UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS Honours Project Final Paper 2024

Title: [*A Comparison Between Desktop and Virtual Reality Shape Selection for Volumetric Data with No Priors*]

Author: Willem Van Ryswyck

Project Abbreviation: VRAstro

Supervisor(s): James Gain, Lucia Marchetti and Alex Sivitilli

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	5
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	15
System Development and Implementation	*0*	*20*	10
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

A Comparison Between Desktop and Virtual Reality Shape Selection for Volumetric Data with No Priors

Willem Van Ryswyck
University of Cape Town
Cape Town, South Africa
VRYWIL006@myuct.ac.za

Abstract

Current sub-region selection of H1 astronomical data within a virtual-reality (VR) environment, relies on automated software and manual touch-ups to generate masks over the desired region. As the data does not take on rigid shapes and outlines, the selection process is more complex. This paper explores whether a newly developed VR shape selection method that does not rely on any priors, provides a significant improvement over a desktop selection method. This comparison aims to answer the question of whether selection in VR has benefits over 2D selection on a desktop. The desktop method acts as the control to which the shape selection is compared. Results show that overall, there was no significant benefit to either desktop selection or shape selection. However, post hoc analysis revealed that when only the largest selection object was considered, the was a significant reduction in how time pressured participants felt to complete the task in the shape selection, when compared to the desktop selection. This suggests that further experiments should be conducted with more complex data cubes to see if this improvement applies more generally and reflects across other metrics.

CCS Concepts

• **Hardware** → *Emerging tools and methodologies*; • **General and reference** → **Performance**; • **Human-centered computing** → *Visualization*.

Keywords

Virtual Reality, Methodologies, Volumetric Selection, Visualization

1 INTRODUCTION

With the advent of new telescope technologies, large amounts of astronomical data can be recorded and subsequently analysed for importance, some of which have not been visible before [26]. Due to this, new technologies have been emerging for both visualizing and selecting sections of this data. The traditional way of visualizing data on a desktop is not necessarily the best way to view three dimensional data - as it limits immersion [15]. Studies have shown that using virtual reality to explore data has many benefits[21], such as fewer inaccurate insights and improved satisfaction. Therefore, exploration into the use of VR for volumetric data display is a viable avenue. Volumetric data and its subsequent visualization, extends beyond astronomical data. However, this paper is focusing on the astronomy field, specifically, the selection of sub-volumes of radio astronomical data.

Being able to navigate the raw data does not come without its benefits[21]. However, to get the most insight, sub-selections of the data need to be made and analyzed. When volumetric data is visualized it is represented by a data cube, essentially representing a block of the universe. Within this data cube lies possible regions of interest that need to be singled out. Currently, there exist multiple ways of selecting these sub-regions, however, they are either done from a desktop or rely on automated software, such as SOFIA [28]. The use of automated software is inherently useful in that it can pick up faint sources that are hard to see in the surrounding noise [1, 30], yet they struggle to create perfect masks given that the voxels being selected are rigid in shape but the regions needing to be selected are not, such as in (Fig. 1). This can result in parts of the intended

Figure 1: Data from MeerKAT and the Dark Energy Survey at Cerro Tololo Inter-American Observatory. [13]

sub-region being excluded from the mask (which represents the sub-region that has been selected) [14]. Thus, the astronomical sub-volume selection methods in VR that do exist, have room for improvement as they rely on this imperfect automated software.

Sub-volume selection in VR remains an unexplored area and given the use of VR in exploring astronomy data [14], effective selection could be beneficial. However, if effective VR selection turns out not to be possible, this would be a good indicator that focus should rather be on improving automated and desktop selection methods. Additionally, more precise selection will allow these methods to branch into other fields, such as selection of cardiovascular sections or brain scans.

This leads to the overall question of whether there is a benefit to sub-volume selection in virtual reality compared to selection on the desktop. More specifically, a VR selection method that uses shapes, will be compared to a newly developed desktop selection method that takes inspiration from CARTA [8] by acting on the data cube slice by slice. To answer this question, an experiment was conducted where participants attempted to recreate a mask of specific sub-volume in different data cubes using both selection methods. Various metrics were gathered and compared (i.e. TLX, SUS Score, MCC, F1 Score, and Completion time). TLX (Task Load Index) developed by NASA, measures subjective mental workload through 6 scales (Mental demand, Physical demand, Temporal demand, Performance, Effort, and Frustration) [6]. The SUS (System Usability Scale) Score, is a quick and dirty method to measure the usability of a system through 10 questions [5]. MCC (Matthews correlation coefficient) and F1 Score are two binary classification metrics that are calculated directly from the confusion matrix. The confusion matrix contains the counts of True Positives, False Positives, False Negatives, and True Negatives, to classify a selection "accuracy". In the case of MCC, True Negatives are also considered. These metrics are used to evaluate the performance of a selection compared to a base selection [7]. Finally, completion time recorded how long it took for the participants to recreate the mask to their satisfaction.

The design, development and implementation of the VR Shape and Desktop selection will be discussed in depth as well as the experimental procedure. The selection methods are implemented into a pre-existing astronomical data visualization software called iDaVIE [14], which allows for the easy addition of these methods and subsequent testing and mask generation.

2 BACKGROUND AND RELATED WORK

2.1 Volumetric Selection

A variety of options exist for selection of volumetric data that have yet to be implemented in virtual reality.

2.1.1 Lasso. CloudLasso and TeddyLasso were developed [33] as spatial selection was seen as an avenue to be explored for detecting low density objects, which are hard to select manually [1]. These two lasso methods convert a user defined 2D selection into a 3D selection over the desired volumetric object, where the selection is then refined using a threshold source finding technique [33]. CloudLasso particularly, shows the ability to refine the selection correctly around objects that have noise between desired areas (e.g. doughnut shaped objects) [26]. CloudLasso currently, is only implemented in a desktop environment and does not exist within a VR environment. However, the ability to navigate the 3D space efficiently and see the 3D projection from different angles in VR, may mean implementing CloudLasso in this environment would be effective.

2.1.2 Automated Pipeline. Automated pipeline methods have come a long way from their first implementations seen in the likes of SExtractor [3], which has since become obsolete due to the data produced by modern technology [32]. Following from SExtractor comes one of the first publicly available generic 3D source finders, called DUCHAMP [32]. DUCHAMP uses various preprocesseing and thresholding processing techniques to produce a FITS image that contains a mask over volumetric objects that it manages to detect. It lacks as it only applies a single algorithm for finding objects. This led to the development of SOFIA, which is using multiple source finding algorithms, and thus is applicable to a wider variety of astronomical data sets [28]. While SOFIA still requires some touch-ups, it is a popular selection method that is still used today, especially due to the customization it offers [28]. The biggest competitor to SOFIA is MTObjects [30], which is trained on astronomical data images and uses a Max-Tree-Based method for extraction of identified sources. However, both SOFIA and MTObjects still struggle when the areas they are trying to detect objects within, are highly occluded and have lots of noise [1].

2.1.3 Hybrid. Hybrid selection techniques have been developed that make use of a combination of existing selection methods. The most notable is SlicerAstro, which uses SOFIA [28] for the initial source identification and then uses a modified CloudLasso technique for touch-ups to fine tune the mask [26]. Another hybrid selection method in use is called SlicingVolume, which makes use of an additional piece of hardware – a tablet – which the initial 3D selection is projected onto, and then allows for fine tuning and adjustments [22].

