DEPARTMENT OF COMPUTER SCIENCE

CS Honours Project Final Paper 2024

Title: [*Sub-Volume Lasso Selection with No Priors in Virtual Reality*]

Author: Jesse Plant

Project Abbreviation: VR Astro

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**	

Sub-Volume Lasso Selection with No Priors in Virtual Reality

Jesse Plant
Computer Science
University of Cape Town
Cape Town, Western Cape, South Africa.
plnjes001@myuct.ac.za

ABSTRACT

Astronomy is faced with the pressing issue of how to best analyze astrophysical data given ever increasing data sizes. The issue is perpetuated by astrophysical datasets consisting of three dimensions. One solution is to reduce the data to subsets and analyze the newly divided subsets in greater detail in isolation. This requires the usage of a selection tool, which raises the question as to which selection tool environment would prove most advantageous in terms of accuracy, completion time and usability. This paper presents one method for selecting subsets of volumetric data in virtual reality (VR), the lasso method. A desktop method was also developed as a benchmark for comparison. Selection methods are commonly compared using accuracy, efficiency, and usability metrics [4, 22, 23, 24]. A formal user test was conducted to statistically determine the performance of the selection methods. As volumetric datasets comprise of three dimensions, it may prove beneficial to interact and explore the data within a 3D environment. The lasso method developed to test this theory allows a user to select sub regions through two semi-transparent planes and a user drawn shape on the close plane (The planes are shown in figure 1, as the blue squares). The user must have the region of interest placed between the two planes by positioning and orientating the planes. The region can be selected by drawing a shape around the desired volume. The lasso and desktop method are implemented within the iDaVIE-V system developed by Marchetti et al. [15] as their software facilitates the visualization of and interaction with volumetric data. The objective of this paper is to determine if selections made in VR are more accurate and easier to use compared to more traditional methods like the developed desktop. The user experiments found that the lasso method performed significantly worse than the desktop in accuracy, time taken and usability.

KEYWORDS

Lasso Selection, VR, iDaVIE-V, Volumetric selection, Volumetric Data.

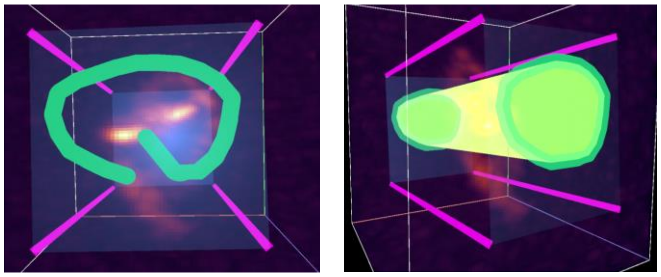

Figure 1: A demonstration of the developed lasso method.

1 INTRODUCTION

The scientific community continually advances through new technologies and ideas; these advances result in improved methodologies for retrieving and analyzing data. This ever-increasing quality and quantity of data often births a bottleneck [16] in data analysis across numerous scientific fields. Hence, there is always a need for improved sub-selection and analysis techniques for data [3]. This need is further compounded in the context of volumetric data (data with three dimensions) as analysis is typically limited to a 2D environment. Fields such as medicine and astronomy make extensive use of volumetric data as medical scans and astrophysical data are stored in 3D datasets. Astronomers commonly store this data in data cubes (Three dimensional datasets) comprised of two spatial dimensions and one spectral [12]. Astronomers are increasingly approaching the analysis of 3D datasets within a 3D environment, mainly through virtual reality. VR enables users to immerse themselves within the data, allowing easier data navigation and analysis. Software such as iDaVIE-V developed by the astronomy department at the University of Cape Town offers astronomers a 3D environment to visualize and interact with volumetric datasets. iDaVIE-V served as a framework to conduct the user experiments and implement the selection methods in. Astronomers study regions of interest in isolation to reduce noise from the remaining dataset. These regions are often unconventionally shaped, encouraging user input to define selections. The lasso selection method enables users to create custom 2D shapes that are drawn onto a plane acting as a whiteboard. The depth of the 2D shape is controlled by the distance of a second plane from the first. Once the user is satisfied a mesh is generated to encompass the lasso volume. An inside-outside test is then performed over the lasso mesh to identify points within for selection. By contrast, the desktop method follows Slicer-Astro [6] closely and goes through the data on a slice-by-slice basis. On each slice the user draws a shape around regions of interest. The core functionality of lasso-based selection allows for accurate and intricate sub-volume selection to better deal with irregularly shaped volumes of data. The implementation of a lasso selection method is based on CloudLasso developed by Yu et al. [23] from which the key features of a lasso were derived. As CloudLasso is a desktop development, the developed method also drew on the Tangible Brush method developed by Besançon et al. [24] for the user input and VR based implementations. Their work will be further discussed in section 2.

This project's purpose was to explore the effectiveness of selection methods in VR with reference to a desktop method. To achieve this, formal user experiments were conducted, involving two other VR based selection methods. The study aimed to answer the research question; *Does volumetric selection through a lasso method in virtual reality provide an improvement (increased efficiency, accuracy, usability and decreased workload) in data selection compared to a desktop environment.* The following hypotheses were formed to assess the study results regarding the research question.

Hypotheses:

- Null Hypothesis (H0): There is no difference between desktop and VR environments for volumetric selections in terms of (i) accuracy, (ii) completion time, and (iii) usability.
- Alternative Hypothesis (H1): There is a significant difference between desktop and VR environments for volumetric selections in terms of (i) accuracy, (ii) completion time, and (iii) usability.

If H0 is rejected, the following will also be checked.

- Null Hypothesis (H2): The significant differences are a result of VR environments improving volumetric selection.
- Alternative Hypothesis (H3): The significant differences are a result of VR environments hindering volumetric selection.

Efficiency was evaluated through completion time. Accuracy was measured through the F1 Score and Matthew's Correlation Coefficient (MCC). Usability was measured with the NASA Task the System Usability Scale (SUS). The workload is measured by the NASA Task Load Index (TLX). All measurement metrics are further elaborated on in this paper. Furthermore, this paper goes into the development of both the lasso and desktop methods, followed by the specifics of the user testing. Lastly, it analyzes and reports the study results and discusses the issues experienced.

