of the bars with an extra finger. If the a vertical bar is held, the rotation is restricted to the vertical axis; holding a horizontal bar has the same effect around the horizontal

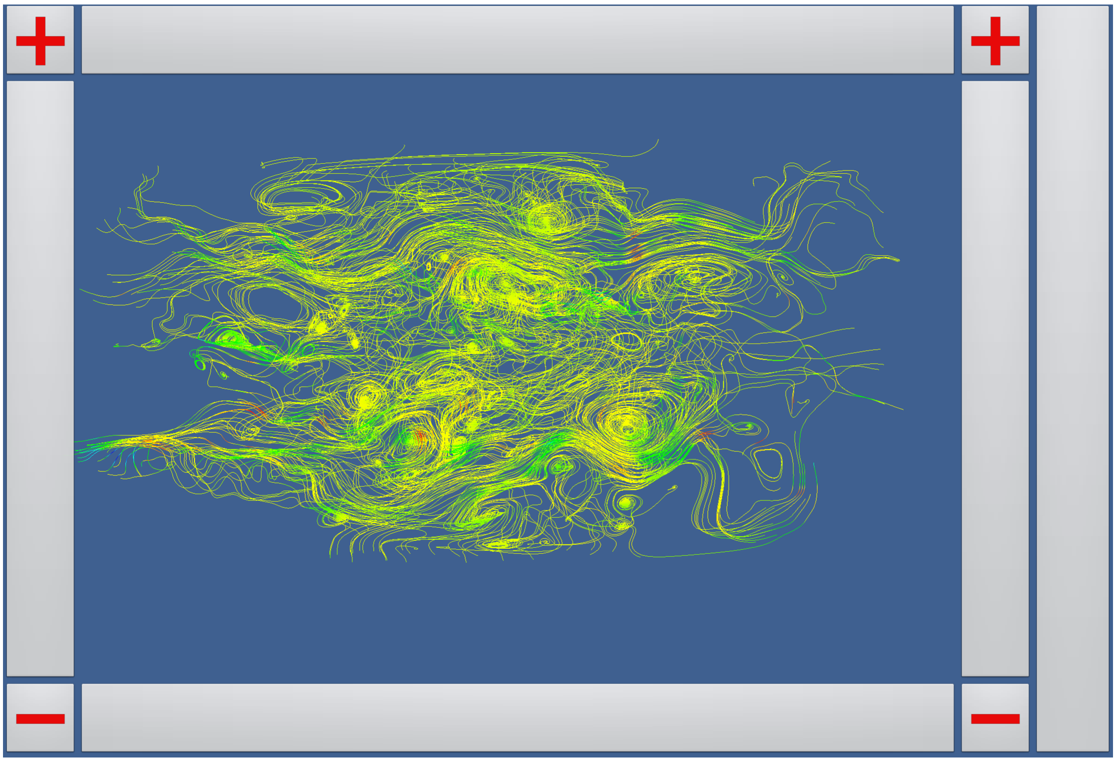


FIGURE 3.7: Tablet interface using the FI3D widget.

axis. Dragging along any bar triggers a rotation around the axis coming out of the screen (rolling) following the rule of the right hand. The same rotation can be performed with a multi-touch gesture by dragging circularly with two fingers in the view area.

Translation The translation in the x- and y-axes (panning) starts by dragging the view with one finger in the direction of the intended axis. It is also possible to translate in the axis coming out of the screen (z-axis) by dragging along the bar located in the right edge. Dragging down pushes the data towards the screen, and dragging up pulls it out.

Zoom Zooming can be invoked in two ways: By dragging away from the buttons located at the corners, or by performing a pinching interaction with two fingers in the view area.

The fast and unrestricted transformations provided by the the FI3D make it a good choice for exploration using touch interaction.

Tactile sensor

Beside the tactile screen, tablets are equipped with more sensors that could be considered for 3D interaction. The accelerometer is a common sensor found in many handhelds. It is used to determine the total acceleration of the tablet, as a result of adding the individual forces applied to it. Another sensor frequently found in the tablets is the gyroscope. It gives the rotational (or angular) velocity of the tablet. Microphones can be used for capturing voice commands, but it has not been an attractive option for 3D interaction, performing worse than other common interaction techniques [57]. Other sensors such as barometers, proximity sensors or GPSs are not suitable for 3D interaction.

Gyroscope

With the values of the accelerometer, magnetometer and gyroscope it is possible to accurately measure the angle of rotation of the tablet in the real world. The tablet's real rotation angle can be used to transform the angular variables of the system, as shown in [58]. Some of them are:

Camera - Point of view The rotation angle of the tablet can be used to manipulate the focal point of the virtual camera. The position of the camera is fixed, while the view direction (the segment going from the camera's position to the focal point) is rotated accordingly with the rotation of the tablet.

Camera - Trackball The position of the virtual camera can be rotated around the its focal point accordingly with the rotation of the table.

Data - Sticky rotation The rotation angle can also be applied to the data. Using a button as the triggering command, gives the idea of physically "grabbing" the visualized object, and applying the transformations to it.

Similar to the touch transformations, it is possible to apply globally or locally the angle read from the tablet depending of which view is the one affected. Because of the same reasons exposed with the touch interaction, it is preferred a global application of the transformation. Additionally, there are two modes of using the input obtained from the physical rotation of the tablet: absolute or spring-loaded. If the rotation is used as an absolute angle, it is directly applied to the data transformation, so any physical rotation immediately affects the data. On the other hand, a spring-loaded mode applies while a triggering input action is performed. This way, the application of the angle is incremental (applied as a delta to the existing angle) and can be controlled by the trigger action. The spring-loaded mode is preferred for giving more control to the viewers, and more importantly, they are not forced to maintain a pose in order to view the data from a certain perspective. They can resume the transformations at any time they require it.

Interaction Summary

In summary, the proposed system presents a stereoscopic visualization of a dataset, using head-tracking for rendering a perspective corrected stereo image. This way, the viewers can freely move around the room while having an undistorted image of the data for their current position. For such goal, the rendering employs an off-axis camera model, which is more flexible than the traditional camera abstraction.

From the interaction point, a tablet handles entirely the interaction with the data, presenting a monoscopic *steady* view at anytime. This implies that the image the tablet display is not updated with the head-tracking information, making both virtual cameras (the ones used for the stereo display and the tablet rendering) to go out of sync. Being out of sync makes interactions with the data unintuitive, because the gesture interaction outcomes are seen differently in both screens because of the difference in perspectives. That problem motivated the introduction of a synchronization mechanism in order to make them sync again.

Two widgets offering seven degrees of freedom take care of the data transformation. The tBox and the FI3D are used for rotating, translating and zooming. Additionally, the accelerometer, magnetometer and gyroscope sensors also integrated in the tablet, provides the system with a lecture of the tablet physical inclination angle. It is used to transform the data in a spring loaded fashion, using a button as a trigger for starting the rotation.

3.5 Prototype Implementation

This section describes the implementation of the proposed visualization system. In contrast with the previous section, this one contains a lower level view of the system with a focus on the techniques and technologies employed. This section completes the description of the solution.

3.5.1 Data and View Transformations

The main use case of the prototype is to provide navigation capabilities to the users, making them able to visualize the data from different points of view. This can be conceptually achieved by transforming the virtual camera of the scene or transforming the data. Transforming the virtual camera in the stereo screen is not a good option because it is already tied to the eye position of the viewers. Hence, the prototype should handle both: data and camera transformations.

Data Transformations

The data transformations required in the system, rotation, scaling, and translation, are of linear nature and can be efficiently represented by matrices. This makes it possible to represent the total transformation on the data as one single matrix, resulting of the application of each single transformation matrix. Now, the interaction metaphor comprises two different views, one for the tablet and one for the stereo display, that gives origin to two matrices in the implementation: a tablet model matrix and a stereo model matrix.

The values of both matrices depend exclusively on the user input, and thanks to the linearity of the transformations, the new values calculated as:

$$\boldsymbol{M'} = \boldsymbol{Tr} * \boldsymbol{M}$$

where $\boldsymbol{Tr}$ corresponds to the transformation read from the input mechanism and $\boldsymbol{M}$ is the data transformation. The transformations $\boldsymbol{Tr}$ are caused by the interaction technique used, converting input gestures into matrices. The implementation of the FI3D can be found in [55]. As indicated previously the tBox is slightly modified from its original implementation shown in [56], adding the pinching to zoom functionality described in for the FI3D. It only rests to present the gyroscope transformations.

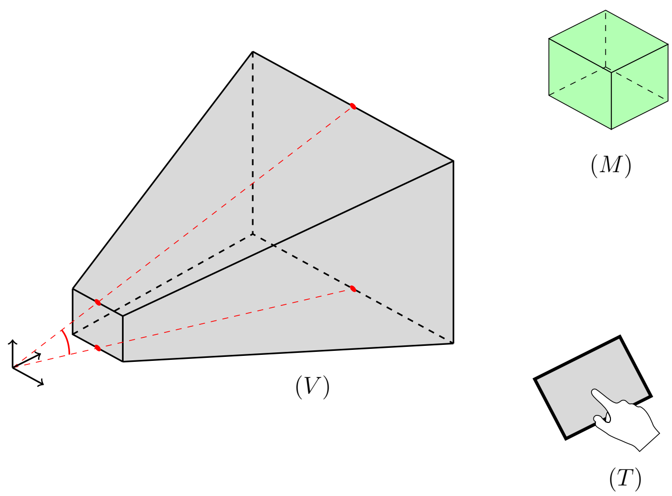


FIGURE 3.8: Matrices used in the implementation. (V) represents the view transformation of the camera, (M) is the model or data transformation and (T) is the transformation read from the tablet.

Gyroscope Transformation

As it was previously stated in the design section, the gyroscope values are used for the rotation of the data in the system. The gyroscope value $\boldsymbol{g}$ is a quaternion indicating the rotation state of the tablet with respect to a "natural base". This system is formed by a vector pointing to the magnetic north of the earth, read with the magnetometer, and the gravity vector pointing down, read with the accelerometer. Taking the cross product of those two vectors gives a third vector pointing west. The three vectors are combined in a "natural base" used as a reference for the readings of the gyroscope.

Getting $\boldsymbol{Tr}$ from $\boldsymbol{g}$, involves the temporal storing of $\boldsymbol{g_o}$ at the moment the viewers press the trigger button for the gyroscope. Then, the quaternion $\boldsymbol{q}$ representing the rotation between $\boldsymbol{g_o}$ and the current read $\boldsymbol{g}$ is computed as

$$\boldsymbol{q} = \boldsymbol{g} * \boldsymbol{g_o^{-1}}$$

with $\boldsymbol{g_o^{-1}}$ being the inverse quaternion of $\boldsymbol{g_o}$. Finally, getting $\boldsymbol{Tr}$ is just a matter of converting the quaternion $\boldsymbol{q}$ to a rotation matrix.

$$\boldsymbol{Tr} = \text{toMatrix}(q)$$

3.6 View Transformations

In the rendering process there is a transformation of 3D points to 2D. In this transformation it is involved the use of the model, view and projection matrices. The model matrix represents the transformations applied directly on the data, which has already been mentioned. The view matrix represents the transformations caused by the camera location in the virtual world. Lastly, the projection matrix contains the information about the projection model used in the scene. Because the view and projection matrices are related with the concept of the virtual camera, in this work they are merged into a single matrix simply called view transformation.

The calculation of the view transformation for the stereo screen $\boldsymbol{Vs}$, uses the viewers' eye position ($\boldsymbol{E}$) as a parameter, provided that it implements an off-axis camera model.

$$\boldsymbol{Vs} = \text{offAxis}(\boldsymbol{E})$$

In contrast, the view transformation of the tablet is not calculated each frame, and it is only modified with the synchronization operations.