2.2 Selection in VR

As will be seen, Shape selection includes placing 3D objects in a VR environment, making it important to explore the various ways to subsequently select these objects. There exist multiple ways to select objects with defined shapes, with the two categories being hand selection (Fig. 2) and ray casting (Fig. 3). Hand selection relies on being able to reach out and grab a shape, which is possible when the data cube can be navigated [12]. A basic ray cast is made up of a ray the protrudes outward and is aimed towards the desired object. While this method is effective for selecting large shapes, it presents problems when disambiguation of multiple objects is needed to get to the desired object, which is why a depth ray was developed [12]. However, these ray cast methods are not beneficial for selecting clouds as seen in radio astronomy data [31] as there would be too many objects to select [33], but can be beneficial for selecting rigid shapes.

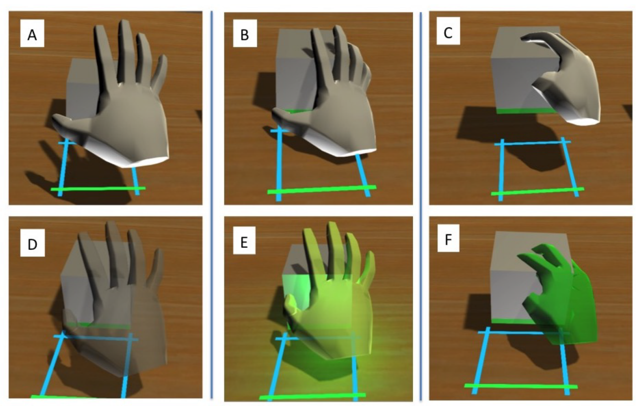

Figure 2: Hand Tracking for Grab-and-Place. [19]

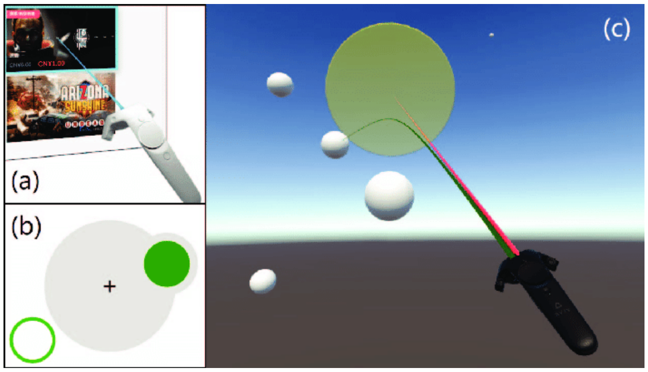

Figure 3: Ray casting as implemented by Lu et al,. [18]

A study investigating what the most beneficial two-hand selection technique is, reveals it is possible to create a cube and select all the volumetric data within it [31], but this does not provide precision. However, it does reveal the benefit of bi-manual symmetric selection, where each hand has the same task and thus allows one hand to rest, providing more overall comfort [31].

Current software implemented by iDaVIE combines the use of the automated selection algorithm SOFIA [28], with manual VR touch ups to individual voxels, to get the desired sub-region mask [14].

3 REQUIREMENT ANALYSIS AND DESIGN

Initial discussions with the team - consisting of two other students, two co-supervisors and the lead iDaVIE developer - revealed multiple ideas for possible virtual reality selection methods. Ultimately, it was decided that the three virtual reality methods to be implemented, would be selection using shapes, using a lasso and using deformation. This paper looks at the implementation of the Shape selection method. Shape selection refers to the idea of using combinations of basic 3D geometric shapes to create a larger more complex shape, which encompasses what needs to be selected. Additionally, it was indicated that there would be a need for a desktop selection method to act as the basis for comparison to the virtual reality methods, and in this paper the Shape selection is compared to it.

The problem and research questions was clearly communicated upon project initiation, which made the process of determining the design requirements simple - “design an effective VR selection method”, and "design an effective desktop selection method." From there, a user-centred design cycle for the Shape selection method was followed of Design, Prototype, Evaluate and then repeat. This approach allowed for regular feedback from those invested in the project (co-supervisors and lead developer), and ensured a working product was ready timelessly. This process, similar to agile development with regular communication, is shown to be more beneficial in reaching the project goal and creating a satisfactory product [20].

The Shape and Desktop selection methods are designed with the end user in mind, as this ensured that the methods were catered to users from the get-go. The potential end users are assumed to have some degree of understanding astronomical data and why they would use these selection techniques. High levels of technical competence however, is not assumed, which results in an end design that is simple to understand and grasp.

3.1 Shape Selection

The broad idea for the Shape selection is to have a palette of 3D semi-transparent shapes that the user can drop into the data cube containing the volumetric data, with Boolean operations defining what happens when these shapes overlap. Additive shapes will add all the voxels within the shape to the mask while subtracting shapes will deselect any mask voxels. Thus, any complex looking shape can be created over the desired object (e.g. a galaxy) and the mask applied. An initial design (Fig. 4) was created using Blender which served as a very basic low-fidelity prototype with no interaction possible. Following feedback, a second design (Fig. **??**) again created using Blender, was presented and the feedback from that was used to begin development and implementation.

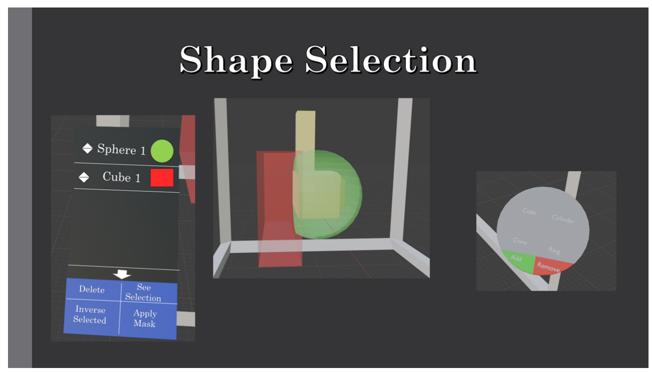

Figure 4: Prototype 1.

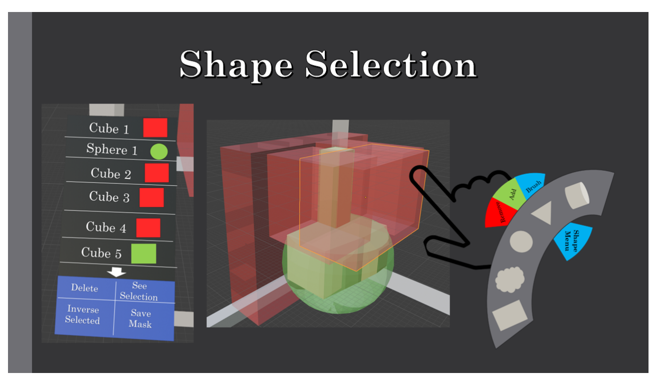

Figure 5: Prototype 2.

Following the initial system development, another feedback session was held with a higher fidelity working prototype. The final changes in design occurred after the pilot experiment, where two participants used the system and were asked to think aloud and suggest any feedback for the Shape selection.

The resulting design is composed of two parts. The first part is a shape palette where the user can spawn a 3D shape, and that shape is either green and additive or red and subtracting (indicating whether it will add to the mask or remove from the mask). The second part is the shape menu, which contains a list of all the shapes that had been spawned in, along with the options to delete a shape, copy a shape, change its additive property, delete all shapes, or apply the mask. Text descriptions are used for all the buttons except the apply mask button which uses a paint brush icon, indicative of the "painting" of the mask voxels. Shapes change colour when selected so that they stand out from the other shapes but still keep a shade of their original colour for recognition over recall.

