Review of Volumetric Selection Methods in 3D Virtual Environments

Thomas van Coller
Department of Computer Science
University of Cape Town
Cape Town, South Africa

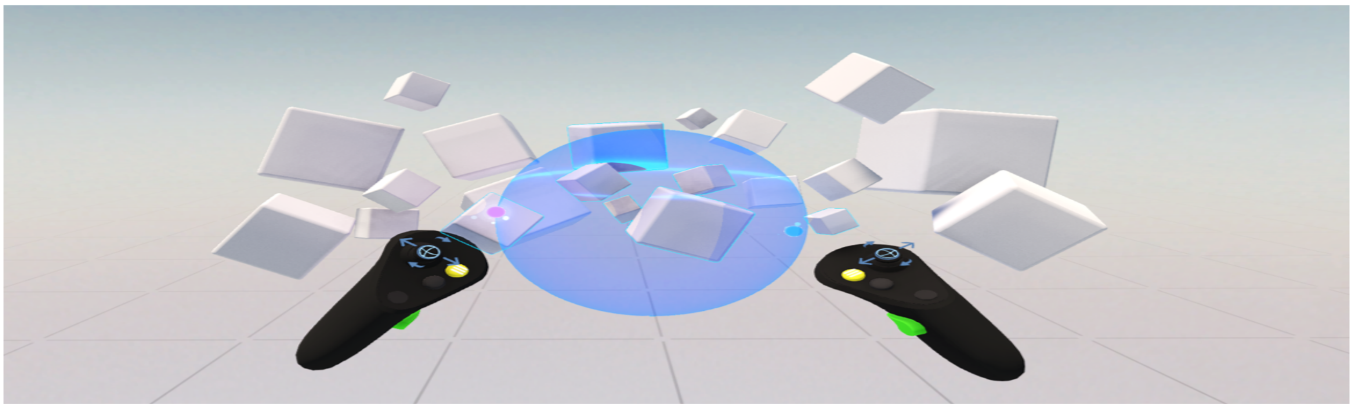

Figure 1: Volumetric selection of objects with a sphere in Virtual Reality

ABSTRACT

A virtual environment, or in other words virtual reality, represents some real-life or tangible concept with computer graphics. Representing volumetric data in virtual reality shows great promise in the visualisation and manipulation of data. To fully embrace these environments, methods of selection need to be incorporated to ensure that users can have control over the data represented to them. There are two core strategies when selecting sources in a virtual environment; object and volumetric selection. Object selection deals with selecting individual objects/sources within the data whereas volumetric selection involves selecting any source(s) within a volume, as demonstrated with a selection sphere in Figure 1. This article serves to review and compare current techniques of selection within a virtual reality. This review will focus on volumetric selection methods, with techniques such as selection by cuboids, lassos, and automatic source detection. The result of assessing these methods shows that lassos are best for dense, large objects whereas cuboids are best for sparsely spread data. Both lassos and cuboids can be used in conjunction with automatic source finding techniques to fine tune the sources detected by the automatic source finder. These methods all have their optimal use cases, ensuring a user has access to as many as possible will ensure the most efficient selection within a virtual environment.

KEYWORDS

Volumetric selection, object selection, volumetric data, virtual reality, volumetric visualisation

1 INTRODUCTION

The volume of data produced by scientific studies every year is substantial, each year producing more than the last. Developing visualisation techniques to represent this data is vital in analyzing the data. In addition to this, subsets of the data need to be selected and isolated to provide adequate analysis. This introduces the idea of selection; given a visual representation of some dataset, how can subsets of this data be selected, isolated and potentially manipulated. Thomas Jarrett et al. [9] states: *It is through visualisation and critical analysis that revelations of underlying and nuanced knowledge and understanding are made.* This speaks to the need for solid visualisation and selection techniques of data.