View synchronization

Both synchronization methods operate on the matrices $\boldsymbol{Vt}, \boldsymbol{Vs}$ and $\boldsymbol{M}$ with the final goal of making the rendered image the same in both screens depending on $\boldsymbol{E}$.

Screen-to-tablet

This synchronization method changes the view on the screen, so the viewer can see the same image on the screen and the tablet. Given that it is possible to manipulate the view matrix in the tablet, this method is as simple as setting the position component of that matrix equal to $\boldsymbol{E}$.

$$\text{setPosition}(\boldsymbol{Vt}, \boldsymbol{E})$$

Tablet-to-screen

On the other, the implementation for the tablet to screen operation is slightly more complicated given that $\boldsymbol{Vs}$ can not be changed, entailing the modification of $\boldsymbol{M}$ instead. However, changing $\boldsymbol{M}$ brings the undesired effect of also changing the view in the tablet. Therefore, in order to stop the tablet from changing $\boldsymbol{Vt}$ has to be affected too.

The tablet to screen operation is then a multi-step operation. The first step is to calculate the angle α of rotation between the position of $\boldsymbol{Vt}$ and $\boldsymbol{E}$. The next step is to rotate $\boldsymbol{M}$ according to the angle α. And finally, the position of $\boldsymbol{Vt}$ is set equal to $\boldsymbol{E}$.

Tablet-to-screen

```
procedure TabletToScreen(E, Vt, M)
    P ← getPosition(Vt)
    α ← arccos(E · P)                ▷ Get the angle between the two points
    R ← makeRotation(α)
    M ← R · M                        ▷ Rotate the data
    setPosition(Vt, E)               ▷ Update the position
end procedure
```

The screen-to-tablet and tablet-to-screen operations describe how to synchronize the tablet and the stereo screen views modifying $\boldsymbol{Vt}$ and $\boldsymbol{M}$. The description of these two operations completes the view transformation section.

3.6.1 Software implementation

The whole system is implemented in three big software components: server, tablet, and networking.

Server Implementation

The server has as its main purpose handling the datasets, reading $\boldsymbol{E}$ from the eye-tracking mechanism and $\boldsymbol{M}$ from the tablet, and rendering the stereo data with that information. It is programmed as a stand alone application in C++, using the toolkit VTK[1] for handling and rendering the visualizations dataset.

The server is implemented as an endless loop. At each step it performs a non-blocking read operation from a socket, looking for a new message from the tablet, and a render call for the dataset. This loop continues until the tablet sends an stop command.

Tablet Implementation

The tablet is a Google Nexus 10 with a 10.1" screen, runs Android 4.3 as its operative system and has a built-in gyroscope. The software on the tablet uses VES[2] which is a port of VTK for mobiles.

Part of the application is programmed in Java as an Android Activity, using the application framework for the Android platform. The lower level part interacting with the VES framework is programmed in C++ and linked with the application through the NDK toolset[3].

Networking

The networking component ties the tablet and the server components. Before presenting the implementation, it is important to analyze the characteristics of the environment in order to understand the motives that guided the design of the networking component.

C1 The number of peers connected is trivial: only server and tablet.

C2 The system has to run at interactive rates.

C3 The tablet should be physically disconnected so the viewers can move freely.

Because of **C1**, a simple network design suffices, making irrelevant to implement a robust networking component. **C2** suggests that the communication has to support delivering at least 30 packets per second for keeping the data transformations fluid. **C3** imposes the use of a wireless network. The networking component is then implemented with UDP datagrams, looking for simplicity and performance.

The messages interchanged follow the RVProtocol (see appendix D for more details), which is a simple protocol created for this specific system. Through these messages the tablet sends to the server the value of $\boldsymbol{M}$ and receives from the server the value of $\boldsymbol{E}$.

[1] http://www.vtk.org/
[2] http://www.vtk.org/Wiki/VES
[3] https://developer.android.com/tools/sdk/ndk/index.html

Chapter 4

Evaluation Study

Evaluating a system is a crucial part of its life cycle. The use and evaluation of any system unveils details about the implementation that are difficult to realize otherwise. Just by themselves, the analysis and design phases lack of the low level depth exposed in the evaluation phase, which reveals flaws that should be corrected, or important features that should be enhanced. Therefore, the iterative construction of a system should consider the evaluation as an important phase of the project life cycle.

4.1 Study Design

In the literature it is possible to find different tools for evaluating user interfaces. Formative, summative, and expert guidelines-based evaluations are employed in different stages of the development of an interaction prototype [59].

A formative study with experts is proposed for the evaluation of the system. The main idea of the study is to present the system to a group of scientists that use 3D data on their daily work and observe how they interact with it. Likewise, it is important to directly question the scientists about their preferences and experiences using the system. This way, the study gathers descriptive and qualitative information which can be used for the evaluation of the current state of the system and the planning of future iterations. Other quantitative metrics recollected in the study, are used for comparing the system with other existent approaches that offer solutions for related tasks. Nonetheless, for this type of study, the qualitative data gathered is more valuable than the quantitative [60].

Several elements of the system are prone to be evaluated in the study. For example, it is interesting to know how the participants hold the tablet, what elements of the UI they use the most, what are the most recurrent actions, how they use the gyroscope, among many others. Getting this information involves the planning of tasks and the logging of actions when they are performed.

The first proposed task for the study is related to the exploration. Observing users while they explore a dataset reflects a good insight on the system features, use cases, and issues. Besides of plain observation, all the actions the participants perform could be tracked for statistical evaluations of the times spent in the interactions.

Another task should be directed towards finding how the participants interact in situation where both screens are not in synch. Seeing how they deal with the difference in the views and knowing their preferences about the matter are interesting feedback for the evaluation of the metaphor. Besides the observing tasks, the study also has to contemplate a set of training tasks for introducing the system features to the participants.

All the study tasks should be carried out for each interaction technique, this is, the tBox and FI3D. The purpose of doing this is not to compare the techniques but instead to evaluate in a broader spectrum the combination of touch interaction with a stereo display. Moreover, two datasets are needed, one for each technique, because there could also be learning effects the second time a participant explores the same data.

Finally, some details are easier to capture from the participants through a questionnaire. Asking the participants directly about their preferences and experiences during the test is a practical way of obtaining qualitative information about the system.

4.1.1 Experiment plan

The experiment plan contains the detailed description of what will be done in the study. The script of the experiment plan is condensed in the protocol for the study, that can be consulted in appendix A.1. Figure 4.1 presents the setup of the experiment room.

Experiment introduction

The participant arrives to the room. The evaluator explains the participant the general goals of the experiment and then presents an initial questionnaire for capturing the information about any previous knowledge and experiences they might had. This information helps to categorize and characterize the participants, the summary of the results of the initial questionnaire can be found at the annexes appendix C. In this initial stage, the participant is given a unique ID that will identify the information gathered in the current session. After the finishing the questionnaire, the eye-height and the eye separation of the participant are measured. These two values in conjunction with the position read from the head tracking let approximate the eye position of the viewer.

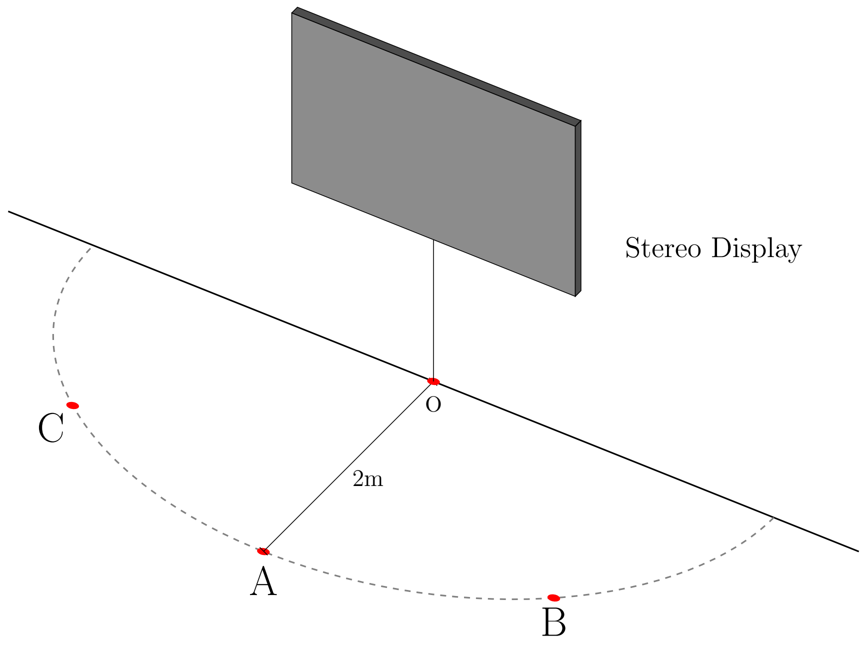


Figure 4.1: This figure describes the setup of the experiment room. The point O corresponds to the center of coordinates used throughout the experiment.

Interaction Technique introduction

The participant is then asked to move to point A, where they receive the tablet and a pair of shutter glasses. The training dataset is loaded in both screens guaranteeing an initial synchronization. After verifying that the glasses are working normally and the participant can see the stereographic image, the evaluator starts with the introduction of the first technique.

The introduction of an interaction technique comprises the explanation of each action available for navigation. First rotation, then translation and scaling. The gyroscope is introduced once, with the first technique, after explaining how to do rotations with the widget. After showing each action, the participant is asked to reproduce a predefined view that can only be achieved by using the actions previously introduced. After explaining all the actions, the participant is asked to freely play with the system some more minutes. Lastly, half of the participants are introduced with the tBox first and the other half with the FI3D first, in order to counterbalance any possible effect that could appear when a technique is presented before the other.

Exploration

Subsequently, the participant is presented with the first dataset and asked to find an interesting view within it. While the participant explores through the dataset is encouraged to discuss and talk aloud, which is a good practice suggested by Harston [60]. The whole task is performed in front of the screen from the point A as shown in fig. 4.1 with both screens in synchronization. The participants finish this task when they find an interesting view. At this point the evaluator asks the participant to memorize the view because in the next step will have to reconstruct it.

Reconstruction

At this step the participant is moved to the point B of the setup. The evaluator explains that the stereo screen is now going to be showing the dataset from the new perspective but the tablet will remain presenting the dataset from the previous viewpoint. The participant is asked to reconstruct the view achieved in the previous step, and also encouraged to talk aloud and express any difficulty experimented while trying to do it. The precision in this task is not very important because the participant memory is not the variable evaluated. Instead, a rough view may suffice.

When the participant finds what appears to be the previous view, the evaluator asks them for any difficulty found (in the case that any problem has not been mentioned yet) and introduces the synchronization mechanisms, "stereo to tablet" and "tablet to stereo". After a quick explanation of each, the participant is prompted to press one of the two buttons putting the screens in synchronization.

Presentation

The participant is again moved, this time to point C, where the offset angle between both screen views is larger, given that they were synchronized at point B. The task at this place is trickier than before and a trivial solution is not possible. The participants are asked to recreate the preferred view for someone who is located at A. This obligates them to imagine the offset C-A but to first handle the offset B-C, getting in synch first. Just as the previous steps, the task ends when the participant is confident of the view achieved and the precision is not considered as an important factor.

When the task is completed, the participant goes back to point A and the same tasks are repeated with the other interaction technique and a diferent data set. The experimental part finishes after the second cycle is over.

Post-Experiment

When the interaction tasks are completed, the participant is invited to fill in a questionnaire, asking about their preferences, difficulties and thoughts about the system. Once the questionnaire is finished, the experimenter invites the participant to a short interview with open questions that reflect their impressions about the interaction techniques and the system in general. Comments and suggestions are also welcomed during the interview.

4.2 Study Execution

The study took place between November 27th and December 2nd 2013 at the installations of the DIGITEO building Orsay, France. The experiments had an approximate duration of 1 hour, with 30 min. for setting up the room for the next session.