3.2 Desktop Selection

No formal process was taken for this design method as the only requirement was to design a desktop method that does not require any use of VR. Designs for the method were created during a discussion with the lead developer of iDaVIE. From there frequent updates and feedback was given and received in unstructured weekly updates and communication. During the presentation of the high-fidelity VR selection methods, the desktop method was also included and critiqued. It was also part of the pilot experiment where the final design changes were made.

The design for the method drew inspiration from CARTA[8], a data cube rendering tool. The desktop method was designed to to show the the user a slice of voxels from the data cube, which meant the the whole cube could be viewed by cycling through all these slices. The user can create a mask for each slice so when combined with the other slices, creates a full 3D mask encompassing the whole de sired object. It was communicated the need for shortcuts to actions of navigating slices and creating/removing the mask to shorten the time needed to perform operations. A secondary view of the 3D data cube with in-time updates of the mask as it is applied is included as this allows users to see what selection they have at that moment made.

4 SYSTEM DEVELOPMENT AND IMPLEMENTATION

The project is implemented into the pre-existing iDaVIE software, as this software is already able to visualize data cubes and paint masks [14]. The current version of iDaVIE allows the user to traverse in VR the astronomy data cube using VR controllers and a headset. Users are also able to define a sub-volume selection and create masks over objects by controlling a paint brush with a controller, which adds voxels to the mask as the brush goes over them. The cube visualization and navigation features existing are used by the Shape selection, but the sub volume selection is done using a new shape method, instead of the paint brush. Extending the software meant that the same environment would need to be used for developing the Shape and Desktop selection methods that iDaVIE uses. Thus, both these methods were to be developed using the Unity engine and built on top of the original code. A separate code branch was created for the three separate VR selection methods, to prevent any accidental clashes or reliability issues with the implementation of Shape and Desktop selection.

IDaVIE requires SteamVR to be the engine that controls all the VR interactions so this was used. The methods were used and tested using two VIVE VR controllers and a VIVE headset. While this particular technology is slightly outdated, it is still sufficient. The navigation controls for all the VR selection methods were kept constant with the navigation of the existing software, as this meant that it would be easy for users to remember the controls when it came to experimentation which included the other VR selection methods. Controls differed between Shape and Desktop selection as the desktop used the keypad and mouse instead of controllers.

4.1 Shape Selection

4.1.1 User Interface. The user interface kept consistent with the style already in use in the working software, in terms of menu placement and size. There are two menus, a Shape Menu and a Shape Palette, which were each displayed above a controller and thus follow the user's hands (Fig. 6). This ensures their size stays constant even when the data cube is resized.

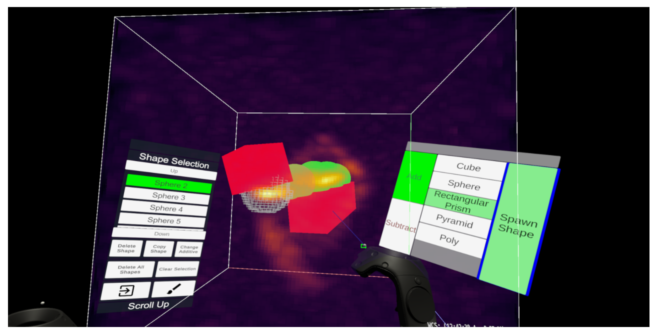

Figure 6: Snapshot of the Shape selection method.

The Shape palette has three columns, each controlling a different feature of the shape. The first column determines if the shape is additive or subtracting and stays highlighted for continuous visual feedback. Selecting "Add" makes the button green and "Subtract" red, which colour is then also used when buttons from the other columns are selected for consistency. The second column has a list of buttons with the type of shape on them (only one can be selected at a time), and the final column is a single button called "Spawn Shape". The palette can be navigated by pointing and clicking on the button with the opposite controller, or it can be navigated using the controller D-Pad where the current button has a light blue outline so the user can see where they are.

The Shape menu is similar in design and appears above the left controller. The menu has a scrolling window showing all the shapes that are in the cube, with the shape that has been selected being highlighted. Beneath that are large easy to read buttons saying what they do to reduce any unnecessary confusion of the functionality, with the exception of the exit button and apply mask button. These are represented by symbols, which are also used across the other VR methods for consistency. The other buttons give the option for the selected shape to be deleted, copied, change state, deselected, or to delete all shapes. There is a bottom container in the menu which displays additional text about each of the buttons functionality.

These buttons are selected by pointing with the opposite controller and clicking on the button.

The important buttons on the Shape Palette ("Spawn Shape", "Add/Subtract") were made large as they are used most and as per Fitts Law [10], large objects are quicker to select. The Shape Palette has the "Add" and "Cube" button selected by default, which follows the principle of error prevention [24], as the user is not able to spawn a shape when a shape has not been specified. Another heuristic followed is Recognition over Recall [24] as either green or red (depending if the shape is either a subtracting or adding shape on the Shape menu, or which of those options is chosen on the Shape Palette), is reflected as a colour across all the buttons, so that the user does not need to remember what option they chose. All buttons on both menus (barring the list of shapes on the Shape menu) are visible, in accordance to the Visibility of System Status principle [24] as well as the Norman's principle of Make things visible [25]. This means that the users are able to see everything at once with nothing hidden, giving them more control.

Being able grab the shapes and move them around with one hand, or select and spawn shapes with the other hand is beneficial as it is shown to reduce fatigue [31], given that one hand can rest while the other is being used.

4.1.2 Code. Shape selection is developed in Unity and makes use of C# scripts, as already done by iDaVIE. Unity was mainly used for creating the user interface and the scripts controlled most of the back-end logic.

Multiple design patterns were taken into consideration when developing the Shape selection method, such as Chain of Responsibility, Observer, Singleton and Mediator [23]. There are two main components for the shape selection, the Shape palette and the Shape menu. Both these components need to interact with each other frequently and a mediator design pattern is used to manage the communications between these two components. This is done with a singleton ShapeManager class, which keeps a list of all the shapes that are created, controls all the manipulations on the shapes, and the creations or removals of the shapes. The Chain of Responsibility behavioural pattern was also used, as the call to add the shapes is received by the ShapePalette class but command is passed onto the ShapeManager to spawn the actual shape. The ShapeManager also acts as an observer, as when it spawns a shape, it lets the ShapeMenu class know that a shape has been added and must update its list of current shapes. The Chain of Responsibility pattern is also used by the ShapeMenu class, as whenever manipulations are done to shapes that have been spawned, the ShapeMenu only determines the request that was made from the user. The actual implementation of the request is done by the ShapeManager.

The ShapeMenu class is responsible for displaying the list of shapes received from the ShapeManager class, and manages their selection. This class passes on the commands to the ShapeManager to delete a shape, change its state, copy the selected shape, delete all shapes in the list, and generate the mask. This class also stores which shape has been selected as certain commands require a shape to be selected.

The ShapePalette class is responsible for receiving information from the user about what shape is being spawned, its properties (either an adding or subtracting shape), and the call to spawn it. It manages what type of shape is spawned, either a sphere, cube, rectangular prism, pyramid, or a poly (which is an asymmetrical polygon). This irregular shape is included as it has unusual sides and angles to it, like what can be found with a galaxy being selected.

Other individual scripts are attached in Unity to shape prefabs that are added into the data cube. These scripts are accessed through the mediator object which is the ShapeManager class. The ClickableShape script is vital for managing shape interactions within the data cube, and ensures that the shape can be selected with a raycast projected from the controller. A more complicated method such as a depth ray [12] was not needed, as the raycast was already implemented by iDaVIE and made sense to extend. The ClickableShape script also controls the colour of the shape and acts as a listener for when the user controller is placed inside the shape which allows it to be moved. The shape can also be scaled when the controller is inside a shape, by clicking up when the up and down buttons on the VR controller. Thus giving control to the user for size, placement and rotation of the shape.