Visualisation and interaction of multi-dimensional data is applicable to various professions. Some examples of these include regions of science such as astronomy and medicine, engineering and even business. For example, astronomical data is generally collected in two spatial dimensions with an additional spectral dimension [9]. Hence, having methods of visualisation and selection of this data in a three-dimensional space would allow users to isolate sources and critical components of the data for extensive analysis. We call this volumetric data as it can be represented as a volume due it being comprised of three dimensions. In addition to astronomy, medical professionals have found great need for viewing medical data in a three dimensional environment to aid in the care of their patients [7]. Hence, the visualisation of this data collected by astronomers and medical professionals naturally lends itself to virtual reality. Virtual reality allows users to view volumetric data from many perspectives due to the immersion into three dimensions. In addition to this, interaction and selection techniques can be integrated into VR software, allowing users to manipulate their data as they please. Take the medical profession for example, being able to represent a

CT or MRI scan in 3D and slicing this data as needed allows professionals to hone in on critical regions of the scan [7] making the analysis of this data intuitive and efficient.

However, interacting with data in a 3D setting has shown to be challenging. The user enters an environment that mimics reality, but cannot rely on haptic feedback or other senses that one is used to [13]. Moreover, navigating through the virtual space may feel unconventional, making it difficult to view occluded data in the environment [10]. This motivates the need for selection and interaction methods that allow the user to feel as comfortable as possible in the virtual space. With the volume of data ever increasing, research and development into visualising and interacting with said data is ongoing [4].

This review will outline and critically analyse current visualisation and selection techniques for volumetric data. Moreover, the degree to which these methods are useful within a VR environment will be discussed through the comparison of said methods.

2 CHALLENGES WITH VISUALISING AND INTERACTING WITH VOLUMETRIC DATA

Most of us are used to interacting with a 2D work environment, either a desktop or simply writing on paper. Three-dimensional tools for enhancing work and analysis of data can make things complex. In 2D the most navigation you have to do is up, down, left or right and this can all be done sitting in one place. In virtual reality there is a full 360° view, and hence physical movement in the real environment needs to be mapped into the virtual environment to access all view points. This movement in VR to complete tasks or move to different view points can cause physical fatigue and strain. Moreover, if the environment is not set up to be smooth and feel like a real environment in some capacity, it can cause simulation sickness. Natalia Dużmańska et al. [3] defines simulation sickness to be *a syndrome similar to motion sickness and can be experienced as a side effect during and after exposure to different virtual reality environments*. It can cause headaches, eye-strain, nausea and in some cases even vomiting.

Aside from the physical strain that working in virtual reality can cause, even just viewing the data has difficulties. Highly dense environments may experience lots of occlusion [10]. Occlusion is when some objects are obstructed by others in a certain view point. Along with the movement in VR not feeling familiar, navigating through occluded data can prove difficult. The volume of data generated every year is monumental, the forms of which it takes on changing all the time. It is worth considering whether our hardware and software capabilities can continue to grow to support our data needs. This motivates the need for the development and analysis of data visualisation and interaction techniques, with ever-changing data we need people ever-changing and innovating these techniques.

3 METRICS FOR COMPARING VOLUMETRIC SELECTION METHODS

As with any means of comparison, there are metrics commonly used in the evaluation of volumetric selection methods. Having these metrics is vital as new methods of selection need to have some form of a benchmark to evaluate their usefulness compared to existing methods. Metrics for volumetric selection assess both user experience and performance when using a given selection mechanism. These metrics are as follows:

3.1 Completion Time

Experimentally, volumetric selection methods are tested with sample environments in which users are asked to select various objects with different selection mechanisms. The completion time refers to the time it takes to complete a given task. Completion time is often taken for both individual tasks (e.g. time to select an object and then time to deselect the same object [13]) as well as an entire test as a whole.