Hardware used

A 3D TV of 55 inch diagonal with a resolution of 1920 x 1080 pixels is used as the stereoscopic screen. The TV comes with a set of active shutter glasses. A primitive way of head tracking is implemented with a Microsoft Kinect. The tablet is a Google Nexus 10 (pixel resolution: 2560 x 1600, display diagonal: 10.1 inch = 25.5 cm, spatial device size: 263.9 mm x 177.6 mm x 8.9 mm, weight: 603 g).

Participants

For the study execution two groups of experts were selected: four voluntaries 2 male, 2 female, from the field of Structural Biology, working with molecules, with IDs M1-M4 and other four males from the Fluid Mechanics field, with IDs F1 - F4, working with flow data. Participants' ages had a median of 31.5 years and with a median of 9.5 years of experience at their respective fields. None of the participants were paid for the experiment.

Datasets

Two datasets from each field of expertise were selected. The Structural Biology participants were presented with a balls-and-sticks representation of the tomato aspermy virus protein 2b (PDB: 2ZIO) and the E. coli WrbA holoprotein (PDB: 3ZHO). A streamline integration of the flows was the type of representation selected for the fluid mechanics. The datasets for them were a snapshot of a flow over a thick plate and the atmospheric air motion over a region in Europe. The Stanford Bunny was selected for the training/introductory dataset of both groups. The datasets can be consulted in appendix B.

Supporting Software

A study controller was programmed in order to manage the whole experiment. This software implemented all the steps of the experiment, giving the experimenters control on the tablet, allowing them to remotely change the datasets and UI elements according to the current step. The controller also received all the actions performed by the participants and save them into a log. At every moment there were at least two examiners in the room, one using the study controller and other taking notes and observing the participants.

Records

All the interaction gestures were captured in a log of the Study Controller indicating the time in milliseconds and the parameters of the action. An example of the logs read from the user can be seen in figure 4.2. The test was also video recorded with the participant consent for fair use of the footage, primarily for further analysis and closer observation. Likewise, the final interview was also recorded on video. The platform Google Forms[1] was used for recompiling the questionnaires information.


```
[{
  "participantTag": "F1",
  "eyeHeight": 1.71,
  "typeDataset": 2,
  "interactionStarter": 0,
  "type": 40,
  "timestamp": 1385556486066
},
{
  "mode": "Frame",
  "type": 102,
  "timestamp": 1385556494849
},
{
  "type": 47,
  "timestamp": 1385556497657
},

//Rotations
{
  "dx": 3.494995,
  "dy": -16.415466,
  "type": 56,
  "timestamp": 1385556874242
},
{
  "dx": 9.383301,
  "dy": -53.055725,
  "type": 56,
  "timestamp": 1385556874288
},


...

//Zooming
{
  "factor": 0.9954521,
  "type": 55,
  "timestamp": 1385559732699
},
{
  "factor": 0.9998912,
  "type": 55,
  "timestamp": 1385559732699
},
{
  "factor": 0.99498695,
  "type": 55,
  "timestamp": 1385559732699
}

...
```


FIGURE 4.2: Example of the test log.

4.3 Results

This section presents a summary of the results gathered during the experiment. The results include qualitative answers of the participants and relevant statistical analyses performed on the interaction data.

[1] Google Forms is a service provided by Google that lets people make formularies online and obtain the information as spreadsheets

4.3.1 Usability and Learnability[2]

The participants expressed a good opinion regarding the general usability of the system. They were asked if they could do what they wanted and the answer was positive: median of 4 (in a scale of 1 meaning "I completely disagree" to 5 "I completely agree") and standard deviations (SD) of 0.52 (FI3D) and 0.89 (tBox). Similarly when asked if it was clear how to achieve certain views, the median answer was 4 with an SD of 0.71 for both techniques. Also, to the question about the system allowing them to achieve the goals quickly, the participants answered with a median of 4 and SDs of 0.53 (FI3D) and 1.60 (tBox). Additionally, M3 expressed a favorable view regarding the usability of the system: As a smartphone user she never did any molecular visualization in handhelds, but found interesting how she quickly interiorized how to apply her already known gestures (e.g., pinching to zoom) in the system.

Regarding the ease of learning how to work with the system, the study registered similar answers. Participants agreed that the system could be used without much explanation, scoring 4 as the median with SDs of 0.83 (FI3D) and 1.16 (tBox). Finally, to the question about the system requiring a lot of mental effort to use the mean answer was 4 with an SD of 0.74 (FI3D) and 0.93 (tBox).

4.3.2 Dual Screen Interaction

The participants also expressed their opinions about having two screens presenting the data from different perspectives. Participant F2 said he sees the separation of views as a good thing for scientific applications. F3 also said that it is better to have two different perspectives because a duplicated view would not be useful. On the same matter, M3 also expressed to like the separation, and sees another possible use for it when she wants to show something to others. F1 liked that the tablet is not head-tracked and instead remains static, having the handy buttons for synchronizing the views when needed.

During the experiment the participants were changing their focus from the tablet screen to the stereo screen and viceversa, so they were asked how they felt about having to do this. None of them reported having any problems with looking back and forth. However, a pattern was noticed during the interaction: most of the participants (M1, M2, M4, F2, F3, F4) tend to focus primarily on their tablet at first, but soon changed their focus to the stereo screen when presented with a dataset.

[2]Usability and Learnability are two system qualities defined in the ISO/IEC 9126 usually interpreted as usability only. In this context Usability refers only to the "ease of use" and Learnability to the "ease of learn".

Nonetheless, some of them said to have a preference for the interaction where looking at the tablet was not required. M2 said that she liked that FI3D let her focus more on the TV, using the tablet only as an input device. With the same reasoning, M4 said that he liked the pinching gesture of FI3D because he could do it while focusing on the stereo screen. F2 found simpler the rotations in FI3D because it did not require him to look at the tablet. And finally, F3 preferred FI3D because with the tBox he had to look at the tablet looking for where the edges or the center.

Regarding to what was presented in the tablet, only participant M4 gave a different possibility, saying that the tablet could show a minimap of the dataset and mark the location of the viewer inside of it instead of the current render. In that way, the tablet could be used as a guide for exploring more complex datasets.

4.3.3 Interaction techniques

Even though the main purpose of the study was not to make a comparison and declare a “better” interface, it is interesting to know how participants perceived and used each interaction techniques presented in the system.

tBox

The tBox was praised for the simplicity of its design compared with FI3D. F1 said that he liked having only one control in which he could center. M1 mentioned that the restriction that the tBox imposes on the axes can sometimes be good because it provides more precision in the transformations. M2 felt that the tBox was better for achieving a specific view rather than free exploration.

However, other participants indicated some difficulties with the tBox. F1 felt that the tBox has an implicit order of operations, being implicitly necessary to do the rotations first and then the translations. M1 found limiting the impossibility of a direct rotation around the z-axis with the tBox. About this same topic M4 found cumbersome to do a z-axis rotation, having to rotate a different face before the intended. M2 also noticed this and said that the tBox requires more planning.

FI3D

The participants also showed a good valoration for the FI3D technique. When asked about the preference after the test, six of them preferred FI3D instead of the tBox. Among other remarks, F2 liked that the order of the interactions was not important in FI3D when compared with the tBox. Others (M2, M4) found similarities between the

FI3D gestures and PyMol[3], which help them to grasp quickly the mechanics of FI3D. Regarding the purpose, M1 and M2 noted how FI3D provided an easier interface for free exploration.

Nevertheless, some participants pointed out the drawbacks and limitations they found with FI3D. F1 found "tricky" to perform a z-Rotation with the interface, because the same bar also starts a translation, forcing him to pay more attention at the moment of doing the gesture. He also felt that the interface was convoluted and presented too many elements combined in the same interface. M2 found precise rotations more difficult to do with FI3D when compared with the tBox.

Gyroscope

Regarding the gyroscope, the participants expressed diverse perceptions. F1 thought that the gyroscope was not necessary and he felt it artificial compared with the touch screen which was more natural for him. Similarly, F3 did not feel a real advantage with the gyroscope.

In contrast, M1 found that the gyroscope simplified his rotations. M4 and F1 used the gyroscope more when combined with the tBox than with FI3D, feeling that the rotations were easier with the gyro than with the tBox. M3 supported that same idea even though he did not like the gyroscope much. F4 and M2 said that the gyroscope helped them to do rotations more precisely than with the tBox.

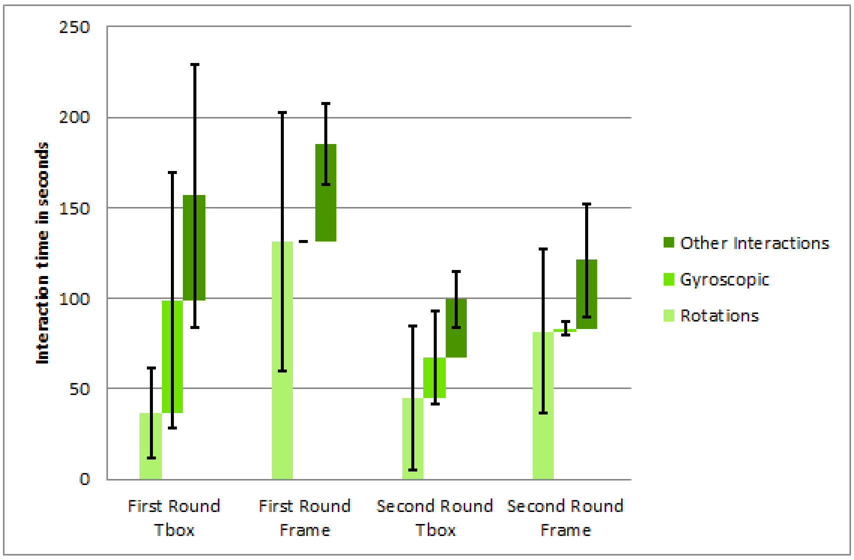
4.3.4 Interaction evaluation

Using the logs recorded during the study, it was possible to extract the interaction information in the form of interaction times. The log records were grouped in clusters according to the participant action and the difference of time with other records. If two records corresponded to the same task and had close timestamps, were put in the same cluster. The new sort of the data served as input for a quantitative study of the interaction. Figure 4.3 shows these frequencies with their respective confidence intervals.