As this method is extending existing software, it is kept robust by following the layout and design of the existing working code. It is modular in design, and only makes reference to existing code when it applies the mask and when it receives input from the listeners of the VR controllers.

4.2 Desktop Selection

4.2.1 User Interface. The user interface is designed to give as much detail on a 2D screen of a 3D data cube. Three cameras are placed in the VR world which shows the data cube from various angles (by looking down the X, Y, or Z axis of the cube), letting the user change how they view the cube. This camera view is displayed on the bottom right of the screen so that there is a view of the entire cube. The main part of the desktop user interface is the representation of a slice of the data cube, which shows a single voxel layer of the data cube. The slice being displayed is indicated by a plane within the 3D cube on the bottom right of the interface, so that users can see where the slice they were looking at is located. This slice of the data cube is where the user creates the mask. By clicking and tracing the outline of the desired selection on the slice (Fig. 7), a temporary selection is created, from where users choose to either reset the selection, or apply a mask to that area. Users also have the option to remove any existing mask from within the selection. If nothing was selected, users can clear the whole selection. The user can also copy the previous slice's mask across to the current slice. Slices are navigable and viewing every slice without changing the camera axis, allows the user to see every voxel of the data cube.

When a mask is applied to a slice, it is indicated by an orange outline on that slice, as well as the highlighting of the voxels inside the 3D data cube displayed on the bottom right. This gives users immediate feedback to the mask they have created/deleted and see what they have done so far. Objects with a hollow center can still be selected by creating a large selection including the whole, and then removing the mask for the hollow area. This can then be viewed by changing the axis, so the 3D view shows the hole created.

4.2.2 Code. Essentially, the desktop method created looks at the full 3D data cube that has already been loaded, and extracts from it

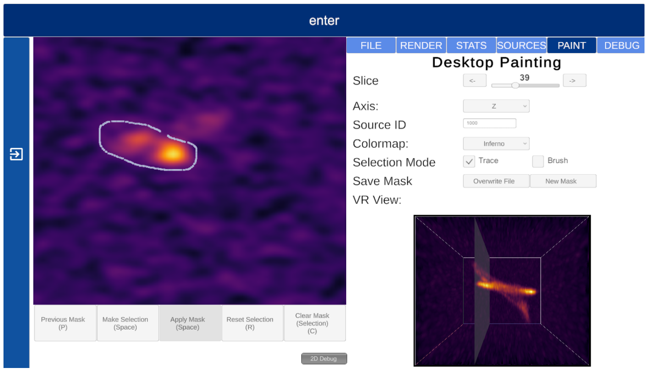

Figure 7: Snapshot of the Desktop selection method with the user outline created.

a single slice. What this means is that a single voxel layer (either looking down the x, y, or z axis) is displayed separately from the whole cube, and on there the user can add or remove the mask, but only for that slice.

The desktop selection does not make use of any VR equipment for the actual selection process. However, to ensure the method stays consistent with the VR methods, it is only accessible once an existing iDaVIE menu is entered while in VR. This is a check done to ensure the datacubes can be edited (create sub-selections). All code for the desktop selection is implemented in a singular class, as there are minimal dependencies.

A slice from the data cube is transferred onto a textured image, allowing the user to have a visual representation of each slice. When the user changes slice, the new slice is calculated dynamically by looking at the FITS data from the whole data cube. When users make a selection on the image of the slice, the user cursor position is localized to a voxel position, allowing the mask to be applied to the correct voxels within the selection.

A secondary FITS file for the visualized data cube is looked at which contains information about what voxels have a mask over them. This file is used to generate new mask selections and deletions, as well as visualize on the slice where a mask already exists. All manipulations were done per slice which constrains changes to only happen to one area at a time.

5 EXPERIMENT DESIGN AND EXECUTION

The experiment was conducted to answer the hypothesis that Shape selection in virtual reality is more beneficial than desktop selection. Along with the desktop selection and shape selection, two other VR selection methods are also included, namely Lasso selection and selection by Deformation. However, these methods are not relevant to test the hypothesis, but must be mentioned as participants also used them as the experiment was a within-subjects study for all methods. The desktop selection was essential as this served as the baseline for comparison to the virtual reality methods. As discussed in section 5.4, the two accuracy scores for the methods are not looked at for their score against the perfect model, but rather to compare it to the score the desktop method received. The same comparison is done for all metrics.

5.1 Participant Selection

Participants were found via convenience sampling with the condition that they were studying a Bachelor of Science degree at the University of Cape Town and were not prone to motion sickness. The motion sickness was checked as VR users can experience symptoms similar to motion sickness while in VR[9], and thus ensuring users are not prone reduces their risk of personal harm. Ideally, the experiment would have been conducted with only astronomy students, but external factors (an Astronomy Conference held during the same week of experiments) meant that majority of participants were Computer Science students. All participants signed a consent form and were given the option to pull out at any time without consequence, with additional warnings given about possible simulation sickness. To give an incentive for participation, a raffle was held to give one random participant a R1000 voucher.

5.2 Experimental Data Sources and Methods

As there were 4 different selection methods done sequentially, participants would have more confidence with basic VR navigability the later on the method appeared. This posed a risk it meant more time could be spent on actual selection for those methods and they would have an advantage. To make the learning effects inconsequential, the order in which the selection methods appeared (either first, second, third, or fourth) was permuted so that an even spread occurred of placement occurred. This ensured that each method appeared in the same spot the same number of times as every other method (or as close as possible). To achieve this, a list was generated which specified for each participant what order they must do the selection methods in, and each participant got a different order to ensure an even spread.

As well as there being 4 different selection methods, there were also 4 different data cubes and subsequent masks shown to each participant (excluding the training cube). These data cubes were intended to have similar complexity, but all had to be different to ensure that each selection method had a unique data cube that the participant had not seen before. Across different participants, the particular data cube used for each selection method was also permuted so that the results would be fair. While the data cubes were similar in complexity and mask shape, the size of the mask did vary with DataCube1 being significantly larger than the others and DataCube2 being smaller by comparison.

5.3 Experimental Outline

The experimental design was kept constant throughout the whole experiment to ensure all participants received the same information and training.

Each participant, upon arrival at the testing venue (a lab within the UCT campus), was given a consent form to read and sign, which gave the general outline of the experiment. A script was followed after this to ensure consistency and following the consent form, participants were given an explanation about what they were to be doing for the rest of the time.

Participants were shown the VIVE VR controllers and told what each button was called. They were then told to put on the headset and a training data cube was loaded onto the system for them to see, while a timer for 5 minutes was started. Whilst looking at this data cube, the controls for moving around, zooming in and out, as well as rotating were explained. They were then able to use whatever time left of the 5 minutes to practice this navigation. During the training, participants were also shown what a mask looks like so that it would not be a foreign concept when it was shown to them for the first selection method.

The next process was repeated 4 times over, once for each selection method (Desktop selection, Shape selection, Lasso selection, selection by Deformation): The participant was given 5 minutes of training (using the same training data cube) on how to use the selection method. Post training, the headset was removed and the actual data cube was loaded into the system along with its mask. The participant was shown what the mask looks like while one of the experiment conductors navigated around the cube to show all angles of looking at the mask. The data cube was loaded back into the system without the mask and the headset was given back to the participant. They were tasked with recreating the mask they had just seen, using the selection method they had just received training on how to use. Participants had 10 minutes (or until they said they were done) to best recreate the mask they were shown prior to beginning the selection. Reiteration of controls was allowed to be given by the experiment conductor but no help on how they should best go about the selection was permitted. Two timers were used to measure how long the participants took to make their selection and reminders were given after 5 minutes, 8 minutes, and 9 minutes. When the participant was done, the time taken was recorded as an average between both times (taken by two different experiment conductors) if they were different. Following the selection the VR headset was removed and the participant was given a questionnaire to fill out while their mask was saved for later analysis. The questionnaire took no more than 5 minutes to complete. No indication to how well the participant did in terms of recreating the mask was given as this would influence the questionnaire responses. This whole process was then repeated for the remaining selection methods.