3.2 Selection Accuracy

Measuring the extent to which a given selection technique enables the user to select the object(s) they intend to select is vital in evaluating the technique. This is what selection accuracy entails; observing the degree to which the intended selections are made against the actual selection performed. Two metrics are commonly used to assess selection accuracy: The F_1 score and the Matthews correlation coefficient (MCC). Before expanding on these two metrics, I will outline a few variables used in calculating the metrics:

Table 1: Variables for Selection Accuracy

Variable	Abbreviation	Description
True Positives	*TP*	Correctly selected points
False Positives	*FP*	Incorrectly selected points
False Negatives	*FN*	Missing points that had to be selected
True Negatives	*TN*	Correctly non-selected points

3.2.1 F_1 score.

Lingyun Yu et al. [17] states that the F_1 *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 $F_1 = P \cdot R/(P+R)$ where P is precision and is calculated by $P = TP/(TP + FP)$ and R is recall (the fraction of particles that were selected) and is calculated by $R = TP/(TP + FN)$.

3.2.2 MCC.

The F_1 score does not take TN into consideration, MCC provides a metric in which it is. MCC is defined as follows:

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

Both of these metrics are commonly used in conjunction with one another when evaluating the selection accuracy of a selection technique. Papers where these metrics are included are [2, 5, 13, 17].

3.2.3 Over/Under selection.

Over and under selection is a subcategory of selection accuracy in which only the FP and TN fields are examined to get a sense of to what degree the user is selecting too many or too little points.

3.3 Task Load Index (TLX)

A task load index (TLX) is a tool for measuring and assessing the mental workload a task has on the user performing that task. TLX was developed by NASA and is sometimes referred to as NASA-TLX

[11]. TLX is evaluated on a few sub-metrics that are then weighted and combined to form the overall TLX value. These sub-metrics are mental demand, physical demand, temporal demand, effort, performance and frustration. For each task observers (usually those conducting research) watch the user while they perform the task and assign a rating to each field. Each field is then weighted, the weighting of each of these in the final calculation is decided by the observers. Hence, TLX allows for the mental workload of a task to be assessed from many different viewpoints, providing insight into the strain the task puts on the users. However, this test is limited by its subjectivity and could cause varied results based on the observer.

3.4 Fatigue

Fatigue measures the physical strain a task creates for a user. Measuring fatigue involves a qualitative questionnaire that the users fill out and indicate which body parts experienced the most fatigue [2, 5]. Fatigue is also subjective to the users partaking in the experiment; users with longer endurance may not experience fatigue that others do. It is important to have a variety in the users that undergo the task.
This concludes the metrics for evaluating volumetric selection methods. Table 2 gives an overview of the selection metrics discussed.

Table 2: Metrics for Volumetric Selection

Metric	Description
Completion Time	How long a given task took to complete
Selection Accuracy	Intended selection vs actual selection
Over/Under Selection	Subcategory of selection accuracy
Task Load Index (TLX)	Assesses mental workload of a task
Fatigue	Measures physical strain on users

4 VOLUMETRIC SELECTION

When visualising data in any form, methods to select parts of that data for analysis and manipulation are key to the interaction with the data. There are many methods that have been developed to do such a thing. Selection in a two-dimensional environment such as a desktop screen is very straight forward; a cursor is simply used to make selections. This can either be done by clicking on your desired selection, or in some cases creating a rectangular area to select any items within that area. When moving into a three-dimensional space such as virtual reality, selection is not as intuitive. However, two of the primary methods of selection in three dimensions are analogous to that of two dimensions. Namely, we have object selection and volumetric selection. Object selection can be likened to that of making a selection by clicking with a cursor in two dimensions. In three dimensions we can make use of rays that users can point at the object they require to select [10]. There is a lot more to object selection, the details of which are not covered in this review as the focus is methods of volumetric selection. We can compare volumetric selection to that of selecting items within an area in a 2D environment. However, in three dimensions we are dealing with volumes and not areas. Instead of selecting with a rectangle we could select with a cuboid, or even a sphere. Many of these methods will be discussed shortly. Being able to select or deselect objects with volumes allows users to fine-tune selection of one or more objects precisely and efficiently. Volumetric selection provides benefits in dense environments with many objects close together, especially in comparison to object selection [13].