2 BACKGROUND AND RELATED WORK

Lasso Selection enables a user to make a selection based on their 2D input, which most commonly is in the form of a drawing. Making selections in a 2D environment reduces the complexity of the selection to an area selection. For a volumetric selection in a 3D environment, one must provide a 2D surface to register the user input. Ideally the user will also need to control the distance into the 3D environment that the lasso selects. The user's drawing is then mapped into the 3D environment. Lasso selection ties itself closely with volumetric selection as the shape of the lasso defines a volume to be selected, similarly to how a circle defines a cylinder. John Lucas et al. [14] made use of a tablet-based lasso selection called Tablet Freehand Lasso Technique, also known as Cylindrical Selection. With Cylindrical Selection, users view a section of a 3D dataset through a traditional 2D view, such as a drawing tablet. By utilizing the tablet and its accompanying pen, the user can draw 2D input from a specific point and orientation. This 2D input is then projected into the 3D space. The selection volume is conical in nature with it stemming from the user input. The volume bound by the lasso or the objects within the volume are then selected. The lasso used in cylindrical selection does not account for the spatial structure of volume or objects being selected, instead it concerns itself purely with what resides within the selection volume.

In contrast to Cylindrical Selection, researchers Lingyun Yu and Konstantinos Efstathiou [23] developed a pair of lasso selection methods Cloud Lasso and Teddy Selection, designed to factor in the spatial structure of selection targets. To substantiate their work, they conducted a user experiment with the aim of directly comparing the new methods to the Cylindrical Selection method. Both new methods incorporated the core principle of lasso selection to select a region and used some form of density calculations to estimate the point of interest's spatial structure. Teddy Selection works by compartmentalizing the lasso region into "bins". These bins are then filtered by a chosen density threshold and only those bins that meet the threshold continue to operate. [23]. The shape of the objects are estimated through the triangulation of the remaining bins. Teddy Selection is rather rigid in its functionality as the program cannot change the density threshold, nor the length of the selection region into the 3D space. The lack of customization often results in additional unwanted selections.

Cloud Lasso overcomes some of Teddy Selection's shortcomings and is less computationally expensive than Teddy Selection in terms of the density calculations as it can be parallelized. Cloud Lasso improved on Teddy Selection through additional functionality that allowed the user to control both the density threshold as well as the depth of the lasso. Cloud Lasso makes use of an algorithm called "The Marching Cubes" [23] and it works by constructing a minimal cuboid to bound the lasso region or object for selection. The minimal cuboid is decomposed into smaller cuboids, with each small cuboid being subjected to a density calculation. Similarly to Teddy Selection, Cloud Lasso disregards the cubes that do not meet the density threshold set by the user. The estimated spatial structure of selections with Cloud Lasso were better than Teddy Selection.

Due to the improved design and implementation of Cloud Lasso, Yu and Efstathiou decided to only test Cloud Lasso against Cylindrical Selection. This study concluded that Cloud Lasso was significantly more accurate than the cylindrical counterpart according to F1 and MCC [23] calculations. However, the increased accuracy came at the expense of increased required computational power. Both Teddy Selection and Cloud Lasso inherently struggle with sparse datasets due to their reliance on density calculations. This limitation renders these methods obsolete for the purpose of volumetric selection.

Yu and Efstathiou continued to improve their selection methods in collaboration with Petra and Tobias Isenberg [22]. to produce SpaceCast. SpaceCast aimed to improve Cloud Lasso by factoring in the shape of the user's input to only select objects or volumes like the input shape [22]. To accomplish this, the method compared the shape of the selection in terms of a 2D plane and the user input shape. In a study comparing SpaceCast to Cloud Lasso and Cylindrical Selection it was found that SpaceCast was significantly faster in terms of completion time. There were no substantial differences in accuracies.

While Cylindrical Selection, Cloud Lasso and SpaceCast have all proved the merit and strengths in lasso selections they are all limited by the requirement of drawing the lasso in the virtual environment without a sense of depth. The Tangible Brush method developed by Lonni Besançon et al. [24] made use of a hybrid environment to achieve lasso selection. This hybrid environment used a tablet to record user input. The Tangible Brush method differs from the previously mentioned methods by requiring users to navigate the 3D environment by physically manipulating the point of view when drawing input. Tangible Brush based its selections off of multiple lassos from different points of reference instead of one lasso like the other methods. Once again, a study was conducted for comparisons, and it was found that Tangible Brush was slower but more accurate than SpaceCast. The insignificant accuracy improvements were outweighed by the decrease in completion time. The study also concluded that Tangible Brush had a higher mental workload than SpaceCast. In all other TLX factors, there was no significant difference. Tangible Brush is best for the use case of complicated datasets and selections due to the increased accuracy of multiple lassos in conjunction with multiple angles of observation on the data.

Currently, iDaVIE-V has two approaches to volumetric selection. Manual selection requires a user to be involved with the selection such as the lasso and desktop method. Automatic selection uses software to make selections. iDaVIE-V currently uses SoFiA developed by P. Serra et al [19]. It makes selections in astrophysical data by source detection. Although it is an automatic selection tool, SoEiA often requires an Astronomer to perfect the mask. Currently, a paintbrush is used to edit these selections. There are many possible methods of manual selection. A. Ulinski et al. [4] suggested using 3D shapes to define selections. This lends itself to VR as it offers increased ease to move and manipulate the shapes. This idea can be expanded to mold shapes into desirable volumes for selection. Using direct manipulation models developed by J. Gain [10] could achieve this.

3 LASSO METHOD DESIGN AND IMPLEMENTATION

The development of this project followed an iterative approach, incorporating user-centered design [1], including demonstrations of the design and implementation to the supervisors of the project. On average, the meetings took place every second week either in person or online depending on the supervisors' schedules. After acquiring their feedback, changes were implemented to improve the design and usability. Before user experiments began, pilot testing was conducted with two individuals to focus on a user-centered design.