The design was done as such to ensure participants did not suffer from fatigue, by ensuring the time within the VR world was kept under an hour, and the overall experiment under 1 hour and 30 minutes. Fatigue has been shown to reduce objective performance [2], and thus would take away from the results of the later methods if fatigue was experienced then.

5.4 Data Collected

30 participants completed the experiment but 5 participants had their data excluded (1 participant was an extreme outlier, 2 participants had a method crash during their experiment, 1 participant had their their results overwritten, and for 1 participant their mask was not saved for a method), which left the total at N = 25. Notes were also taken during the experiment about particular actions to seek phenomena.

During each selection method the time taken for participants to complete their mask was taken by two stopwatches (the average was taken when they differed). This was not automated as participants could accidentally leave the method early which would have ended the timer, even though they can go back in and carry on. The completion time helps directly answer the research question, as a more efficient system will not take a significantly longer amount of time compared to the average or what is expected.

After each selection method, participants were asked to complete a survey about that particular method. The first 6 questions were used to conduct the NASA Task Load Index (TLX) survey [6]. The participants were not asked to weigh the 6 questions to each other, meaning Raw TLX was recorded, which decreased the chance of fatigue as they could complete it faster. The TLX score is a good indicator of overall fatigue, and studies have shown that there is a direct link between user satisfaction and reduced cognitive load [27].

Following those 6 questions, 10 more were asked which were used to calculate the System Usability Scale (SUS) score for that method [5]. This score gave a number out of 100 (not indicative of percent) about how user friendly the method was. These questions were answered using a Likert Scale. Determining the perceived usability of a system is vital for determining if Shape selection is more beneficial than Desktop selection, as an unusable system is as specified – unusable.

The masks generated by the participant for each method, were compared to the ideal mask the that was shown before the selection began to. From this comparison, an F1 Score:

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{1}$$

and MCC Score:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2}$$

was calculated. The F1 Score does not take into account the true negatives which is why the MCC score is seen as a more accurate measure [7], but both were recorded. These two accuracy scores give insight into the most important goal which is too create a mask over a specific sub-volume. Thus how well the methods did in accuracy is the strongest indicator towards what is more beneficial.

5.5 Ethical Clearance

Ethical clearance was gathered prior to the experiment starting and ensured no personal binding details would be recorded, bar those needed for the raffle (but these could not be associated with individual results). The clearance gave approval for participants to come from the University of Cape Town and for the experiment outlined to the ethics committee to go ahead.

5.6 Ethics

Respect for persons and informed consent was assured by each participant signing the consent form, which states what the experiment entails and should be expected. Risk was minimized as the only concern was simulator sickness, which the participants were informed about and told they can pull out of the experiment for any reason, at any time, with no consequence. Fairness, in terms of participation selection was not an issue as the raffle was not coercive, and thus participants were not manipulated into taking part. All participation was purely voluntary.

6 RESULTS AND FINDINGS

As mentioned, the experiment was run with participants using 4 selection methods. The data was collected for all methods and was tested for normal distribution together, as the experiments took place together. Two normal distribution tests were run, the Skewness and Kurtois normality test [16], and the Shapiro-Wilks normality test [29]. Both results had a p-value <0.05 for the individual metrics, which indicated the data is not normally distributed. With this constraint, non-parametric equations were run on the complete data set to obtain overall significance, and post hoc analysis to determine where the significance was.

The Friedman rank sum test [11], which was checking for difference between methods for a particular metric, was run for all the metrics. Some of the results from the Friedman test had a p value >0.05, which indicates significance. However, post hoc analysis was performed on those results, using Pairwise comparisons with the Wilcoxon rank sum test with continuity correction (with the Boneferronu adjustment to the p-value). This analysis revealed that specifically between the Desktop and Shape selection, there was no significant difference, meaning it was between other methods that the significant difference was found.

The Kruskal-Wallis H test [17] was also conducted to see if there was significance when individual data cubes were looked at, instead of looking at all the data cubes at once. Initially there is a significant improvement in MCC accuracy for DataCube1, when you compare the Shape method to the Desktop method. However, when the Boneferroni p-value adjustment was made, the p-value was no longer significant. There was no other significance found in the data cubes for any of the metrics using this test.

The final test run was the GLMM test (Generalized Linear Mixed Models) [4] which takes into account the whole data set but looks at two factors (both the selection method and the data cube). During this test all comparisons were insignificant with the exception of one. Temporal demand was significantly higher for the desktop selection method compared to the shape selection method, for DataCube1. "Desktop - Shape Contrast: Estimate = 7.220, SE = 2.49, t-ratio = 2.897, p-value = 0.0244." This has a medium to large effect size of approximately 0.579, indicating substantial effect.

The distribution for each metric also gives interesting insight into the data. The MCC, F1 score and Completion time (Fig. 8) diagrams are displayed below. Both MCC and F1 score averages are similar for Shape and Desktop selection, although particularly with MCC, Shape selection is not as spread out. The completion time is more distributed with Desktop selection performing better (although not significantly as mentioned).

Performance, Effort, and Frustration (Fig. 9), all performed similarly. Although, Shape selection was slightly more frustrating for certain individuals, and Desktop selection required more effort.

Mental, Physical, and Temporal demand (Fig. 10), also had similar distributions. Shape selection on average, had a higher demand for all of the metrics. While Temporal demand seems very similar for Shape and Desktop selection in the diagram, it is comparing all the data cubes, not only DataCube1, which is where there was significant difference.

The final metric, the SUS score (Fig. 11), was slightly better for the Shape selection, which had higher and less distributed values.

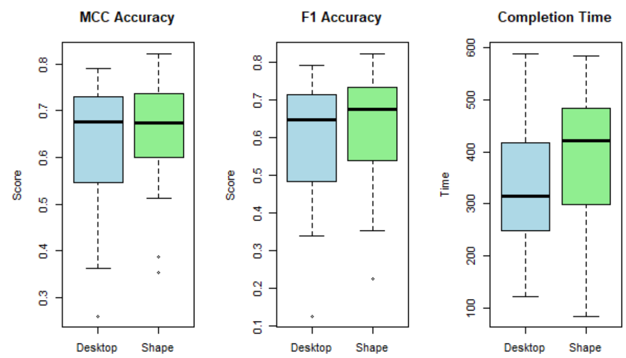

Figure 8: Box and whisker diagram for MCC, F1 Score, and Completion Time.

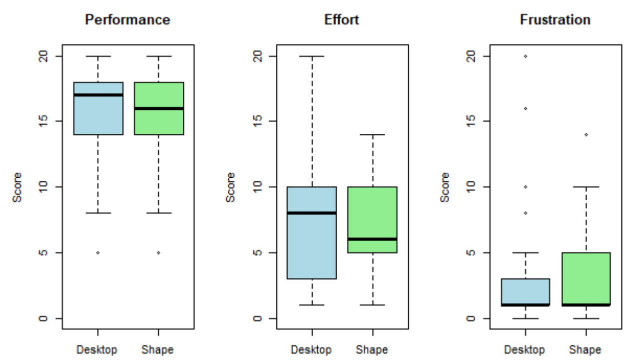

Figure 9: Box and whisker diagram for Performance, Effort, and Frustration.