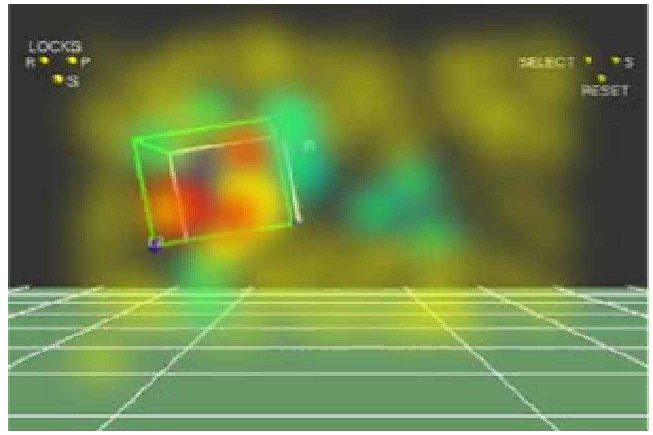

Figure 2: Performing a selection with a cuboid, taken from [5].

4.1 Cuboid Volume Selection

As mentioned above, volumetric selection in three-dimensions can be approached by constructing three-dimensional shapes around the desired selections. This subsection aims to explore some of these methods. In 1996, David Ebert et al. [6] explored the interaction of glyph-rendered data in a 3D environment. They made use of two magnetic trackers with buttons to select data in the environment. The procedure for selecting a source is as follows: the left hand controls the view point of the scene while the right hand is used to make a selection. This method is attributed more to object selection, but served as a basis for the selection techniques developed by Amy Ulinski et al. [5]. Ulinski et al. developed three methods for selection with cuboids: Hand-on-Corner (HOC), Hand-in-Middle (HIM), and Two-Corners (TC). These methods made use of two magnetic trackers similar to Ebert [6], except these trackers were used to create cuboids for selection. Figure 2 illustrates what this looks like in their software.
The three methods for selection worked as follows: HOC placed the non-dominant hand of the user onto the bottom left/right (depending on the side of the non-dominant hand of the user) corner of the front face of the cuboid and controlled the positioning of the box. The dominant hand was placed in the top right/left corner (depending on the side of the dominant hand) of the back face of the cuboid and controlled the size of the box. The size was increased or decreased by moving the dominant hand relative to the non-dominant hand. HIM places the non-dominant hand in the middle of the box to control positioning and orientation. The dominant hand is not assigned a position but is rather moved relative to the non-dominant hand like in HOC. TC places the user's hands in the same way as HOC except both hands are used together to control the positioning, size and orientation of the box. Ulinski et al.

[5] conducted an experiment to see which one of these methods performed the best. The results of this experiment indicated that TC had the best accuracy of the three, while HIM had the quickest completion times for the selections. Moreover, TC had the greatest workload strain while HIM had the least. Users also stated that HIM felt the most natural of all the methods. Note that these methods were not developed in virtual reality, but could easily be adopted in VR.

Robert Montano et al. [13] developed a hybrid cuboid selection technique. In this case hybrid refers to interacting with both the VR environment as well as some tangible hardware to make a selection. This method used two controllers, one with a pen and the other with a tablet attached. A method using a virtual tablet within VR was also tested. The idea is that the user selects object(s) using a cuboid in VR, this is then mapped to the tablet to make more precise selections. Both the real and virtual tablet methods were compared against a "mid-air" (selection made only in VR without use of a tablet) selection technique similar to that of Ulinski et al. [5]. The experiment conducted showed that overall the hybrid technique provided significant benefit in environments with highly occluded data. In environments that did not have lots of occlusion, the hybrid technique showed minimal benefit over a mid-air technique. In addition to this, the method using a real tablet showed greater accuracy than the mid-air technique, the virtual tablet did not have any significant improvement however. Moreover, the real tablet increased completion time and overall workload strain compared to the other two methods.