The lasso system explained below serves as a means of volumetric selection within the iDaVIE-V system. This environment was developed with the Unity Engine. Unity utilizes C#, a scripting language. It follows that the lasso method was developed with Unity in C#. For VR, a Hive HTC Pro headset and accompanying controllers were used. IDaVIE-V handles all VR inputs through steam VR.

3.1 System Requirements

The requirements for a lasso system are listed below:

- The depth of the lasso "into the screen" must be adjustable. This prevents excess selections of undesired voxels. This requirement was identified by the short comings of Teddy Selection compared to Cloud Lasso.
- The lasso system should be able to be moved with the data cube, to increase the intuitiveness.
- The lasso system must emphasize performance to not induce motion sickness within users.
- The lasso system needs to record and reset user input.
- The lasso system should be able to remove sections of input where the line loops over itself.
- The lasso system should allow for selections to be made with one or more iterations of it.
- The lasso system should have a conical option, where the length of the lasso scales up the end of the lasso.

3.2 Integration in iDaVIE-V

VR controllers were used for navigation and interaction in the iDaVIE-V system. These same controllers were used to make selections with the lasso system. Menu interactions positioned the system. The user was able control the length of the lasso with one hand and draw their input with the other, thus defining the lasso. To change the length of the lasso, the user held down a button and moved their hand up or down to shorten or lengthen it. Drawing required the user to point their right hand at the system and hold down a button to record their input. The final deliverable prevented users from positioning and orientating the lasso system with the data cube. This was due to an issue causing the mesh to offset from the lasso. Instead, the user had to position the lasso with any alterations requiring the repositioning of the data cube.

3.3 Lasso System

Due to a lack of experience with VR development, a desktop prototype of the lasso was developed first. The desktop version served to identify the Unity components and game objects needed to build the lasso.

The base of the lasso system included two planes, a close and a far plane. On clicking the play button in figure 4, the close plane spawned in front of the camera following and facing the headset. The user then locked the closer plane's position and orientation which spawned the second plane. The second plane was always in line with the first plane - the planes were in parallel. At any point the user could reposition the plane by unlocking and relocking its position. With the first plane locked in position, users could control the lasso's length by moving the second plane towards or away from the first.

Before being able to draw a lasso, the user needed to lock the close plane. The second plane's visibility enforced an understanding of the lasso depth and therefore, the range of the selection. The need for length control arose from the strengths of CloudLasso as well as the shortcomings of Teddy Selection. The benefits of positioning and orientation stem from the Hybrid Tangible Brush system as it strongly lends itself to a VR environment because users can navigate themselves within the data. The ability to orientate and position the lasso system was a necessity. The Tangible Brush made use of a hybrid system with a physical pen and pad which cannot translate into a VR environment. It follows that the lasso system approached the positioning, orientation and user input from a VR standpoint.

Line renderers are a component in Unity that create lines by adding co-ordinates to an array and joining a line through them. A line renderer allows for a user's input to be recorded. This was achieved by adding a line renderer component to each plane. Ray casting was used to record input. For input to be recorded, the system had to be locked before holding down a specified button while pointing the controller to the plane. The input was recorded such that the ray cast position had to be a greater distance from the most recently input position than some minimum threshold. This was to prevent excessive positions being added to the line renderer. The line renderers' arrays were used in constructing the mesh, limiting the number of values improved performance and produced smoother meshes. Once the user was satisfied with their input, they clicked the tick button on the menu which generated the mesh. This button is discussed further later in the paper. The mesh and or user input could be reset at any time with the reset button. Before the mesh was created, the input line renderer's values were copied to the second plane's line renderer. With the two arrays of positions, the mesh was constructed. In Unity meshes are constructed using a triangle index array of size six. This mesh was constructed between the two sets of vertices from their respective line renderers. Two triangles made up a quad with which the mesh was constructed. It looped through all the vertices and on each vertex, it filled out the triangle array. The first triangle (first three values) took in current vertex from the first array (top left of the quad), then the next vertex (top right) and lastly the second array's current vertex (bottom left). The second triangle first took in the top right vertex, then the bottom right vertex and lastly the bottom left vertex. After this the vertices and triangles were then set to the mesh, where Unity handled the rest.

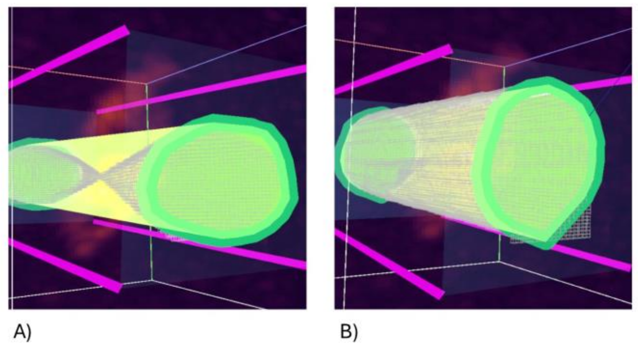

Figure 2: A): Shows the cone bug. B): Shows a correct selection.

Being able to move the lasso system with the data cube would have allowed users to easily gain a sense of depth and confirm their selection from multiple perspectives. This would have greatly enhanced the lasso's usability. Despite the intuitive benefits of moving the lasso system with the data cube, it was not implemented. This was the result of a bug where the lasso mesh would detach from the system. Due to time constraints this bug was worked around by preventing the user from moving the lasso with the environment. Instead, users needed to rely on physically relocating themselves in the real world. Depth cues were added to reduce the effect of this limitation. Bright pink lines connected the corners of the two planes to indicate the depth. Lastly, the user could add the voxels in the lasso by clicking the paint brush. Alternatively, users could remove previously selected voxels by toggling the additive or subtractive mode. Through this, users can iteratively refine their selections, an example is shown in figure 3. However, a bug was discovered during the experiment that came to be known as the cone bug. This bug caused the voxels to converge at the center of the lasso in a cone like manner and then disperse in a similar manner towards the end of the lasso, creating an hourglass shape. Figure 2 shows an example of the cone bug compared to a correct selection.