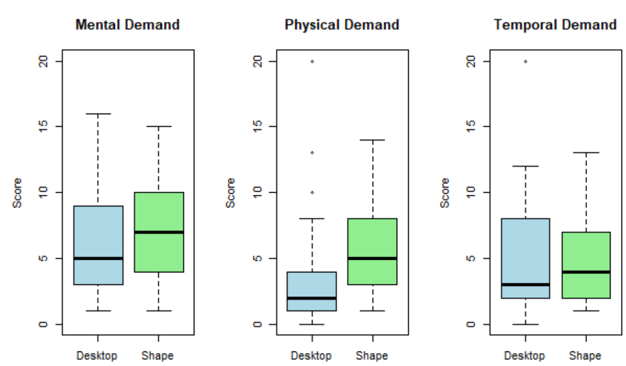

Figure 10: Box and whisker diagram for Temporal, Mental, and Physical Demand.

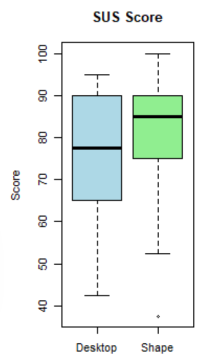

Figure 11: Box and whisker diagram for the SUS score.

7 DISCUSSION

7.1 Results

With the exception of Temporal Demand for DataCube1, there was no significant difference between Desktop and Shape selection for any metrics. This means that the hypothesis cannot accurately be answered as there is no evidence VR Shape selection is substantially better than the other Desktop selection, and vice versa. There are a variety of reasons why the data may have ended this way and will be discussed in the next section.

The significance of the Temporal demand for DataCube1 is as expected. DataCube1 as mentioned, is much larger and takes a lot more slices to get through, which explains why the metric covering the time pressure the participant felt they were under, was the one that was significantly higher for the Desktop selection.

While there is no significance for the other metrics, it is still interesting to analyze the box and whisker diagrams. Specifically interesting to note is that the completion time was quicker for Desktop selection, which given the manual process of going through the cube slice by slice, was not expected. However, this could be due to VR being harder grasp than a traditional desktop environment, which people are used to. Another interesting take from the diagrams, is that the Desktop selection has a lot more outliers than the Shape selection. This indicated that some participants struggled a lot with certain aspects such as high frustration levels and physical demand.

Low accuracy scores overall, could be linked to the fact that the "perfect" mask often included a large area of sub selection that was not actually making up the desired object (i.e. the mask had a large border around the galaxy included), which participants often forgot about. This was especially seen with the desktop method.

7.2 Experiment

There was a lot that was discovered during the experiment which could have been easy fixes, which meant the pilot experiment should have been done with more participants to have found these earlier. It was notable that 5 minutes of navigation was not sufficient for proper training, and while the learning effect was mitigated through permutations, some participants still had not grasped the VR navigation by the final method. This meant focus was taken away from the actual selection.

Except for DataCube1, the data cubes were quite small which is beneficial to the desktop method as it has to go through slice by slice, and thus favors the smaller cubes. It would be interesting seeing what the results were if all the data cubes were large and/or complex. There was also significant complexity differences between the data cubes, when they should have all been the same. DataCube1 had a large ring which did not form part of the mask, and especially with the desktop selection, participants sometimes mistakenly selected parts of it. DataCube4 had a very faint tail which formed part of the selection, which was quite hard to see. DataCube3 was substantially smaller than the other data cubes and was much quicker to select for all the methods.

Hardware caused some problems as the controllers struggled to pick up some inputs on the d-pad and also caused the computer to be slow when trying to generate large masks for certain methods. This resulted in the system crashing twice which meant those two participant's data had to be discarded.

The low number of participants was felt particularly when 5 sets of data had to be scrapped, as the pool became significantly smaller. Additionally, having less people meant that outliers may not be identified as such, and thus contribute more than necessary to the data analysis.

7.3 Future Work

There would be a great benefit in redoing the experiment with a couple of changes. There were no participants from the astronomy department (except one) which meant that often they were confused about what they were selecting, as the explanation given was very high level. Including astronomy students should give more realistic results. Additionally, having more participants N=40 or above, would be a good idea, as it will stop small outliers from skewing the data and if some data has to be scrapped, it is not felt as much.

The metrics should remain the same except additional questions of whether the participant has used VR before should be asked and if they play video games regularly.

Participants should be brought in the day before their experiment for training, to allow there to be substantial time to learn the system and less pressure during the actual experiment.

The data cubes should be more complex, as is more reflective of reality. They should also not be as different from each other.

7.4 Methods

7.4.1 Shape Selection. The Shape selection, in its entirety performed very well and as expected, it did not crash once. There were some minor issues/improvements that were noticed or pointed out. The first was the colour choice. The red (subtracting) shapes were not very visible for DataCube3, which had a red background and made the shape hard to make out.

The "Delete All Shapes" button does not come with a confirm message, even though it is a destructive action. This should be added, as one participant spent 9 minutes adding multiple shapes then accidentally deleted them all before applying the mask.

There were a couple of cases where an additive shape was left floating somewhere in the cube, which still added to the mask when it was applied. However, this is from the user side and there is no easy fix.

In terms of design, the simple feature of adding a cylinder as a shape would have been useful, and was echoed by multiple participants.

7.4.2 Desktop Selection. For the method training, more emphasis should have been put on what changing the axis does. Changing the view can significantly change how quick the slices can be cycled through, and especially if the participant had DataCube1 (the large one), not changing the axis caused a lot more time to be needed to create the mask.

From a design side, the shortcuts worked well, but it was recommended that the wheel on the mouse be used to change the slice. The arrow keys require you to take your hand off the mouse, which slows down the process, especially for larger data cubes.

8 CONCLUSIONS

The paper set out to compare Shape VR selection with Desktop selection and see if Shape selection was a significant improvement. This resulted in two new methods being designed, developed and implemented into the pre-existing iDaVIE software. The experiment conducted to test this hypothesis did not reveal any significant difference, except when looking at Temporal demand for DataCube1. It would be beneficial to do a repeat experiment under stricter conditions and better planning, to see if this finding is consistent, and to see if it extends to other data cube and other metrics. This field of comparison still remains unexplored and this paper shows that it is something that should be looked at closer.

References

[1] Barkai, J. A., Verheijen, M. A. W., Talavera, E., and Wilkinson, M. H. F. 2023. A comparative study of source-finding techniques in HI emission line cubes using SoFiA, MTObjects, and supervised deep learning. *A&A* 670 (2023), A55. https://doi.org/10.1051/0004-6361/202244708

[2] Martin Behrens, Martin Gube, Helmi Chaabene, Olaf Prieske, Alexandre Zénon, Kim-Charline Broscheid, Lutz Schega, Florian Husmann, and Matthias Weippert. 2022. Fatigue and Human Performance: An Updated Framework. *Sports Medicine* 53, 1 (Oct. 2022), 7–31. https://doi.org/10.1007/s40279-022-01748-2

[3] E. Bertin, S. Arnouts, and S. N. Shore. 2009. SExtractor: Software for source extraction. Commentary. *Astron. Astrophys. Suppl. Ser.* 500, 1 (Jan. 2009).