Comparing the hybrid technique to the two-handed selection techniques presented by Ulinski et al. [5] is difficult since they were tested in different virtual environments and with different experiments. However, based on the similarity of the mid-air technique and these two-handed techniques one could argue that the hybrid method would provide an improvement over the two-handed techniques in highly occluded environments. A formal experiment to substantiate this would need to be conducted.

4.2 Lasso Methods

It was seen in the previous section that volumetric selection can take place by containing selections within 3D shapes. While this method provides a simple, intuitive means of selection, it can have drawbacks when the shape of the data is not cuboidal. This section deals with a set of selection techniques called lasso selection. Lasso selection allows users to draw a lasso around objects they want to select and all those contained within the lasso are selected. In a 2D environment this is quite simple as the surface you can draw the lasso on and the physical environment you are working on are the same. In 3D however, it is not so straightforward as you need to introduce some sort of 2D surface for the user to draw a lasso.

John Lucas et al. [12] developed a tablet freehand lasso technique, Cylindrical Selection. This allows the users to have a 2D view of part of the 3D environment on a virtual tablet. Users can then draw a lasso on this tablet that is then projected to the 3D space for a selection to be made. Figure 3 illustrates an example of how this selection is done.

However, cylindrical selection does not factor in the spatial structure of the object(s) being selected, it simply selects all that falls within the cone mapped from the drawn lasso. Lingyun Yu and Konstantinos Efstathiou [17] developed two lasso methods, Cloud Lasso and Teddy Selection, that can construct an estimation of the spatial structure of a selected object. In addition to this, they conducted a user experiment to directly compare these methods to Cylindrical Selection. Before diving into the comparison of these methods, I will outline how they work. Teddy Selection splits the cone created by the lasso into many bins. A density threshold is used to only include bins that are above the threshold. Triangulation of the bins then allows for an estimation of the object shape to be interpolated. Teddy Selection does not allow for this threshold to be adjusted nor does it let the user adjust the depth the cone should be limited to. This makes this method prone to including unwanted selections in the selection process, more specifically dense objects hidden behind the intended object being selected. Cloud Lasso aimed to improve this by allowing the threshold and depth to be adjusted by the user. Moreover, Cloud Lasso implemented a new method for extracting the spacial structure of an object. Instead of bins, a minimal cuboid is constructed around the desired object for selection. This cuboid is then split into many smaller cubes, each of which undergo a density calculation. Those cubes above the density threshold are kept in the selection while the others are not. These methods do not differ from the perspective of the user as they only differ in the way they extract the shape which is abstracted from the user. It was found that Cloud Lasso produced smoother shape estimations than Teddy Selection.

Hence, Yu and Efstathiou decided it best to only compare Cloud Lasso with Cylindrical Selection. The results of the experiment done by Yu and Efstathiou [17] showed that Cloud Lasso produced faster completion times for selection than Cylindrical Selection. This is because Cylindrical Selection selects everything within the lasso cone, and hence required a lot of fine-tuning. Moreover, the F_1 scores and MCC values measured indicated that Cloud Lasso is significantly more accurate than Cylindrical Selection. All participants in the test chose Cloud Lasso as their favourite technique. However, Cloud Lasso is computationally expensive due to the density calculation within the cubes. It is mentioned that parallelising these computations could significantly improve this. It was also found that methods like Cloud Lasso and Teddy Selection do not work well with sparse data.

Cloud Lasso and Teddy Selection rely heavily on finding a suitable view point to draw the lasso, nor do they take the shape of the lasso into consideration for the selection. Yu and Efstathiou collaborated with two others, Petra Isenberg and Tobias Isenberg [16] to develop methods to combat these issues. SpaceCast was introduced as an alternate lasso technique and compared to Cloud Lasso and Cylindrical Selection. SpaceCast aimed to improve selection by only selecting shapes similar to the shape created by the drawn lasso [16]. This is done by projecting the shape of the volume being selected onto a 2D plane and comparing the that shape to that of the drawn lasso.