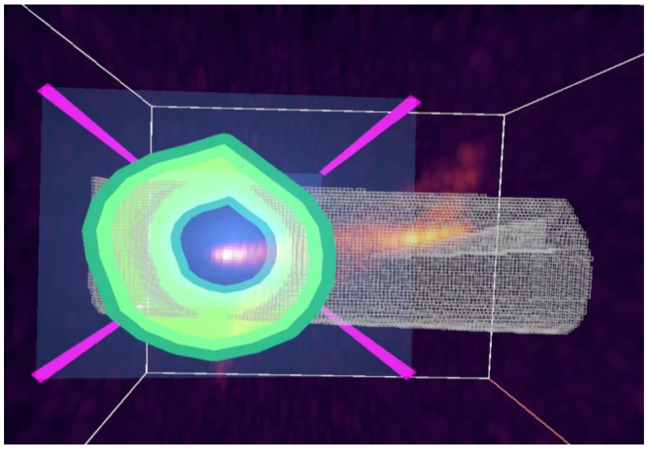
Figure 3: A view of the subtractive function being performed on a selection.

3.4 Lasso menu and controls.

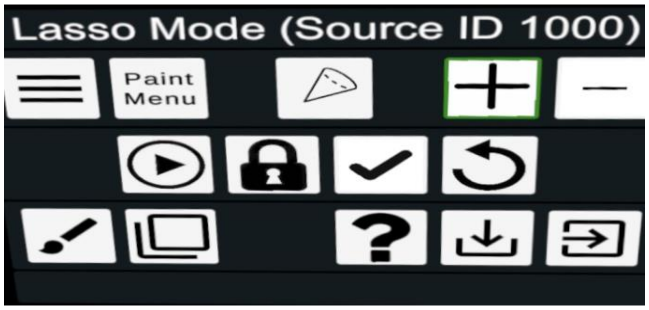

Figure 4: A view of the menu screen used to control the lasso system.

This section uses figure 2 above to explain the controls of the system. Firstly, the user should have spawned in the close plane (the middle row, left button). After this the user orientated the system and locked it position with the lock icon (middle row, middle left). With the system locked users can then draw on the board similarly to figure 1. Once the user was content with their input, they clicked the tick button (middle row, middle right) to close their input and generate the lasso. If at any point they wanted to reset the lasso or their input, the user could click the reset button (middle row, right) and begin drawing. To add the volume within the lasso, the user could click the paint brush button (bottom row, left) to add the selection to the mask. The additive and subtractive feature was toggled by the top right buttons, with the green or red outline indicating the current setting.

3.5 User Input Handling

To prevent undesired inputs after the user had clicked the tick button, the array of positions in the line renderer were altered. Firstly, this code closed the shape by appending the first value in the array to the last. Afterwards, using the minimum distance between points threshold, it checked for any two points in closer proximity than allowed. The array was then altered to remove all points making up the loop, including the second identified point. This method prevented the input from intersecting itself. However, it is noted that this method was not always accurate. This resulted in some undesired inputs, requiring the user to reset their input and redraw it. These cases were few, but often occurred when a user closed the shape instead of allowing the software to handle it.

3.6 Inside-Outside Test

Although the lasso mesh encompassed a volume for selection, further action was needed for selection. Some terminology is defined before discussing how the mesh results in selections. IDaVIE-V uses data cubes to display data, with these cubes comprising of voxels. Voxels are simply 3D pixels and in this case each voxel stores a piece of data about the cube. The lasso mesh needed to identify which voxels were within it and those that were not. To test every voxel was inefficient. Therefore, a bounding box was created to fully encompass the mesh. Only voxels in the box were tested, reducing the check to voxels close to the mesh. However, difficulties with the rotation of the bounding box left it larger than necessary contrasting the minimum size to contain requirement [21]. Voxels were tested by ray casting from their position to the center of the lasso mesh. This required any alterations in the lasso to update this center. If there was a collision with the mesh, the normal vector of the collision point was dotted with the ray's direction. A positive result implied that the voxel was inside the lasso, otherwise it lay outside of the mesh. This worked due to the lasso mesh being enclosed by the planes. Each treatment in the study used this method.

3.7 Desktop Selection

In addition to developing the lasso method, a desktop method was developed. The desktop method enabled the study to answer the research question regarding whether VR offered improved performance in volumetric selection over desktop. The desktop method drew inspiration from Slicer-Astro by navigating the data through slices. This involved separating the data cube into planes where the third dimension of the data was navigated by changing slices. On each slice the user selected regions of interest by drawing an outline of them with their mouse. The user was also able to copy the mask on the previous slice and put it in on the next viewed slice. The user could also edit these selections by adding or removing parts of the mask. Figure 5 Shows the desktop UI. Multiple features were developed for the astronomy co-supervisors; however, they were not involved with the experiments.

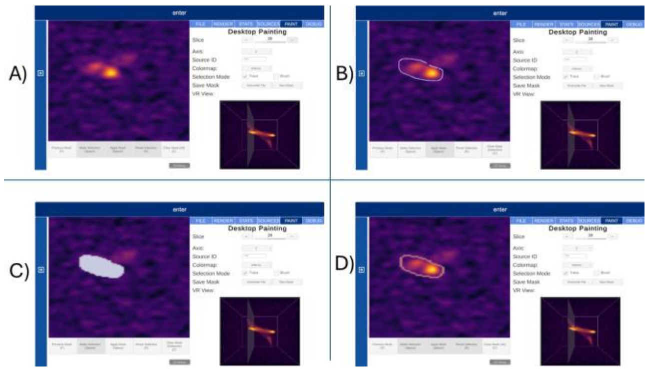

Figure 5: A) Shows the desktop method idly viewing a data cube. B) Shows the desktop method beginning a selection. C) Shows the intermediate step before applying the mask. D) Shows an applied mask.