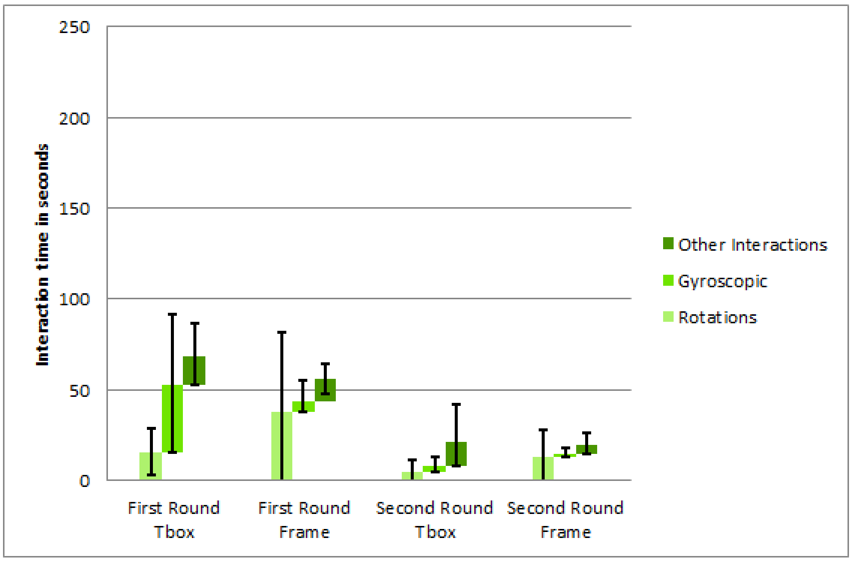
Gyroscope Combination

During the exploration step there was possible to notice how the gyroscope was used more with the tBox that with FI3D. Running a t-test with the percentage of gyroscope used from the tBox and the FI3D exploration time, it was possible to find a significant difference ($t(7) = -2.502, p = 0.041 < 0.05$; with an effect size calculated with Cohen's

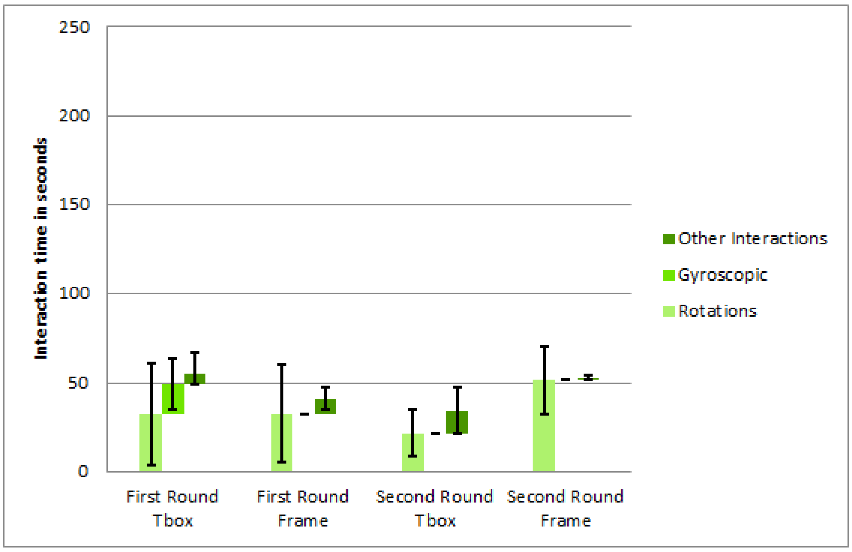
[3]PyMol is an opensource molecular visualization system. http://www.pymol.org



(a) Interaction at A



(b) Interaction at B



(c) Interaction at C

FIGURE 4.3: Interaction times at each point of the test, distributed by technique, and round. The interactions are subdivided by rotations, use of gyroscope and other rotations (scale and translation).

	Levene's test		ANOVA	
viewpoint	$F(1,14)$	p	$F(1,14)$	p
A	1.278	0.278	4.56	0.051
B	0.019	0.964	2.435	0.141
C	0.234	0.636	0.819	0.381

Table 4.1: Summaries of the ANOVA tests ran for discarding the effects of the difference in datasets on the interaction time.

$d = 0.844 \in [0.035, 1.692]$), being greater the average percentage of time with the tBox: 29.730% vs 0.724% with FI3D.

Synchronization mechanisms effect

After the synchronization buttons were available, the participants used them when changed to another viewpoint for finishing the tasks. The effect of the synchronization mechanisms can be appreciated when comparing the first and second interaction times at point B with an ANOVA. Before running the ANOVA test, it is important to check if the variances of both group of values fulfill a homogeneity condition. A Levene's test did not show a violation of the variance homogeneity assumption ($F(1,14) = 1.365, p = 0.252 > 0.05$). Then an ANOVA proved the significant effect ($F(1,14) = 4.909, p = 0.044 < 0.05$, effect size $\eta^2 = 0.260 \in [0, 0.536]$).

Zoom Possibilities

FI3D gives the possibility of changing the apparent size of a visualized object with the zoom, scale and z translation operations. Even though they offer similar results, their use rates are quite diverse. The zoom was not used by any participant at the exploration stage, so it is excluded from the comparisons. A t-test shown that scale, triggered by pinching, was used significantly more than the z translation, triggered by the right bar ($t(7) = 2.043, p = 0.040 < 0.05$; Cohen's $d = 3.092, \in [1.353, 4.807]$)

Non-Effects

During the tests, the participant F4 felt that the second dataset was more complex than the first and that it could have an effect on the study. This motivated to run several ANOVA tests in order to discard any potential effect. Table 4.1 shows the results of the respective Levene's tests and ANOVAs comparing the times spent in each dataset and viewpoint. There was not significant effects for any viewpoint.

Finally, it was also interesting to find if the technique —tBox or FI3D— had any effect in the interaction time. Whit this purpose, another set of ANOVA tests was employed, and their results can be seen in table 4.2. Again, there was not any significant effect.

	Levene's test		ANOVA	
viewpoint	$F(1,14)$	p	$F(1,14)$	p
A	0.0038	0.9518	0.4134	0.531
B	0.0107	0.919	0.102	0.754
C	0.234	0.636	0.312	0.585

TABLE 4.2: Results for the ANOVA tests looking for any potential effect of the technique on the interaction times.

4.3.5 Post-Interview

At the end of the test the participants were also asked about their experiences and any new feature or suggestions for the system. None of them presented any visual nor corporal fatigue, that could be caused by watching stereoscopic images or holding the tablet during the whole experiment.

Improvements

Something interesting realized during the experiment was that the participant F3 tried to use the gyroscope for translations. He intuitively thought that any physical translation applied on the gyroscope would also apply on the data, so he tried to pull the gyroscope towards him with the intention of translating the data along the z-axis. This did not work because that feature was not programmed in the current state of the system, but the participant did suggest it at the end of the interview.

The participants M3 and M2 discussed about the possibility of bringing collaboration capabilities to the system. The interest for sharing with other colleagues was also expressed by F4, suggesting to add a way of sharing the views. With a similar idea M5, F4 and F3 wanted that the system let them to store views.

Another idea was resetting the axis alignment in the tBox, which currently is programmed to always conserve the data axes. That idea was mentioned by M1 and M3. In a similar fashion, F4 and M3 had the idea of changing the area of interest of the view, for example by moving the cutting plane.

Finally, the participant F3 said that he would like to have a music stand for placing the tablet and being able to interact with both hands at the same time. He also noticed how sometimes when he was interacting without looking, his fingers went out of the interaction area. He suggested that the vibration motors of the tablet could be used as a signal for alerting the viewers when they are performing a gesture close to the borders.

4.4 Results Discussion

The study results provided a good insight on the points that were aimed by the system, providing a valuable feedback for the next steps of the project. Following, there is a discussion of the main results found with the experiment.

Regarding the dual screen interaction, there are several points worth to mention. The first thing noticed is the focus that the participants put on each screen. Based on our observations, it is possible to categorize the focus/attention distribution of the participants on the screens in a small framework. First, at the moment where the user is learning the interaction mechanic or widget, there is an **exclusive focus in the tablet** with only small glimpses on the stereo screen to confirm or validate the result of the interaction. Next, after getting more confidence in their actions, they **look at the tablet for starting the gesture**, quickly directing their sight to the stereo screen to continue and finish the interaction. Finally, in some cases, there was a **predominant focus in the stereo screen**, looking at tablet only when it is unavoidable. This was possible because of the availability of elements that does not require to look at the tablet, such as the RST interaction of the FI3D or the gyroscope.

In the same matter, it was intriguing to note that the participants valued and considered important not having to look at the tablet while interacting, even to the point of expressing their preferences for a technique because it allowed them to do this. But at the same time, none of them complained for having to look back and forth between screens. According to this, even though it is not required, a blind touching input would be a “nice to have” feature that avoids looking back and forth. Nonetheless, this would also reduce the sense of control gained with presenting the data in the tablet for the touching interaction, provided that the viewer does not know exactly which part of the data is touching.

Having the synchronization mechanisms in the system is also perceived as a good thing. The possibility of going sync again on demand, was appealing for the participants who found them useful and a complement in the stereo eye-tracking environment presented. Even though these buttons support an obvious way of interacting, which is directly setting a view from the other, they also open more interaction possibilities. For instance, all the interaction could take place in the tablet and setting the view in the stereoscopic screen for a better examination.

From a different perspective, when comparing the two widgets, the study did not reflect a clear preference for any of them. Also, not even a statistical difference when the interaction times for completing the task where compared. Instead, it becomes clear that each widget has a more specific usage. While FI3D was found to be more suited for free exploration, the tBox shown its strength for performing restricted transformations, useful when there is a specific view that the viewer wants to achieve.

Finally, considering the gyroscope, it looks like it might be a good complement for the tBox. The gyroscope could simplify some rotations that otherwise would be more complex and involve more planning. This fact also seems to be supported by the quantitative data, which reflected a significant difference of the percentage of time using the gyroscope combined with each widget technique, being greater when used with the tBox than FI3D.

Chapter 5

Conclusion

5.1 Work Synopsis

The current work presented the design, implementation, and evaluation of a prototype system for scientific visualization with stereoscopic screens using touch interaction. The work started by introducing the process and the alternatives for generating stereoscopic images, the hardware required for deploy them and the different technologies that support the interaction with these screens. That study led to the conclusion of the need of head-tracking and the application of an off-axis camera for the stereo render. Among the interactive techniques found, the touch-based approaches were of great interest because they could complement the visual immersion provided by the stereo screens with an interaction immersion achieved by giving a sense of control over the data to the viewers.

However, touching directly a stereoscopic screen brings a series of visual perception issues that break the immersion or introduce ambiguities to the interaction. This motivated the proposal of a design that separates the touch-capable input from the stereoscopic screen. From the literature some works presented systems that were not flexible enough or with a very specific application. Other works proposed the use of tablet PCs, which were the technology that ended up being selected. Then, there was an analysis of the possible uses that the tablet could have within the system. A set of questions arose from that decision: What to display in the secondary screen? How to receive the viewer input? What other sensors have the tablet that could be used for the interaction?

For answering those questions the design of the system followed the metaphor of an "interactive digital photo". In this metaphor, the tablet acts as a camera with its screen showing a photo of the data that can be manipulated, transferring all the transformations applied on it to the real data. A good way of receiving the user input was achieved through the use of widgets such as the tBox and FI3D, which were already employed for 3D interaction in other approaches. And finally, the tablet gyroscope is also proposed as another input of rotations to the system.

Because the touch interaction requires a steady view of the data and the image of the stereo screen depends on the viewer location, it was decided that both screens could go unsynchronized. Following, the work introduces the interaction rules in the unsynchronized scenario and the mechanisms for synchronizing the views again.

5.2 Main Contributions

The present work tackled the design and implementation of a stereoscopic visualization system. Accomplishing this goal required the development of four important stages: the evaluation of the related work, the analysis of the problems brought by the interaction, the design of the solution and the analysis of the results. Each of these stages provides a contribution to the overall goal.

In the evaluation of the related work the literature suggested that the off-axis camera model was more accurate than the toed-in camera model because the deformation that the latter brings to the data. Also, from the different interaction approaches, the touch based was interesting because its low curve of adoption, possible universal solution, sense of control on the data and flexibility of the gestural design, making it a good candidate for interacting with 3D data.

The clash of perception cues that appear when the hands try to touch a stereographic screen, makes unpractical the incorporation of touch interaction in a stereoscopic environment. Nonetheless, it still possible to separate the touch area from the visualization area, remotely linking the touch gestures to the data, as it normally happens with a mouse interface. Tablets are found to have the desired features for handling the users' input to the system.

With the addition of a new screen to the system, it is now important to think about what to put in it and how to teach it to the users. Because of this, the *interactive digital photo* metaphor is introduced in the system as a result of adopting other ideas from the Augmented Reality field into the Scientific Visualization domain. That metaphor also gives a general guideline for the design of the interaction in the system.

Finally, after the system is evaluated, several points are found from the behavior of the participants:

- Looking back and forth is not a big problem, but there is a preference for not having to look at the tablet screen in the current setup.
- The focus of the participants in each screen can be categorized in a framework that accepts three states: exclusive focus on the tablet, tablet glimpse for starting the interaction and predominant focus on the tablet.
- It is useful to have two screens presenting the same dataset from different points of view, with the option of synchronizing them on demand.
- FI3D and the tBox seem to complement each other. One's weaknesses are filled up by the other's strengths.