[4] N. E. Breslow and D. G. Clayton. 1993. Approximate Inference in Generalized Linear Mixed Models. *J. Amer. Statist. Assoc.* 88, 421 (1993), 9–25. https://doi.org/10.1080/01621459.1993.10594284 arXiv:https://doi.org/10.1080/01621459.1993.10594284

[5] John Brooke. 1995. SUS: A quick and dirty usability scale. *Usability Eval. Ind.* 189 (11 1995).

[6] Alex Cao, Keshav Chintamani, Abhilash Pandya, and R. Darin Ellis. 2009. NASA TLX: software for assessing subjective mental workload. *Behavior Research Methods* 41, 1 (Feb. 2009), 113–117. https://doi.org/10.3758/BRM.41.1.113

[7] Davide Chicco and Giuseppe Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21, 1 (Jan. 2020), 1–13. https://doi.org/10.1186/S12864-019-6413-7

[8] Angus Comrie, Kuo-Song Wang, Shou-Chieh Hsu, Anthony Moraghan, Pamela Harris, Qi Pang, Adrianna Pińska, Cheng-Chin Chiang, Rob Simmonds, Tien-Hao Chang, Hengtai Jan, and Ming-Yi Lin. 2021. *CARTA: Cube Analysis and Rendering Tool for Astronomy.*

[9] Hyun Taek Kim Eunhee Chang and Byounghyun Yoo. 2020. Virtual Reality Sickness: A Review of Causes and Measurements. *International Journal of Human–Computer Interaction* 36, 17 (2020), 1658–1682. https://doi.org/10.1080/10447318.2020.1778351

[10] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 3 (1954), 387–398.

[11] Milton Friedman. 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Amer. Statist. Assoc.* 32, 200 (Dec. 1937), 675–701. https://doi.org/10.2307/2279372

[12] Tovi Grossman and Ravin Balakrishnan. 2006. The design and evaluation of selection techniques for 3D volumetric displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. Association for Computing Machinery, New York, NY, USA, 3–12. https://doi.org/10.1145/1166253.1166257

[13] Inter-University Institute for Data Intensive Astronomy. 2022. Astronomers Reveal Best Image Yet of Mysterious 'Orcs in Space'. https://idia.ac.za/2022/03/22/astronomers-reveal-best-image-yet-of-mysterious-orcs-in-space/

[14] T.H. Jarrett, A. Comrie, L. Marchetti, A. Sivitilli, S. Macfarlane, F. Vitello, U. Becciani, A.R. Taylor, J.M. van der Hulst, P. Serra, N. Katz, and M.E. Cluver. 2021. Exploring and interrogating astrophysical data in virtual reality. *Astronomy and Computing* 37 (2021), 100502. https://doi.org/10.1016/j.ascom.2021.100502

[15] A. Kaufman, K.H. Hohne, W. Kruger, L. Rosenblum, and P. Schroder. 1994. Research issues in volume visualization. *IEEE Computer Graphics and Applications* 14, 2 (1994), 63–67. https://doi.org/10.1109/38.267473

[16] Hae-Young Kim. 2013. Statistical notes for clinical researchers: Assessing normal distribution (2) using skewness and kurtosis. *Restorative dentistry & endodontics* 38 (02 2013), 52–54. https://doi.org/10.5395/rde.2013.38.1.52

[17] William H. Kruskal and W. Allen Wallis. 1952. Use of Ranks in One-Criterion Variance Analysis. *J. Amer. Statist. Assoc.* 47, 260 (1952), 583–621. https://doi.org/10.1080/01621459.1952.10483441

[18] Yiqin Lu, Chun Yu, and Yuanchun Shi. 2020. Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality. 35–43. https://doi.org/10.1109/VR46266.2020.00021

[19] Alexander Masurovsky, Paul Chojecki, Detlef Runde, Mustafa Lafci, David Przewozny, and Michael Gaebler. 2020. Controller-Free Hand Tracking for Grab-and-Place Tasks in Immersive Virtual Reality: Design Elements and Their Empirical Study. *Multimodal Technologies and Interaction* 4, 4 (2020). https://doi.org/10.3390/mti4040091

[20] Soly Mathew Biju. 2010. Agile Software Development Methods and its Advantages. (05 2010). https://doi.org/10.1007/978-90-481-9151-2_105

[21] Patrick Millais, Simon L. Jones, and Ryan Kelly. 2018. Exploring Data in Virtual Reality: Comparisons with 2D Data Visualizations. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI EA '18)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3170427.3188537

[22] Roberto A. Montano-Murillo, Cuong Nguyen, Rubaiat Habib Kazi, Sriram Subramanian, Stephen DiVerdi, and Diego Martinez-Plasencia. 2020. Slicing-Volume: Hybrid 3D/2D Multi-target Selection Technique for Dense Virtual Environments. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 53–62. https://doi.org/10.1109/VR46266.2020.00023

[23] Huaxin Mu and Shuai Jiang. 2011. Design patterns in software development. In *2011 IEEE 2nd International Conference on Software Engineering and Service Science*. 322–325. https://doi.org/10.1109/ICSESS.2011.5982228

[24] Jakob Nielsen. 1994. Enhancing the explanatory power of usability heuristics. In *Proceedings of the ACM CHI '94 Conference on Human Factors in Computing Systems* (Boston, MA, USA). ACM Press, New York, NY, USA, 152–158. https://doi.org/10.1145/191666.191729

[25] Donald A. Norman. 1988. The Design of Everyday Things. https://www.freemindworld.com/works/Design.pdf

[26] D. Punzo, J.M. van der Hulst, J.B.T.M. Roerdink, J.C. Fillion-Robin, and L. Yu. 2017. SlicerAstro: A 3-D interactive visual analytics tool for HI data. *Astronomy and Computing* 19 (2017), 45–59. https://doi.org/10.1016/j.ascom.2017.03.004

[27] Laurisha Rampersad, Sarah Blyth, Ed Elson, and Michelle M. Kuttel. 2017. Improving the usability of scientific software with participatory design: a new interface design for radio astronomy visualisation software. In *Proceedings of the South African Institute of Computer Scientists and Information Technologists* (Thaba 'Nchu, South Africa) *(SAICSIT '17)*. Association for Computing Machinery, New York, NY, USA, Article 29, 9 pages. https://doi.org/10.1145/3129416.3129899

[28] Paolo Serra, Tobias Westmeier, Nadine Giese, Russell Jurek, Lars Flöer, Attila Popping, Benjamin Winkel, Thijs van der Hulst, Martin Meyer, Bärbel S. Koribalski, Lister Staveley-Smith, and Hélène Courtois. 2015. SoFiA: a flexible source finder for 3D spectral line data. *Monthly Notices of the Royal Astronomical Society* 448, 2 (02 2015), 1922–1929. https://doi.org/10.1093/mnras/stv079 arXiv:https://academic.oup.com/mnras/article-pdf/448/2/1922/3622360/stv079.pdf

[29] S. S. SHAPIRO and M. B. WILK. 1965. An analysis of variance test for normality (complete samples)†. *Biometrika* 52, 3-4 (12 1965), 591–611. https://doi.org/10.1093/biomet/52.3-4.591 arXiv:https://academic.oup.com/biomet/article-pdf/52/3-4/591/962907/52-3-4-591.pdf

[30] Paul Teeninga, Ugo Moschini, Scott C. Trager, and Michael H. F. Wilkinson. 2015. Improved Detection of Faint Extended Astronomical Objects Through Statistical

Attribute Filtering. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, Jón Atli Benediktsson, Jocelyn Chanussot, Laurent Najman, and Hugues Talbot (Eds.). Springer International Publishing, Cham, 157–168.