An experiment was conducted to compare SpaceCast to both Cloud Lasso and Cylindrical Selection. The results of this experiment indicated that SpaceCast produced faster completion times than the previous two methods. This experiment showed that SpaceCast had no improvement in the accuracy of selection compared to the other two methods. Moreover, users reported that SpaceCast was more

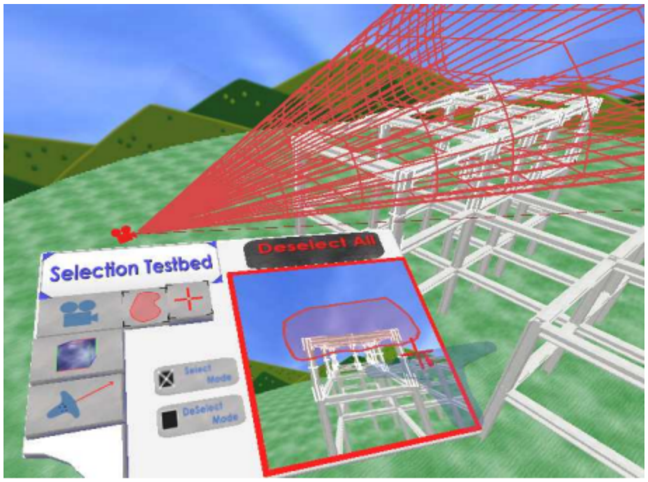

Figure 3: Tablet freehand lasso technique: Cylindrical Selection. Taken from: [12].

effective in selecting targets in occluded environments than Cloud Lasso and Cylindrical Selection.

While SpaceCast, Cloud Lasso, and Cylindrical Selection have shown that selection with a lasso is intuitive and beneficial, they all require the user to draw a lasso within the virtual space. Lonni Besançon et al. [2] developed a hybrid environment for lasso selection called TangibleBrush. This hybrid environment is similar to that of Montano et al. [13] and makes use of a physical tablet to draw lasso's that are then mapped to the 3D environment. In contrast to Cloud Lasso and SpaceCast, this method requires the user to physically rotate the view port and draw multiple lasso's to select the object whereas the other two methods extrapolate the object with only one lasso.

An experiment was conducted by Besançon et al. [2] to compare TangibleBrush to SpaceCast. The results of which showed that although SpaceCast produced faster completion times, TangibleBrush produced more accurate selections. However, this improvement in accuracy was minimal. Moreover, the mental workload from the TLX assessment showed to be greater for TangibleBrush, while all other sub-metrics of TLX showed no difference between the two. Furthermore, they found that the fatigue caused by the methods was similar. TangibleBrush proved most beneficial in selecting more complex datasets like galaxies. While no direct comparison was made from TangibleBrush to Cloud Lasso and Cylindrical Selection, these can be interpolated from the previous experiment between SpaceCast and these two methods.

4.3 Automatic Source Detection

All of the methods dealt with up to this points involve users selecting objects on their own using various methods. However, this does not necessarily need to to be the case. Methods of automatic source (object) detection for 3D data have been developed, allowing users to have their objects selected for them before even seeing the data. Ideally there would be an algorithm that can accurately identify any source from any set of data, however this is not the case. There are various trade offs that need to be made when fine-tuning a source finding algorithm [1]. This section aims to explore the automatic source finding software that already exists. The first of which being SoFiA, developed by Paola Serra et al. [8]. SoFiA is a general source finding software for 3D spectral line data. It was developed with the purpose of detecting sources in astrophysical data, more specifically H1 emission data. SoFiA provides many source-finding techniques, combining many algorithms to optimise the selection of sources. The structure of the software is modular, users can pick and choose the techniques they need for their respective data. It can also be used independently of the H1 data provided to it.

A competitor of SoFiA is MTObjects, developed by Paul Teeninga et al. [15]. MTObjects is a region-based source finding software. It makes use of a data structure called a max tree. Roberta Souza et al. [14] defines a max tree as *a mathematical morphology data structure that represents an image through the hierarchical relationship of connected components resulting from different thresholds*. The max trees for MTObjects are structured in such a way that sources can easily be grouped and related. Max trees provide an advantage in that they can filter the cube data without distorting any edge information. This is because the operations can happen between the relationships of connected data points instead of the individual voxels [1].