5.3 Future Work

As mentioned by some users, it would be interesting to design an eyes-free interaction technique. Using simple touch gestures that do not require to position the fingers in a specific point of the screen, plus the implementation of dead-reckoning in the tablet (reading its rotational state and its position within the room) could quickly add up to seven degrees of freedom. In that case, the tablet would act as a handle that could transmit every physical motion detected to the dataset.

Implementing a more robust head-tracking method for a better approximation of the viewer's eyes could also benefit the overall quality of the stereoscopic effect.

And finally, it is also interesting to analyze to what extent is possible to have a collaborative environment. Presenting an adequate stereo image for different points of view seems to be a hardware issue, but the design of the interaction with several viewpoints at the same time opens a myriad of interesting possibilities.

Appendix A

Evaluation Material

A.1 Study Protocol

[1 hour] Before Study Begins: Load Data, Input Eye Height, Test each tablet interface, clean tablet, clean glasses, test network, check that laptop has internet and survey loaded.

[3 min] Welcome!

Introduce research team, You can put down your bag and coat over [here]. Do you anything before we begin? Also – do you prefer an English or French keyboard? Close door after participant arrives.

[5 min] Consent Form

See Study material: "Consent to participate in research" and "Research Media Records Release Form".

[5 min] Introductory Questionnaire:

See Study material: "Touch Interaction with Stereoscopic Displays: Survey".

Give participant the tablet.

[5 min] Stereoscopic Display Introduction: Training Dataset

Today, you'll be using different tablet interfaces to help you interact with data on a stereoscopic visualization system. You will wear shutter glasses and see 3D datasets like in a 3D movie. We've hard-coded the stereoscopic display to work for someone of your height at three specific points in the room, marked with tape on the floor.

- Please stand at each point and verify that the stereoscopic display is functioning properly.

If at any point you notice something strange about the 3D system, or the glasses, please let us know – we may need to reset the stereoscopic display or restart your glasses. And remember that it's okay if you're having difficulty or make mistakes – we're interested in testing whether or not the technology is effective and understandable. Please let us know if you have problems so that we can identify ways to address them in the next iteration.

[15 min] Interaction Technique 1: Introduction Training Dataset

We will have you use two main interaction techniques, along with a few extra features that will be present in all conditions.

See Study material: appendix A.2 "tBox introduction", appendix A.3 "FI3D introduction"and appendix A.4 "Gyro introduction"

[10 min] Study Task A (Video/Audio Record Session + Observation Notes) [Reproducing viewpoint]

Part 1

This data represents... See Study material: appendix B "Dataset descriptions"

Standing at point A, find a view of your data that represents an interesting phenomena or feature of this dataset. When you've found a good view, say ["finished"]. During the exercise, please "talk aloud" and let us know if or how you're having any difficulty. Could you describe the interesting feature that you see in this view? Capture view with Study Controller.

Please take a moment to remember this view. We will ask you to recreate it in the following tasks. [Flip the screen to black]

Part 2

Move to point B; While your stereoscopic view of the data now showing you the view from Point B, you'll notice that your tablet still has your previous view from Point A. Using the tablet interface, please recreate the same stereoscopic view your data that you had at Point A.

When you think you've recreated your view, say ["finished"]. Please "talk aloud" during the exercise, and let us know if or how you're having any difficulty.

[Set timer for 5 minutes. If no progress has been made when the timer goes off...]

- Did you have any challenges during this task?
- Were you able to remember your view from Point A?
- How do you wish the interface could help you with this task?

[3 min] "Tablet to screen and Screen to tablet" Introduction We implemented an additional feature — "Tablet to screen and Screen to tablet" -– which addresses the issue of synchronizing the stereoscopic view and the view on your tablet.

- "Tablet to screen" places your tablet view on the stereoscopic display at the point where you're standing.
- "Screen to tablet" puts the stereoscopic view on your tablet depending on where you are in the room.

Part 3

From here, recreate the view that you had as if you were to show it for someone situated in A. When you've found the good view, say ["finished"]. During the exercise, please "talk aloud" and let us know if or how you're having any difficulty.

Do you need to take a break or put down the tablet for a bit?

[5 min] Interaction Technique 2: Introduction: Training Dataset

[10 min] Study Task B (Video/Audio Record Session + Observation Notes)

For the second study task, we will do a similar set of tasks as before, but this time with the second interaction technique. We will keep "Screen to tablet" and "Tablet to Screen" available so that you can continue to use them

This data represents… See Study material: appendix B "Dataset descriptions".

Part 1

Standing at point A, find a view of your data that represents an interesting phenomena or feature of this dataset. When you've found a good view, say ["finished"]. During the exercise, please "talk aloud". Could you describe the interesting feature that you see in this view?

Please take a moment to remember this view. We will ask you to recreate it in the following tasks. [Flip the screen to black]

Part 2

Move to point B; recreate your previous view of your data. When you think you've recreated your view, say ["finished"]. During the exercise, please "talk aloud". From here, find another view of your data — different from the view at point A — that represents an interesting phenomena or feature of this dataset. When you've found a good view, say ["finished"]. During the exercise, please "talk aloud" and let us know if or how you're having any difficulty. Could you describe the interesting feature that you see in this view? Capture view with Study Controller.

Part 3

From here, recreate the view that you had as if you were to show it for someone situated in A. When you've found a good view, say ["finished"]. During the exercise, please "talk aloud" and let us know if or how you're having any difficulty.

[5 min] Post-Study Questionnaire: See Study material: "Touch Interaction with Stereoscopic Displays: Post-Questionnaire".

[10 min] Interview Questions (need 2 versions of the questionnaire with the names of the respective 1st and 2nd technique)

Thank you!

After Study: plug in stereoscopic glasses for charging, download data from video camera, and download log files

A.2 tBox Introduction

The tBox is a graphical widget that lets you manipulate data visualization via touch interaction. The tBox appears as soon you touch the view area in the screen. [if they haven't already, instruct the participant to touch the screen to make sure they can see the tBox]. You can do 3 different manipulations: Rotation, Translation and Zoom.

Rotation You can rotate the view by touching a face of the tBox with one finger, and dragging. You will see the face highlighted in purple. The data will rotate around the axis perpendicular to the touched face's normal, and the line traced by your finger drag. The rotation axis is selected based on the initial motion of your finger; after you move your finger a few millimeters, your rotation will be set to that axis. Please play with some rotations; be sure to pay attention to both the tablet and the stereoscopic screen in front of you.

Translation The view is translated when you drag your fingers along the edges of the tBox. When you touch an edge, a cylinder appears indicating the translation direction. The actual translation will be a projection of your dragging on the selected edge.

Zoom The Zoom manipulation is a multi-touch gesture that scales your object to be larger or smaller. Put two fingers on any part of the view and pinch them away from each other to make the object larger; pinch your fingers towards each other to make the object smaller.

Feel free to continue playing with the interface until you feel comfortable using it.

A.3 FI3D Introduction

The FI3D is a graphical widget that allows you to explore your data by touching the different tools in the borders of the screen or by freely touching directly on the view.

Rotation You can rotate the view by dragging the bars around the view. Dragging towards the center will allow you to rotate horizontally and vertically. Dragging parallel to the bar will allow you to roll. You can use an extra finger to restrict the axis of rotation for the pitch or yaw. If you simultaneously drag and touch a vertical bar, you will only rotate around a vertical axis. The same will happen with a horizontal bar, restricting the rotation around a horizontal axis. Rolling can also be done by a multitouch gesture, dragging the screen with two fingers.

Translation X-axis and Y-axis translation is done by dragging the view with one finger in the direction of translation. Z-axis translation is controlled by dragging the horizontal bars located at the top and bottom of your view. Dragging down will move your camera back, making the object smaller; dragging up will move the camera forward, making the object larger.

Zoom Zooming can be done in two ways: Dragging away from the buttons located at the corners, which will modify your field of view angle, or by pinching the view area.

A.4 Gyroscopic Rotation Introduction

For both interaction techniques, we added an additional feature to produce free rotations on the view by physically rotating the tablet. To enable this mode you have to hold this button while you rotate the tablet. You can now manipulate the object's orientation by rotating the tablet. Play around with this feature for a few minutes to get a better sense of how it works.

Appendix B

Dataset Descriptions

Training Dataset

The Stanford Bunny

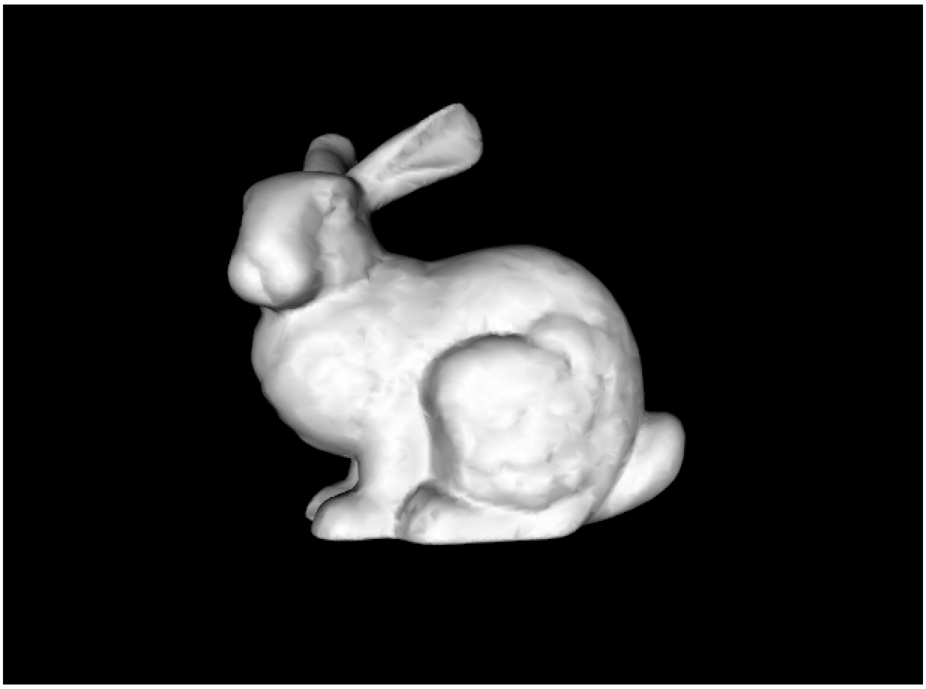


FIGURE B.1: The classic Stanford Bunny.