4 TEST METHODOLOGY

4.1 Overview of user testing

Although this paper focuses on two selection methods, the test was designed and conducted with four selection methods. Each user test comprised of the four selection methods where each selection method test was performed on one of four data cubes, as seen in figure 6. In each test, the user had to perform a sub-volume selection of a desired region in the designated data cube with the specified method.

Structure of user testing. As previously mentioned, there were four data cubes each with accompanying exemplar masks. Each participant performed one selection with each method to be in line with tests within subjects. Each of these selections were on one of the four data cubes. The order of the selection methods and data cubes were permutated to mitigate the learning effect on participants [13]. The data cubes were of varying levels of difficulty, with two being relatively easy and the other two more challenging. Our experiment aimed for 40 user tests under the guidance of our supervisors. Each experiment was expected to take 60 to 90 minutes. For each method a user had five minutes training and ten minutes to perform a selection. Before a user made a selection, they were shown a demonstrative mask which they had to replicate. To enforce consistency across user testing, each method had a script for training.

Table 1: Testing Schedule:

Task	Time (minutes)	Total(minutes)
Introduction	5	10
Tutorial	5	
Training 1 *	5	17 (x4)
Test 1 *	10	
Questionnaire 1 *	2	
Completion	•	78

Those with an * or highlighted blue were repeated 4 times, once per treatment.

Pilot testing. Regrettably it was noticed during the user experiments that the pilot test did not take place far enough in advance to properly correct or adjust the code. Nor was the extent to which we recreated the actual user experiment or number of pilot tests enough to adequately prepare for user testing. This was due to development delays resulting in cutting the schedule short as well as a lack of experience with a project of this scale. Due to our two user testers, some bugs were identified and corrected. It was advised that the lasso remain on a 1:1 scale instead of taking on a conical shape for the scale of selection. This suggestion was incorporated in the final version. The project would have benefited greatly from proper and sufficient user testing. This is evident within the UI design of the method and overly complicated menu system.

Ethics, informed consent and participants. This project was conducted in accordance with the requirements of the University of Cape Town's Ethics committee in the science faculty. Experiments were only run once, acquiring ethics approval. Furthermore, all participants were acquired with DSA approval for access to the student body. Simulation sickness [8] is defined as *a syndrome similar to motion sickness and can be experienced as a side effect during or after exposure to different virtual reality environments.* Hence, the participants needed to sign an informed consent form, which explicitly states participants can withdraw at any point without any consequences. The form explained that the participants would remain anonymous, as well as their data. Any records referred to participants by the number of their experiment. The target population of the study were astronomy students, and students in the science faculty. All but one of our users were outside the astronomy department due to external factors preventing their participation. The participants were compensated by entry to a raffle for a R1000 Takealot voucher.

4.2 Evaluation

Before conducting a scientific test, it is best to have established the criteria of comparison to draw conclusions from. This section outlines and justifies the evaluation metrics by which methods will be compared.

4.2.1 Completion Time. One way to determine the efficiency of the selection methods being tested is to document the time taken by users to complete the selection task at hand. Users had at most 10 minutes to complete a selection. This enforced time limit was to prevent users from spending over an hour in virtual reality. This prevented the user from experiencing extreme fatigue and kept the experiments on schedule.

4.2.2 Selection Accuracy. The aim is to define the extent to which participants were able to recreate the demonstrated mask, and to what extent they made incorrect selections. There exist two common methods for measuring the degree of accuracy of selection methods: F1 score and Matthews Correlation Coefficient (MCC) [11, 20]. Table 1 defines the terms used in these measures.

Table 2: Selection Accuracy terminologies.

Variable	Abbreviation	Description
True Positives	TP	Correctly selected points
False Positives	FP	Incorrectly selected points
False Negatives	FN	Missing points meant for selection
True Negatives	TN	Correctly non-selected points

The F1 score: Lingyun Yu et al. [22] states that *the F1 score calculates the harmonic mean of precision and recall and is often used in information retrieval to measure query classification performance.* It is defined as $F1 = \frac{PR}{(P+R)}$ where P is precision and is calculated by $P = \frac{TP}{(TP+FP)}$ and R is recall (the fraction of volume that was selected) and is calculated by $R = \frac{TP}{(TP+FN)}$. The F1 score produces a value measured between 0 and 1, where 1 is perfect accuracy. Although it is noted that F1 does not take true negatives (TN) into account. The other measure of accuracy, the MCC does factor TN into its results.

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

Both of these were included to better analyze the results, and hopefully draw more accurate conclusions.

4.2.3 System Usability Scale (SUS). SUS is used to measure how the method felt to a user in terms of usability [2]. Users filled out various questions assessing the system strengths and difficulties on a scale of 1-5. These ratings are then involved in a calculation to produce a score in the range of 0 to 100.

4.2.4 Task Load Index (TLX). is a means of scoring experiments with regards to the mental workload experienced by a user during the test. In 1980 Sandra Hart et al. [17] developed the TLX while working for NASA. TLX factors in six areas of assessment with individual weightings and then tallied up to produce the overall TLX score. However, to decrease the time taken to fill out and weight these values, a raw TLX was used. The raw TLX simply calculates the score without weightings. TLX considers mental demand, temporal demand, physical demand, frustration, performance and effort.

4.3 Data Collection

Originally aiming for a sample size of 40 user tests proved to be an ambitious undertaking. After consulting with our supervisors, the scope was decreased to 30 trials due to time constraints. A further five user tests were scrapped, leaving 25 remaining. This loss of five user tests was a result of system failures or mishandling of data during the saving process. Decreasing the sample size of the study reduced the reliability of any conclusions drawn from the results as smaller sample sizes increase the difficulty in identifying significance [9].

For each selection, two researchers recorded the completion time, and the average of these was recorded. After a user had completed their selection for a method, the mask was saved for later accuracy comparisons (F1 Score and MCC) while the user filled out the TLX and SUS questionnaires. The TLX can be used to induce levels of fatigue endured by a user due to the direct link between user satisfaction and reduced cognitive load. The SUS was used to record how usable and friendly each method was to the user.