[31] Amy Ulinski, Catherine Zanbaka, Zachary Wartell, Paula Goolkasian, and Larry F. Hodges. 2007. Two Handed Selection Techniques for Volumetric Data. In *2007 IEEE Symposium on 3D User Interfaces*. https://doi.org/10.1109/3DUI.2007.340782

[32] Matthew T. Whiting. 2012. duchamp: a 3D source finder for spectral-line data. *Monthly Notices of the Royal Astronomical Society* 421, 4 (04 2012), 3242–3256. https://doi.org/10.1111/j.1365-2966.2012.20548.x arXiv:https://academic.oup.com/mnras/article-pdf/421/4/3242/3817619/mnras0421-3242.pdf

[33] Lingyun Yu, Konstantinos Efstathiou, Petra Isenberg, and Tobias Isenberg. 2012. Efficient Structure-Aware Selection Techniques for 3D Point Cloud Visualizations with 2DOF Input. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2245–2254. https://doi.org/10.1109/TVCG.2012.217

A Data Cubes

Data cubes used in the experiment.

A.1 DataCube1

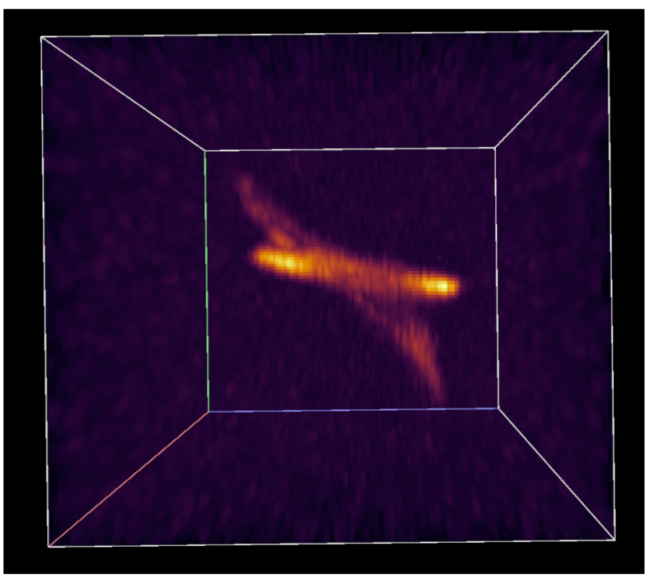

Figure 12: DataCubeOne. Just the central "potato" object was to be selected.

A.2 DataCube2

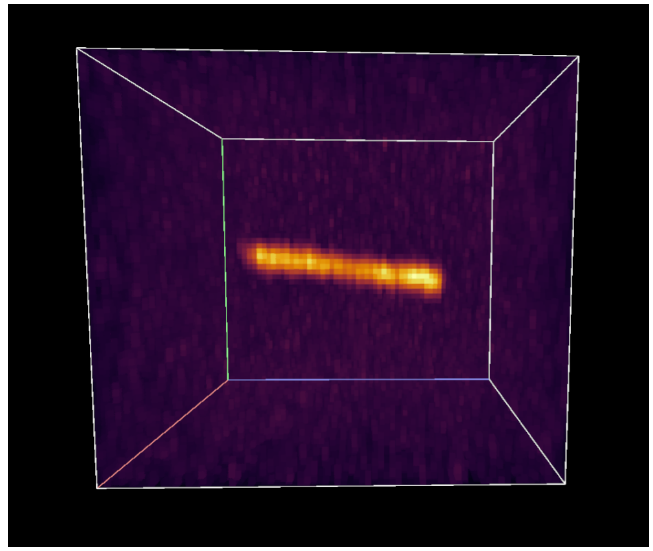

Figure 13: DataCube2. Small data cube with a red background.

A.3 DataCube3

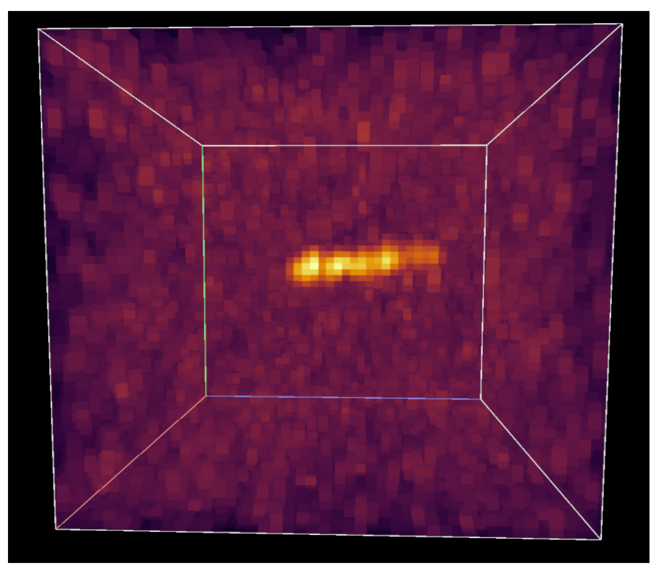

Figure 14: DatCube3

A.4 DataCube4

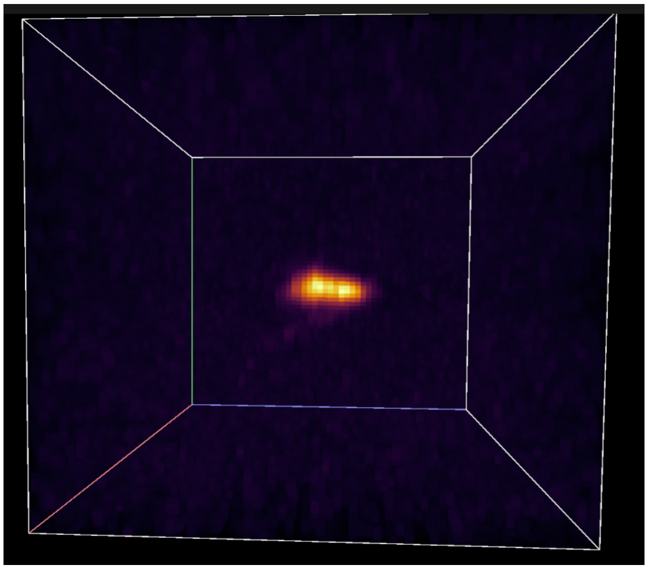

Figure 15: DataCube4. Small tail was included in the desired mask.

A.5 TrainingCube

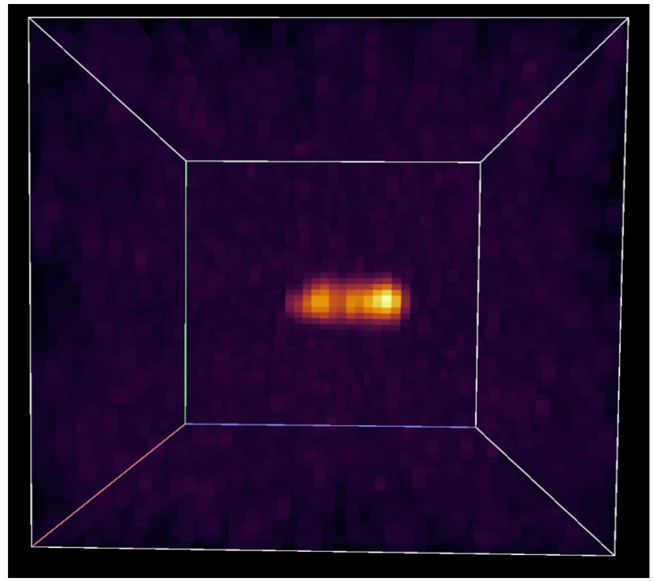

Figure 16: Training Cube used to learn the navigation and methods.