First Molecular Dataset

Tomato aspermy virus protein 2b. Suppression of RNA Silencing

RCSB entry: http://www.rcsb.org/pdb/explore.do?structureId=2zi0

DOI entry: http://dx.doi.org/10.2210/pdb2zi0/pdb

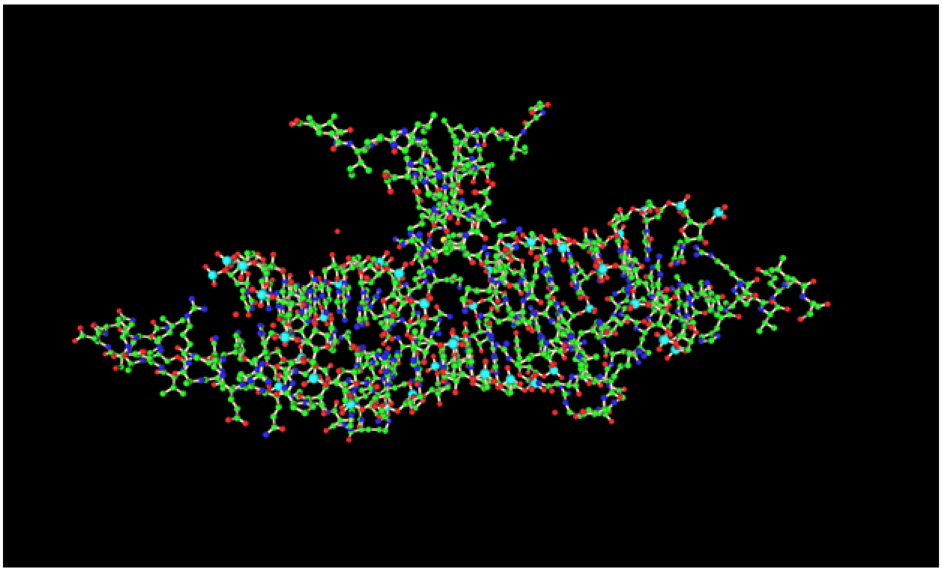


FIGURE B.2: Tomato aspermy virus protein 2b. Suppression of RNA Silencing

Second Molecular Dataset

X-RAY Structure of E.Coli WRBA IN COMPLEX WITH FMN AT 1.2 A

RCSB entry: http://www.rcsb.org/pdb/explore/explore.do?structureId=3ZHO

DOI entry: http://dx.doi.org/10.2210/pdb3zho/pdb

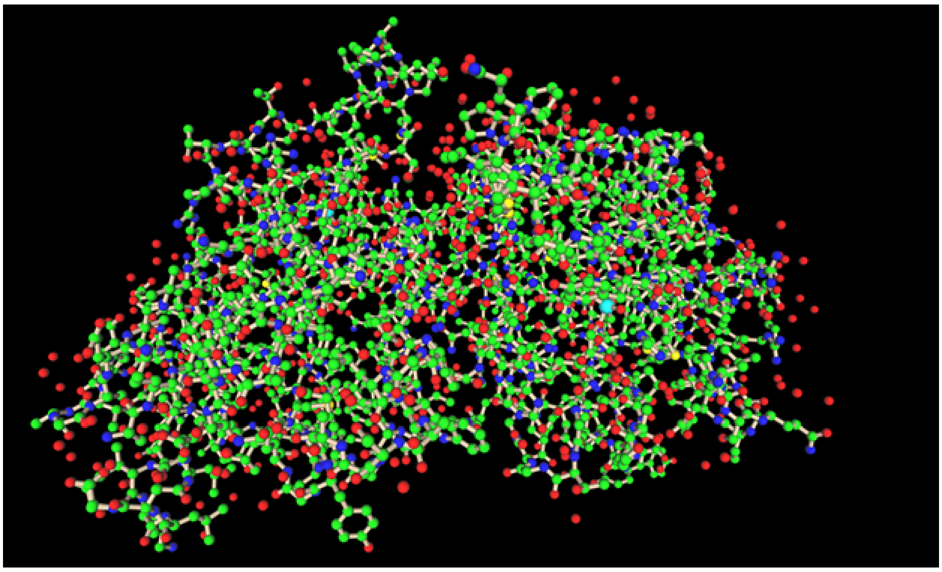


FIGURE B.3: X-RAY Structure of E.Coli WRBA IN COMPLEX WITH FMN AT 1.2 A

First Flow Dataset

Simple time frame of a flow over a thick flat plate. The colors represent the mass density.

Second Flow Dataset

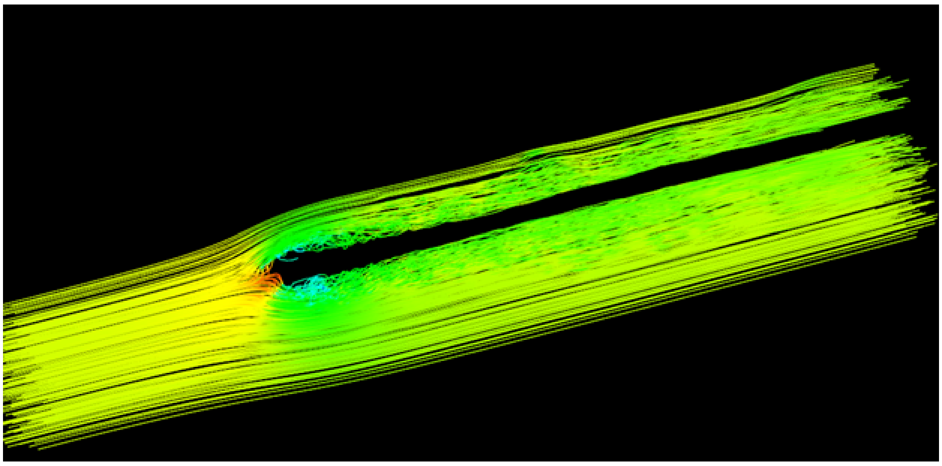


Figure B.4: Flow over a thick flat plate colored by mass density.

Jet stream of the atmosphere in a region of Europe. The colors correspond to the temperature.

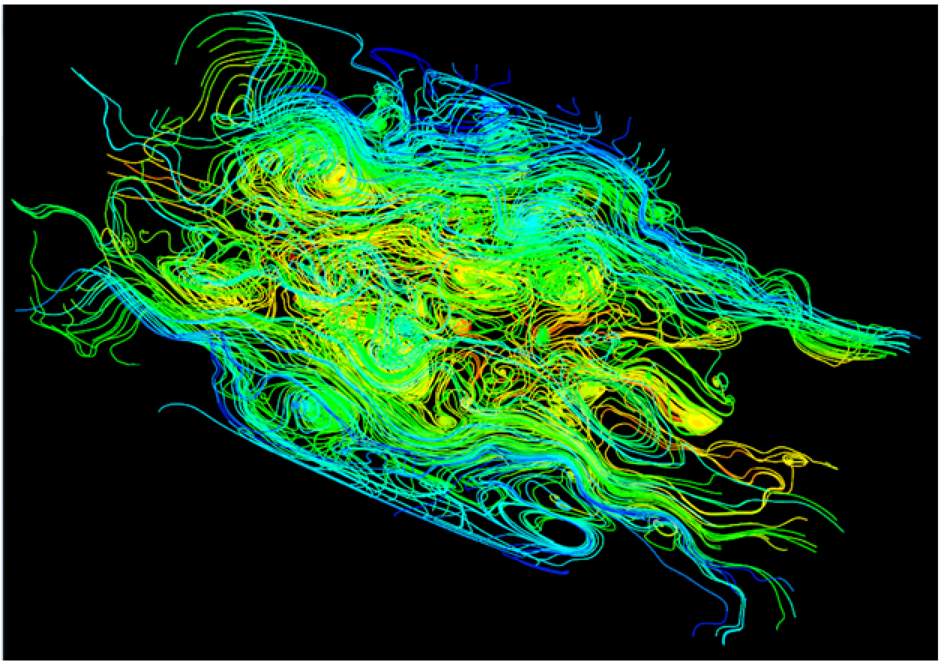


Figure B.5: Jet stream of the atmosphere colored by temperature.

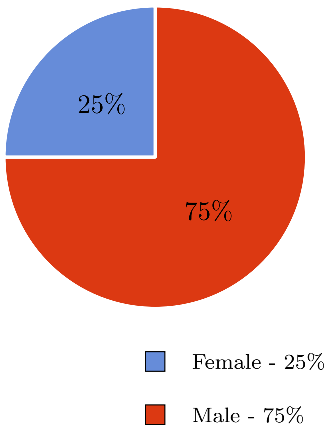
Appendix C

Answers of the initial questionnaire

Age

30, 24, 33, 41, 30, 26, 59, 45

Gender



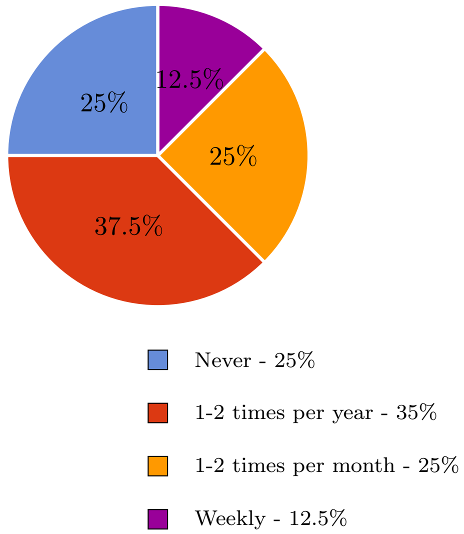
Eye Height

171, 150, 152, 160, 175, 163, 164, 175

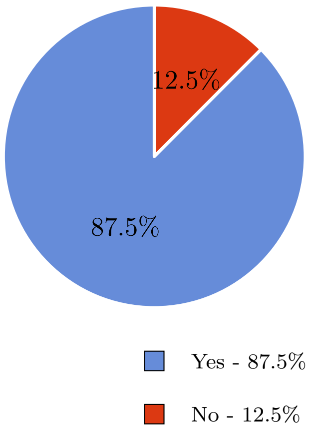
Years of experience in the research field

6, 1, 13, 20, 4, 2, 33, 20

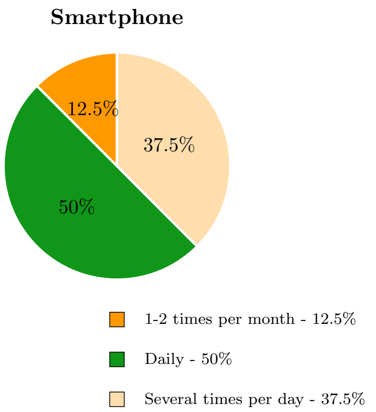
How often do you use stereoscopic displays?

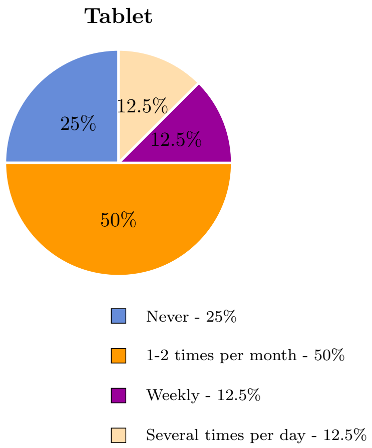


If you have used stereoscopic displays, have you used it to analyze visual data?



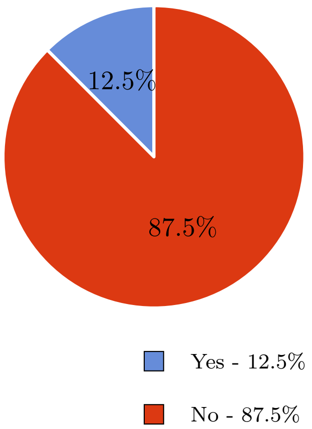
How often have you used touch-devices?



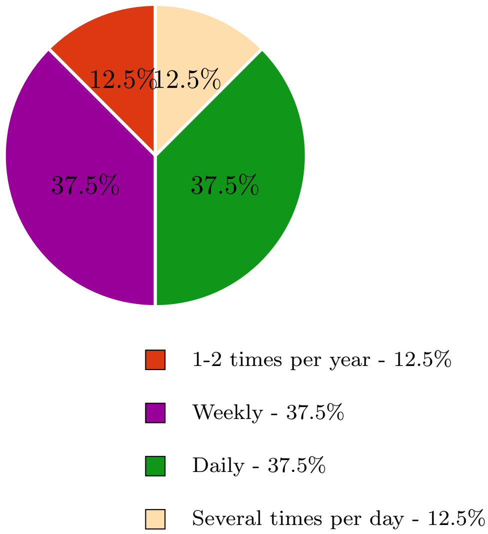


Walls and Large horizontal tables were never used by the participants

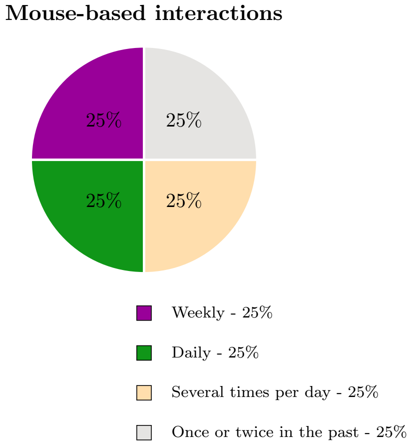
Have you used touch interaction to assist in your normal data analysis practice?