5 RESULTS AND DISCUSSIONS

5.1 Methodology

To address the research question in Section 1, comparisons between the desktop and VR lasso method were made across accuracy, efficiency and usability. Despite focusing on these two methods, all four methods were tested as one experiment. Conducting a statistical analysis was essential to validating the findings of the user tests. Firstly, the data's distribution was examined with the Shapiro-Wilk [18] and D'Agostino's K-squared tests [7]. The p-value for these tests needed to be greater than 0.05 to indicate normally distributed data. The Shapiro-Wilk test was more heavily weighted in considering the data's distribution due to its accuracy on small sample sizes [5]. The use of parametric or non-parametric tests depended on whether the data was normally distributed. As significance between groups were being tested, either the Friedman(non-parametric) or repeated measures ANOVA (parametric) would be used. The difference being parametric tests generally are more accurate. The normality tests indicated the data of all four methods were not normally distributed. Therefore, the Friedman test was run over all four methods to check for the existence of significance. Significant differences between pairs were studied using the Wilcoxen Signed-Rank test. It indicated significant differences between desktop and lasso with a p-value less than 0.05. Effect size was evaluated using Cohen's d, where 0 represents no effect, and 1 refers to a large effect size. A negative Cohen's d means that the desktop performed worse than the VR lasso.

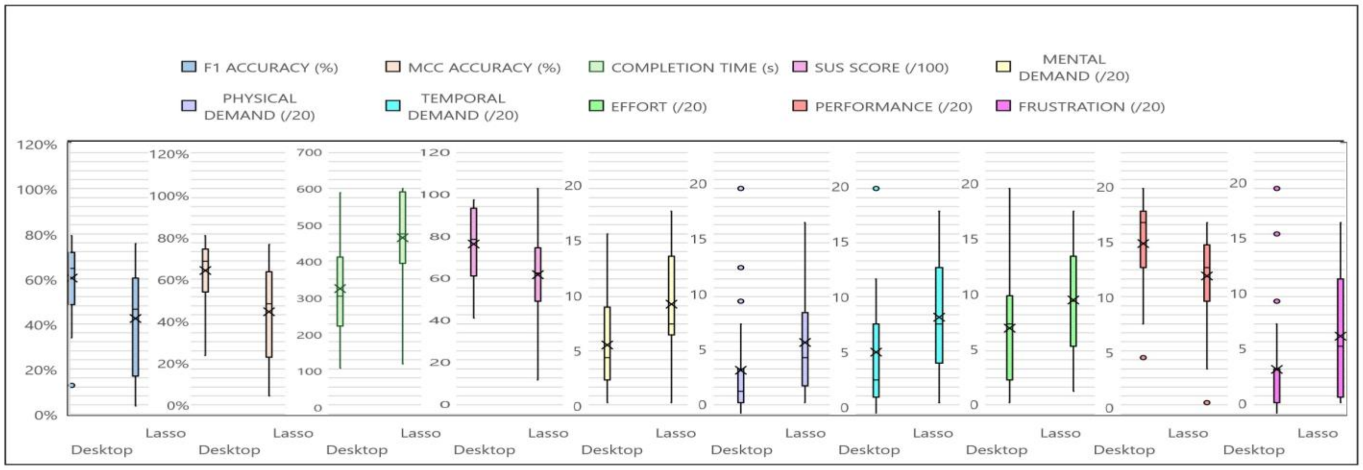

Figure 6: Box and Whiskers diagram for each metric and the six TLX sub metrics.

A post-hoc power test was conducted to assess if a sample size of 25 is adequate to reliably draw conclusions from. The mean and standard deviations (SD) for both methods, across all metrics, have been compiled into Figure 7 as a box and whiskers diagram. Table 3 tabulates the Wilcoxen Signed-Rank p-values and Cohen's d effect size. The results of the data distribution tests are summarized below in Table 4.

Table 3: Summarized Wilcoxen Signed-Rank p-values and Cohen's d effect size.

	Wilcoxen p-value	Effect size
F1 Score	0.00247	0.429
MCC	0.00112	0.462
Completion Time	0.00059	0.487
SUS	0.01217	0.356
Mental Demand	0.0066	0.386
Physical Demand	0.0066	0.407
Temporal Demand	0.0120	0.357
Performance	0.0024	0.430
Effort	0.053	0.275
Frustration	0.0132	0.352

Table 4: Summarized Shapiro-Wilk p-value and D'Agostiono's p-value for desktop and lasso.

Metric	Shapiro-Wilk p-value		D'Agostino's K^2 p-value	
	Desktop	Lasso	Desktop	Lasso
F1 Score	0.016	0.03084	0.0053	0.3180
MCC	0.0105	0.02071	0.0164	0.2724
Completion Time	0.532	0.00580	0.6316	0.0688
SUS	0.01775	0.7072	0.258674	0.678878
Mental Demand	0.003306	0.189	0.11029	0.636982
Physical Demand	0.000	0.02079	0.000	0.164961
Temporal Demand	0.000251	0.23	0.00045	0.38302
Performance	0.000	0.01205	0.02641	0.027994
Effort	0.1055	0.5192	0.298990	0.507302
Frustration	0.000	0.009952	0.000	0.24025

5.2 Accuracy

Accuracy was evaluated with the F1 score and MCC. The lasso method had worse means than the desktop. The Wilcoxen Signed-Rank p-value was less than 0.05, rejecting H0 and accepting significance between the lasso and desktop methods. This is supported by the medium effect size in favor of the desktop as seen with its higher means.

5.3 Efficiency

Referring to figure 6 and table 3, efficiency is measured in terms of completion time. A lower score in completion time indicates a better performance, this swaps the positive and negative interpretations of Cohen's d effect size. The Wilcoxen Signed-Rank p-value is substantially smaller than 0.05, strongly suggesting significance. This is confirmed with a medium Cohen's d of negative value, implying a better desktop performance.

5.4 System Usability Scale (SUS)

SUS scores with the TLX made up the usability component of the methods. The Wilcoxon Signed-Rank p-value was smaller than 0.05, indicating significance. This moderately aligned with Cohen's d effect size. Furthermore, the desktop mean was notably higher than the lasso's.

5.5 NASA Task Load Index (TLX)

TLX followed the trend observed thus far across all six metrics. Similarly, all metrics obtained Wilcoxen Signed-Rank p-values less than 0.05, indicating significance. All metrics had medium Cohen's d effect size approximating 0.3. Following from Cohen's d, the lasso method had worse means across each metric.

5.6 Discussion

This subsection will discuss the results stated in the earlier subsections. There was significance between the desktop and the VR method across almost every metric. Each metric exhibited significance where the Wilcoxen Signed-Rank test rejected its null hypothesis. However, the TLX sub metric of effort was the only factor where the Wilcoxen Signed-Rank test's null hypothesis was accepted. It narrowly exceeded the alpha value of 0.05 with its p-value of 0.053. Subsequently effort had the lowest effect size (0.275). Despite the one sub-metric outlier, the results of the Wilcoxen tests reject H0, instead accepting H1. Having accepted H1 questioned the manner of the significance. To evaluate this H2 and H3 are considered across the metrics.

The accuracy metrics proved the desktop method to outperform the VR lasso method. In both the F1 score and MCC score the difference in method means was close to 20% (17.9 and 18.31, respectively). The worse performance with the lasso can in part be attributed to the conical bug spoken about in 3.3. This sometimes resulted in selections that did not select all voxels within the lasso mesh. The nature of the bug often resulted in users selecting undesired voxels while selecting few of the desired voxels. This was compounded when it occurred during a subtractive selection as it would not select all voxels for subtraction. The cone bug affected 1 out of every 3 users, as such it cannot be ignored in considering the results.

Users further struggled with accuracy due to the locking mechanism of the method. This prevented users from gauging depth by moving the data cubes, like the other VR methods in the experiment. The lasso was significantly worse than the desktop method regarding accuracy. This is further supported by the accuracy metrics having one of the largest effect sizes (0.429 and 0.462). Furthermore, figure 6 shows the larger dispersion of data attesting to the influence of the cone bug. Completion time was the only metric with a larger effect size.

The conical bug similarly affected efficiency as it caused confusion in users. This was a result of the system not working as explained in the training, as well as extra time taken to hopefully fix the bug. This was a time-consuming process that skewed lasso's completion time. Despite this, on average the lasso method took 40% longer to complete a selection than desktop (468.96 to 333.76 seconds). This significance is justified by the small Wilcoxen p-value of 0.00059, indicating significance. This is substantiated by the medium effect size (~0.5) indicating a moderate difference.

The SUS results are indications of how usable the system is to a participant. The results indicate that the desktop method was more usable than the lasso. The desktop's mean was 13.7 % larger. The cone bug created frustration in users and contributed to better desktop ratings. Furthermore, a cumbersome UI, that required many ordered steps, increased the difficulty of using the software. The UI or menu was too complex for users given the 5-minute training allocation per method. Whilst there was significance regarding SUS (p-value: 0.012), the effect size indicates that it was a small significance.

The TLX served to measure the workload endured when using a method to complete a selection. The TLX results indicated significance across five of the six sub metrics, all of which had a small to medium effect. This indicates that the desktop reduced the mental and physical workload of a user compared to the lasso. The increase of workload is expected due to VR being more physically demanding than a desktop. This is furthered by many of the participants having no prior VR experience.

Generally, the worse results exhibited by the lasso method can be attributed to; poor design, method limiting bugs, brief training time, poor participant selection and participants' lack of experience in VR. Despite the lasso's performance, the desktop method also underperformed with its accuracies. This alludes to either both methods performing poorly or poor participant performance. After these findings, a power analysis was performed to determine the feasibility of conclusions drawn from the data with a sample size of 25. The power analysis considers the desired significance level, effect and sample size. It produces a percentage indicating if the sample and effect sizes were large enough to deduce accurate conclusions. 80% is the common deciding value, where greater than 80 indicates the ability to accurately draw conclusions. The power analysis on each metric reported values well below 80%. MCC had the largest power value of 40.17%. This indicates our sample size and effect sizes were insufficient to draw accurate conclusions.

Table 5: Summarized Hypotheses outcomes.

	Null Hypothesis H0:	Null Hypothesis H2:
Accuracy	Rejected	Rejected
Efficiency	Rejected	Rejected
Usability	Rejected	Rejected
Workload	Rejected	Rejected

The desktop method outperformed the lasso method across all metrics. As the significant differences between desktop and lasso are not the result of the VR method improving volumetric selection, H2 was rejected. Alternatively, H3 was accepted, acknowledging that this lasso method hindered users making volumetric selections. Therefore, using VR for volumetric selections did not provide any benefits over a conventional desktop environment in terms of accuracy, efficiency, usability and workload. These results ask the question; would a more complete and better designed lasso system offer any improvements in volumetric selection compared to the desktop method? Whilst this cannot be answered in this paper, the implemented lasso method would undoubtedly perform better without the bugs experienced.

6 CONCLUSIONS AND FUTURE WORKS

This paper aimed to answer the question; Does a virtual reality environment offer any improvements in volumetric selection compared to a desktop environment. The answer to this question was determined through the comparison of a desktop and VR selection method. They were compared according to their performance in accuracy, efficiency, usability and workload. Their performance came from the results of user experiments. Users were given five minutes training with each method before completing a selection task with the method. After each completion, the user filled in a System Usability Scale and a Task Load Index. As such the accuracies, time for completion, usability scores and workload imposed by the method were analyzed. The results indicated that the VR or lasso method was significantly worse than the desktop, across all metrics. The decreased performance in accuracy and completion time is further supported by worse user ratings across the SUS and TLX questionnaires. Despite the power analysis attesting to a small number of participants and effect size, the results produced have significance. Hence, it is concluded that this virtual reality environment offered no improvements over a desktop environment for volumetric selection. Furthermore, it is concluded that this implementation of the lasso selection significantly decreased users' volumetric selections. This decreased performance was noted across accuracy, completion time, usability, and workload. The decrease in performance is attributed to incomplete code, insufficient pilot tests, insufficient testing time and improper participants. The scope of the project prevented further development of code, as well as time-intensive user testing. In future, the lasso system could be improved by fixing the cone bug, allowing the system to be moved with its environment and optimizing the bounding box. Furthermore, integrating this lasso system with the filtering component found in Cloud Lasso and Teddy Selection may prove extremely beneficial. This would allow for some automation from SoFiA in making volumetric selection with a lasso. Despite the results proving the lasso method as a failure, the project was a success as the research question was answered. Additionally, the methods used to conduct the experiments offered insight on how to better explore desktop and virtual reality based volumetric selection.