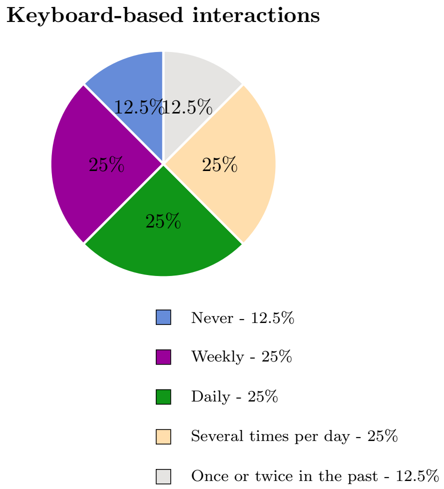


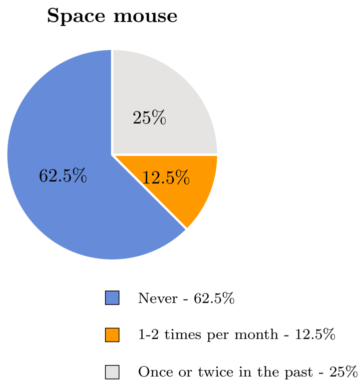
How often do you investigate 3D data in your work?

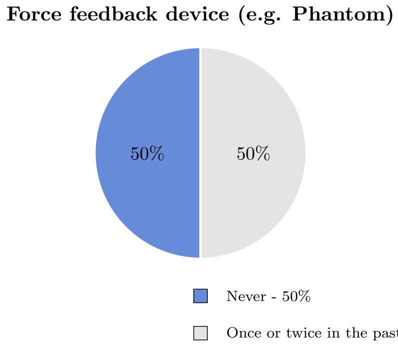


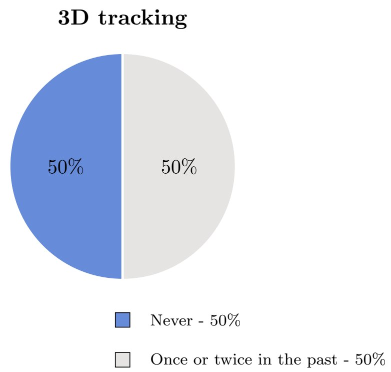
How often do you use other interaction modes to navigate through 3D data?










3D tracking
50%
50%
Never - 50%
Once or twice in the past - 50%

Appendix D

RVProtocol

The RVProtocol (Remote Visualization Protocol) is an extensible application protocol for controlling and following a remote visualization session. It consists on a series of binary messages sent over the UDP network protocol specifying one of two things: an operation and the respective parameters to execute in a server; a response from the server to the client about a previous query.

D.1 Message body

Each message has a maximum length of 256 bytes, with the following layout:

- Byte 0 - 3: Magic Word 0x54F3F00D
- Byte 4: Message identifier. This allows to multiplex the message received depending on the task to execute.
- Byte 5: Message type. It indicates if the message is a query or a response.
- Byte 6: Response required. This byte indicates if the message sender needs to wait for a response or can continue
- Byte 7: unused.
- Byte 8 - 256: Data or parameters area.

The parameters supported are 32 and 64 bits integers and floats following the IEEE 754 single-precision binary floating-point format. All the values are converted to Big-Endian before being sent through the network.

D.2 Client-Server Messages

RV_MSG_CONNECTION. Starts a connection between a client and the visualization server.

RV_SET_USER_POSITION. Updates the location of the viewer's eyes estimation. The parameters are 3 floats with the respective X, Y and Z position. These values are expressed in meters using as reference the setup presented in fig. 4.1

RV_SET_DATA_TRANSFORM. Updates the transformation applied to the visualized data. The transformation is passed as a 4x4 matrix consisting in 16 float values arranged in column major order.

RV_MSG_CAMERA_LOOK_AT. Updates the server's virtual camera, used for the computation of the view. It expects the values for the position (3 floats), focal point (3 floats), and up vector (3 floats).

RV_MSG_LOAD_DATASET. Loads a dataset from the predefined list for the visualization. It receives as a parameter the index (32 bit integer) of the dataset.

RV_MSG_BLACKOUT_SCREEN. Clears the view of the visualization server. The screen can show a view again by changing the virtual camera or loading a new dataset.

RV_MSG_TERMINATE. Ends the connection and the program running the visualization server.

D.3 Experiment Messages

Additional to the Client-Server messages, the study messages are used to control the state of the tablet from a study server, and to register the participant's activities for performing posterior quantitative analysis of the experiment.

RV_MSG_LOG_GYRO_ROTATION registers whenever the viewer performs a rotation with the gyroscope.

RV_MSG_LOG_BUTTON_PRESS indicates when the participant performs a "tablet-to-screen" or a "screen-to-tablet" operation.

Messages starting with **RV_MSG_LOG_TBOX_*** log the operations performed on the tBox: ROTATION, ALIGNED_PAN, and DOLLY_Z.

Messages starting with **RV_MSG_LOG_FRAME_*** log the operations performed on the FI3D: ROTATE_DISPLACEMENT, ROLL_AROUND_POINT, ROLL, PAN, PINCH, RST, MOVE_TOWARDS_FOCUS, and ZOOM.

RV_MSG_LOG_TRAINING_START and **RV_MSG_LOG_TRAINING_FINISH** allow to record the time where each training started and ended.

RV_MSG_CHANGE_PERSON_POSITION allows to move the person to the points A, B or C in the experiment room presented in fig. 4.1.

Bibliography

[1] B. H. McCormick. Visualization in scientific computing. *SIGBIO Newsl.*, 10(1): 15–21, March 1988. ISSN 0163-5697. doi: 10.1145/43965.43966. URL http://doi.acm.org/10.1145/43965.43966.

[2] Can Telkenaroglu and Tolga Capin. Dual-Finger 3D Interaction Techniques for mobile devices. *Personal and Ubiquitous Computing*, pages 1–22, September 2012. ISSN 1617-4909. doi: 10.1007/s00779-012-0594-2. URL http://dx.doi.org/10.1007/s00779-012-0594-2.

[3] Diane Watson, Mark Hancock, Regan L. Mandryk, and Max Birk. Deconstructing the touch experience. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 199–208, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512819. URL http://doi.acm.org/10.1145/2512349.2512819.

[4] Youngsong Cho, Jae-Kwan Kim, Chung-In Won, Joonghyun Ryu, Chong-Min Kim, and Deok-Soo Kim. Betamol: Molecular modeling, analysis, and visualization software based on the beta-complex derived from the voronoi diagram. In *Voronoi Diagrams in Science and Engineering (ISVD), 2011 Eighth International Symposium on*, pages 48–57, June 2011. doi: 10.1109/ISVD.2011.15.

[5] Ralf Kaehler, Tom Abel, and Ji-hoon Kim. Visualization of a high-resolution galaxy formation simulation. In *Proceedings of the 2011 Companion on High Performance Computing Networking, Storage and Analysis Companion*, SC '11 Companion, pages 133–134, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1030-7. doi: 10.1145/2148600.2148670. URL http://doi.acm.org/10.1145/2148600.2148670.

[6] Marcus Magnor, Gordon Kindlmann, and Charles Hansen. Constrained inverse volume rendering for planetary nebulae. In *Proceedings of the Conference on Visualization '04*, VIS '04, pages 83–90, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7803-8788-0. doi: 10.1109/VISUAL.2004.18. URL

http://aplicacionesbiblioteca.udea.edu.co:2141/10.1109/VISUAL.2004.18.

[7] Francesco Volonté, Nicolas C Buchs, François Pugin, Joël Spaltenstein, Minoa Jung, Osman Ratib, and Philippe Morel. Stereoscopic augmented reality for da $Vincii^{TM}$ robotic biliary surgery. *International journal of surgery case reports*, 4 (4):365–367, February 2013. ISSN 2210-2612. doi: 10.1016/j.ijscr.2013.01.021. URL http://www.ncbi.nlm.nih.gov/pubmed/23466685.

[8] Hideo Yamashita, Tatsuya Johkoh, Shinichi Takita, and Eihachiro Nakamae. Interactive visualization of three-dimensional magnetic fields. *The Journal of Visualization and Computer Animation*, 2(1):34–40, 1991. URL http://onlinelibrary.wiley.com/doi/10.1002/vis.4340020108/abstract.

[9] Robert J. Teather and Wolfgang Stuerzlinger. Guidelines for 3d positioning techniques. In *Proceedings of the 2007 Conference on Future Play*, Future Play '07, pages 61–68, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-943-2. doi: 10.1145/1328202.1328214. URL http://doi.acm.org/10.1145/1328202.1328214.

[10] H. Yamanoue. The differences between toed-in camera configurations and parallel camera configurations in shooting stereoscopic images. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1701–1704, July 2006. doi: 10.1109/ICME.2006.262877.

[11] Michael Deering. High resolution virtual reality. *SIGGRAPH Comput. Graph.*, 26 (2):195–202, July 1992. ISSN 0097-8930. doi: 10.1145/142920.134039. URL http://doi.acm.org/10.1145/142920.134039.

[12] Keigo Iizuka. Cellophane as a half-wave plate and its use for converting a laptop computer screen into a three-dimensional display. *Review of Scientific Instruments*, 74(8):3636–3639, Aug 2003. ISSN 0034-6748. doi: 10.1063/1.1592879.

[13] Doug A Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2004.

[14] Miguel A. Otaduy, Takeo Igarashi, and Joseph J. LaViola, Jr. Interaction: Interfaces, algorithms, and applications. In *ACM SIGGRAPH 2009 Courses*, SIGGRAPH '09, pages 14:1–14:66, New York, NY, USA, 2009. ACM. doi: 10.1145/1667239.1667253. URL http://doi.acm.org/10.1145/1667239.1667253.

[15] H. Azari, I. Cheng, and A. Basu. Stereo 3d mouse (s3d-mouse): Measuring ground truth for medical data in a virtual 3d space. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 5744–5747, Sept 2009. doi: 10.1109/IEMBS.2009.5332603.

[16] F. Stenicke, T. Ropinski, G. Bruder, and K. Hinrichs. Interscopic user interface concepts for fish tank virtual reality systems. In *Virtual Reality Conference, 2007. VR '07. IEEE*, pages 27–34, March 2007. doi: 10.1109/VR.2007.352460.

[17] Aaron Bryden, George N Phillips Jr, Yoram Griguer, Jordan Moxon, and Michael Gleicher. Improving collaborative visualization of structural biology. In *Advances in Visual Computing*, pages 518–529. Springer, 2011. URL http://dx.doi.org/10.1007/978-3-642-24028-7.

[18] Miles E. Hansard, Seungkyu Lee, Ouk Choi, and Radu Horaud. *Time-of-Flight Cameras - Principles, Methods and Applications.* Springer Briefs in Computer Science. Springer, 2013. ISBN 978-1-4471-4657-5.

[19] Hrvoje Benko, Ricardo Jota, and Andrew Wilson. Miragetable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 199–208. ACM, 2012. ISBN 978-1-4503-1015-4. URL http://dl.acm.org/citation.cfm?id=2207704.

[20] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber. 3-d gesture-based scene navigation in medical imaging applications using time-of-flight cameras. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–6, June 2008. doi: 10.1109/CVPRW.2008.4563162.

[21] Johannes Schöning, Frank Steinicke, Antonio Krüger, Klaus Hinrichs, and Dimitar Valkov. Bimanual interaction with interscopic multi-touch surfaces. In *Human-Computer Interaction–INTERACT 2009*, pages 40–53. Springer, 2009. URL http://link.springer.com/chapter/10.1007/978-3-642-03658-3_8.

[22] Dimitar Valkov, Frank Steinicke, Gerd Bruder, and Klaus Hinrichs. 2d touching of 3d stereoscopic objects. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 1353–1362. ACM, 2011. URL http://dl.acm.org/citation.cfm?id=1979142.

[23] Otmar Hilliges, David Kim, Shahram Izadi, Malte Weiss, and Andrew Wilson. Holodesk: Direct 3d interactions with a situated see-through display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*,

CHI '12, pages 2421–2430, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208405. URL http://doi.acm.org/10.1145/2207676.2208405.

[24] Li-Wei Chan, Hui-Shan Kao, Mike Y. Chen, Ming-Sui Lee, Jane Hsu, and Yi-Ping Hung. Touching the void: Direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2625–2634, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753725. URL http://doi.acm.org/10.1145/1753326.1753725.

[25] Tilo Westermann. I'm home: Smartphone-enabled gestural interaction with multi-modal smart-home systems. *Informatiktage. LNI*, 9:137–140, 2010.

[26] Harshada Patel, Oliver Stefani, Sarah Sharples, Hilko Hoffmann, Ioannis Karaseitanidis, and Angelos Amditis. Human centred design of 3-d interaction devices to control virtual environments. *International Journal of Human-Computer Studies*, 64(3):207–220, 2006. URL http://www.sciencedirect.com/science/article/pii/S1071581905001576.

[27] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. Multipoint: Comparing laser and manual pointing as remote input in large display interactions. *International Journal of Human-Computer Studies*, 70(10):690–702, 2012. URL http://dx.doi.org/10.1016/j.ijhcs.2012.05.009.

[28] David J Sturman and David Zeltzer. A survey of glove-based input. *Computer Graphics and Applications, IEEE*, 14(1):30–39, 1994.

[29] Ferran Argelaguet and Carlos Andujar. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121 – 136, 2013. ISSN 0097-8493. doi: 10.1016/j.cag.2012.12.003. URL http://www.sciencedirect.com/science/article/pii/S0097849312001793.

[30] G Robinson, JM Ritchie, PN Day, and RG Dewar. System design and user evaluation of co-star: An immersive stereoscopic system for cable harness design. *Computer-Aided Design*, 39(4):245–257, 2007. URL http://www.sciencedirect.com/science/article/pii/S0010448506002235.

[31] Doug A Bowman, Chadwick A Wingrave, JM Campbell, VQ Ly, and CJ Rhoton. Novel uses of pinch gloves™ for virtual environment interaction techniques. *Virtual Reality*, 6(3):122–129, 2002.

[32] F. Argelaguet, A. Kunert, A. Kulik, and B. Froehlich. Improving co-located collaboration with show-through techniques. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 55–62, March 2010. doi: 10.1109/3DUI.2010.5444719.

[33] Nguyen Thong Dang, Monica Tavanti, Ivan Rankin, and Matthew Cooper. A comparison of different input devices for a 3d environment. *International Journal of Industrial Ergonomics*, 39(3):554–563, 2009. URL http://www.sciencedirect.com/science/article/pii/S0169814108001649.

[34] Jonathan Lam, Christopher Collins, Bill Kapralos, Andrew Hogue, and Miguel A Garcia-Ruiz. Wiimote-controlled stereoscopic mri visualization with sonic augmentation. In *Proceedings of the International Academic Conference on the Future of Game Design and Technology*, pages 261–262. ACM, 2010. URL http://dl.acm.org/citation.cfm?id=1920825.

[35] Tobias Isenberg, Mark Hancock, et al. Gestures vs. postures:'gestural'touch interaction in 3d environments. In *Proceedings of the CHI Workshop on"The 3rd Dimension of CHI: Touching and Designing 3D User Interfaces"(3DCHI 2012, May 5, 2012, Austin, TX, USA)*, pages 53–61, 2012.

[36] Alexander Bornik, Reinhard Beichel, Ernst Kruijff, Bernhard Reitinger, and Dieter Schmalstieg. A hybrid user interface for manipulation of volumetric medical data. In *3D User Interfaces, 2006. 3DUI 2006. IEEE Symposium on*, pages 29–36. IEEE, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1647503.

[37] Johannes Schöning, Frank Steinicke, Antonio Krüger, Klaus Hinrichs, and Dimitar Valkov. Bimanual interaction with interscopic multi-touch surfaces. In Tom Gross, Jan Gulliksen, Paula Kotzé, Lars Oestreicher, Philippe Palanque, RaquelOliveira Prates, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2009*, volume 5727 of *Lecture Notes in Computer Science*, pages 40–53. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-03657-6. doi: 10.1007/978-3-642-03658-3_8. URL http://dx.doi.org/10.1007/978-3-642-03658-3_8.

[38] Kenrick Kin, Maneesh Agrawala, and Tony DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of Graphics Interface 2009*, GI '09, pages 119–124, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society. ISBN 978-1-56881-470-4. URL http://aplicacionesbiblioteca.udea.edu.co:2735/citation.cfm?id=1555880.1555910.

[39] Dimitar Valkov, Alexander Giesler, and Klaus Hinrichs. Evaluation of depth perception for touch interaction with stereoscopic rendered objects. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 21–30, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1209-7. doi: 10.1145/2396636.2396640. URL http://doi.acm.org/10.1145/2396636.2396640.

[40] Bernard Mendiburu, Yves Pupulin, and Steve Schklair. Chapter 4 - shooting 3d for broadcast or editing. In Bernard Mendiburu, Yves Pupulin, and Steve Schklair, editors, *3D {TV} and 3D Cinema*, pages 91 – 128. Focal Press, Boston, 2012. ISBN 978-0-240-81461-2. doi: http://dx.doi.org/10.1016/B978-0-240-81461-2.00004-1. URL http://www.sciencedirect.com/science/article/pii/B9780240814612000041.

[41] Martin Hachet, Benoit Bossavit, Aurélie Cohé, and Jean-Baptiste de la Rivière. Toucheo: Multitouch and stereo combined in a seamless workspace. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 587–592, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047273. URL http://doi.acm.org/10.1145/2047196.2047273.

[42] Mike Sinclair, Michel Pahud, and Hrvoje Benko. Touchmover: Actuated 3d touchscreen with haptic feedback. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, pages 287–296, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2271-3. doi: 10.1145/2512349.2512805. URL http://doi.acm.org/10.1145/2512349.2512805.

[43] Dane Coffey, Nicholas Malbraaten, Trung Bao Le, Iman Borazjani, Fotis Sotiropoulos, A.G. Erdman, and Daniel F. Keefe. Interactive slice wim: Navigating and interrogating volume data sets using a multisurface, multitouch vr interface. *Visualization and Computer Graphics, IEEE Transactions on*, 18(10): 1614–1626, Oct 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2011.283.

[44] Jinman Kim, Tom Weidong Cai, Michael Fulham, Stefan Eberl, and David Dagan Feng. 9 - data visualization and display. In David Dagan Feng, editor, *Biomedical Information Technology*, pages 211 – XVI. Academic Press, Burlington, 2008. ISBN 978-0-12-373583-6. doi: 10.1016/B978-012373583-6.50013-X. URL http://www.sciencedirect.com/science/article/pii/B978012373583650013X.

[45] C. Roberts, B. Alper, J. Morin, and T. Hollerer. Augmented textual data viewing in 3d visualizations using tablets. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, pages 101–104, March 2012. doi: 10.1109/3DUI.2012.6184192.

[46] Eric Hamilton and Andrew Hurford. Combining collaborative workspaces with tablet computing: Research in learner engagement and conditions of flow. In *Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual*, pages T3C–3. IEEE, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4418151.

[47] M Malcher and Markus Endler. A context-aware collaborative presentation system for handhelds. In *Sistemas Colaborativos, 2008 Simpósio Brasileiro de*, pages 1–11. IEEE, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4700780.

[48] Jeremy Richards and Patrick E Mantey. Large classroom experience with an interactive tiled display mural. In *Frontiers in Education Conference, 36th Annual*, pages 15–20. IEEE, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4117182.

[49] Ann McNamara, Katerina Mania, Marty Banks, and Christopher Healey. Perceptually-motivated graphics, visualization and 3d displays. In *ACM SIGGRAPH 2010 Courses*, SIGGRAPH '10, pages 7:1–7:159, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0395-8. doi: 10.1145/1837101.1837108. URL http://doi.acm.org/10.1145/1837101.1837108.

[50] Robert T. Held and Martin S. Banks. Misperceptions in stereoscopic displays: A vision science perspective. In *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*, APGV '08, pages 23–32, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-981-4. doi: 10.1145/1394281.1394285. URL http://doi.acm.org/10.1145/1394281.1394285.

[51] Randy Pausch, Tommy Burnette, Dan Brockway, and Michael E. Weiblen. Navigation and locomotion in virtual worlds via flight into hand-held miniatures. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 399–400, New York, NY, USA, 1995. ACM. ISBN 0-89791-701-4. doi: 10.1145/218380.218495. URL http://doi.acm.org/10.1145/218380.218495.

[52] Mark Hancock, Thomas ten Cate, and Sheelagh Carpendale. Sticky tools: Full 6dof force-based interaction for multi-touch tables. In *Proceedings of the ACM*

International Conference on Interactive Tabletops and Surfaces, ITS '09, pages 133–140, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.1145/1731903.1731930. URL http://doi.acm.org/10.1145/1731903.1731930.

[53] Gun A. Lee, Ungyeon Yang, Yongwan Kim, Dongsik Jo, Ki-Hong Kim, Jae Ha Kim, and Jin Sung Choi. Freeze-set-go interaction method for handheld mobile augmented reality environments. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, VRST '09, pages 143–146, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-869-8. doi: 10.1145/1643928.1643961. URL http://doi.acm.org/10.1145/1643928.1643961.

[54] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. Touch projector: Mobile interaction through video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2287–2296, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753671. URL http://doi.acm.org/10.1145/1753326.1753671.

[55] Aurélie Cohé, Fabrice Dècle, and Martin Hachet. tbox: A 3d transformation widget designed for touch-screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 3005–3008, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979387. URL http://doi.acm.org/10.1145/1978942.1979387.

[56] Lingyun Yu, P. Svetachov, P. Isenberg, M.H. Everts, and T. Isenberg. Fi3d: Direct-touch interaction for the exploration of 3d scientific visualization spaces. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1613–1622, Nov 2010. ISSN 1077-2626. doi: 10.1109/TVCG.2010.157.

[57] K. Sabir, C. Stolte, B. Tabor, and S.I. O'Donoghue. The molecular control toolkit: Controlling 3d molecular graphics via gesture and voice. In *Biological Data Visualization (BioVis), 2013 IEEE Symposium on*, pages 49–56, Oct 2013. doi: 10.1109/BioVis.2013.6664346.

[58] Hai-Ning Liang, Cary Williams, Myron Semegen, Wolfgang Stuerzlinger, and Pourang Irani. User-defined surface+motion gestures for 3d manipulation of objects at a distance through a mobile device. In *Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction*, APCHI '12, pages 299–308, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1496-1. doi: 10.1145/2350046.2350098. URL http://doi.acm.org/10.1145/2350046.2350098.

[59] J.L. Gabbard, D. Hix, and J.E. Swan. User-centered design and evaluation of virtual environments. *Computer Graphics and Applications, IEEE*, 19(6):51–59, Nov 1999. ISSN 0272-1716. doi: 10.1109/38.799740.

[60] H. Rex Hartson. Human–computer interaction: Interdisciplinary roots and trends. *Journal of Systems and Software*, 43(2):103 – 118, 1998. ISSN 0164-1212. doi: http://dx.doi.org/10.1016/S0164-1212(98)10026-2. URL http://www.sciencedirect.com/science/article/pii/S0164121298100262.