REFERENCES

[1] Chadia Abras, Diane Maloney-Krichmar, and Jenny Preece. 2004. User-centered design. In Encyclopedia of Human-Computer Interaction, William Bainbridge, Ed. Sage Publications, Thousand Oaks, CA, 445–456.

[2] Aaron Bangor, Philip T. Kortum, and James T. Miller. 2008. An Empirical Evaluation of the System Usability Scale. International Journal of Human–Computer Interaction 24, 6 (2008), 574–594. https://doi.org/10.1080/10447310802205776

[3] Angus Comrie et al. 2021. Virtual Reality and Immersive Collaborative Environments: The New Frontier for Big Data Visualization. arXiv:2103.14397v1. Retrieved from https://arxiv.org/abs/2103.14397

[4] Amy Ulinski et al. 2007. Two Handed Selection Techniques for Volumetric Data. In IEEE Symposium on 3D User Interface. https://doi.org/10.1109/3DUI.2007.340782

[5] Berna Yazici and Senay Asma. 2007. A Comparison of Various Tests of Normality. Journal of Statistical Computation and Simulation 77, 2 (2007), 175–183. https://doi.org/10.1080/10629360600678310

[6] Davide Punzo et al. 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

[7] Ralph B. D'Agostino. 1972. Small Sample Probability Points for the D Test of Normality. Biometrika 59, 1 (1972), 219–221. https://doi.org/10.2307/2334638

[8] Natalia Dużmańska, Paweł Strojny, and Agnieszka Strojny. 2018. Can Simulator Sickness Be Avoided? A Review on Temporal Aspects of Simulator Sickness. Frontiers in Psychology 9 (2018). https://doi.org/10.3389/fpsyg.2018.02132

[9] John Eng. 2003. Sample Size Estimation: How Many Individuals Should Be Studied? Radiology 2 (2003). https://doi.org/10.1148/radiol.2272012051

[10] James Gain. 2000. Enhancing Spatial Deformation for Virtual Sculpting. St. John's College, University of Cambridge, 2000.

[11] Peter Christen, David J. Hand, and Nishadi Kirielle. 2023. A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives. ACM *Comput. Surv.* 56, 3, Article 73 (October 2023), 24 pages. https://doi.org/10.1145/3606367.

[12] Thomas H. Jarrett et al. 2021. Exploring and Interrogating Astrophysical Data in Virtual Reality. Astronomy and Computing 37 (2021). https://doi.org/10.1016/j.ascom.2021.100502

[13] Seunghyok Kim, Jamin Koo, and En Sup Yoon. 2011. Optimization of Sustainable Energy Planning with Consideration of Uncertainties in Learning Rates and External Cost Factors. In 21st European Symposium on Computer Aided Process Engineering. Elsevier, 1914–1918. https://doi.org/10.1016/B978-0-444-54298-4.50161-6

[14] John Lucas et al. 2005. Design and Evaluation of 3D Multiple Object Selection Techniques. Virginia Tech. https://doi.org/10919/31769

[15] Luccia Marchetti et al. 2021. iDaVIE-v: Immersive Data Visualisation Interactive Explorer for Volumetric Rendering. Zenodo (2021). https://doi.org/10.48550/arXiv.2012.11553

[16] Ray Norris. 1994. The Challenge of Astronomical Visualization. Astronomical Data Analysis Software and Systems III. ASP Conference Series, Vol. 61 (1994), 51–56. Bibcode: 1994ASPC...61...51N

[17] Sandra Hart and Loell Staveland. 1988. Developing the NASA-TLX (Task Load Index): Results of Theoretical and Empirical Research. Advances in Psychology 52 (1988), 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[18] Samuel S. Shapiro and Martin B. Wilk. 1965. An Analysis of Variance Test for Normality (Complete Samples). Biometrika 52, 3/4 (1965), 591–611. https://doi.org/10.2307/2333709

[19] Paolo Serra et al. 2015. SoFiA: A Flexible Source Finder for 3D Spectral Line Data. Monthly Notices of the Royal Astronomical Society 448, 2 (2015), 1922–1929. https://doi.org/10.1093/mnras/stv079

[20] Jingxiu Yao and Martin Shepperd. 2020. Assessing Software Defection Prediction Performance: Why Using the Matthews Correlation Coefficient Matters. In Proceedings of the 24th International Conference on Evaluation and Assessment in Software Engineering (EASE '20). Association for Computing Machinery, New York, NY, USA, 120–129. https://doi.org/10.1145/3383219.3383232

[21] Philip Schneider and David H Eberly. 2002. Geometric tools for computer graphics. Elsevier.

[22] Lingyun Yu et al. 2016. CAST: Effective and Efficient User Interaction for Context-Aware Selection in 3D Particle Clouds. https://doi.org/10.1109/TVCG.2015.2467202

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

[24] Lonni Besançon et al. 2019. Hybrid Touch/Tangible Spatial 3D Data Selection. Computer Graphics Forum 38 (2019). https://doi.org/10.1111/cgf.13710

ACKNOWLEDGMENTS
I am grateful to my supervisor Dr. James Gain, and my Co-supervisors Lucia Marchetti and Alex Sivitilli for their guidance and valuable insight offered throughout the project. Thank you to Dr. James Gain for providing the VR headset used to develop and test during this project. Extended thanks to all research participants who partook, as this allowed us to conduct our study. I appreciate the University of Cape Town, for making such a project possible.