Barkai et al. [1] conducted an experiment to compare SoFiA and MTObjects as to find the superior source detection method. This experiment also compared the two to a machine-learning model VNet trained to detect sources. The results of this experiment showed SoFiA to have the fastest computing time compared to MTObjects and VNet. This was not considering the time that VNet spent training on data. SoFiA was also found to be the best at identifying real sources whereas VNet was best at finding "mock galaxies" which were data sets artificially created for testing purposes. VNet was trained on this data, and hence performed best on the tests using this data. MTObjects was found to identify the most incorrect sources of the three. However, since real sources are what we are most concerned about, SoFiA showed to be the most optimal source finding technique. That being said, it is speculated that with more training the VNet method could surpass SoFiA on most metrics.

5 DISCUSSION

In this review various techniques for volumetric selection have been observed. Cuboid, lasso and automatic source selection techniques were discussed. This section aims to provide an overview of these overarching ideas and how we can compare and contrast them. One may think we are looking for one "superior" selection technique. However, looking at selection in such a way is very limiting. In reality we want to identify techniques that can be used in conjunction with one another to provide the user with the most efficient, intuitive environment. With that in mind, while identifying where these methods shine and fall short, I will also be analysing how they can be used together to provide the user with the ideal set up for selection of volumetric data.

Automatic source detection is a good place to start with selection. This is because in most cases, starting with an automatic source detector can significantly lower the workload of a selection task for a user [8, 15] by allowing the user to fine-tune an already selected source. This is where combining several techniques can shine. The fine-tuning can make use of various object and volumetric selection techniques that the user can pick from to optimise their experience. For example, cuboidal selections can be combined with boolean operations to select and deselect voxels around the automatically selected source. The main drawback of the automatic source detection techniques discussed is that they have both been developed for the specific context of astronomy. These techniques would need to be extended to fit the needs of other professions. That being said, the development of automatic selection techniques could dominate the selection space, as it has the potential to be the most efficient selection methodology [1].

Although, the automatic methods are not perfect and require the assistance of other methods to shine. Cuboid selection methods have shown to be best in selecting sparse and regularly shaped data [16, 17]. This is because data that is densely packed might be hard to select without overlapping unwanted objects. Moreover, oddly-shaped data may not lend itself to selection with a cuboid. In addition to this, the user needs to navigate to an ideal view point to select the with a cuboid, which may not necessarily be obvious. This could lead to increased completion times, accuracy issues, and fatigue. Hybrid techniques for selection have been introduced as in [13]. This method makes use of tangible hardware to aid in the selection process, which showed to have benefits in highly-occluded and dense environments.

Lasso techniques are able to deal well with dense environments as the user has full control in outlining the area of selection they desire. The experiment done by Yu et al. [16] showed that 80% of their test users said that lasso selection feels like the most natural way to go about selection. Although both lasso and cuboid selection techniques rely on having a good view point for selection of the data, the ease of drawing a lasso as opposed to moving both hands to control a cuboid [5] could allow for lasso techniques to have faster completion times and less fatigue. The use of a hybrid technique [2] showed to improve accuracy in selection as well. However, in sparsely spread data sets lasso methods perform poorly [2, 16, 17]. This is due to the density calculations used by lasso techniques; sparse data would have very low density and so these calculations would not be able to pick up sources as accurately as cuboid selection methods. The main idea I aim to get across from this is that different selection methods shine in different settings. Ensuring that your user has as many options and techniques to apply to a selection is vital. Allowing the user to have access to multiple techniques also encourages using many techniques together to complete a task to optimise the selection process.

6 CONCLUSIONS

This review has served as a deep dive into methods of volumetric selection for virtual environments. Multiple techniques have been observed and critically analysed. These methods have included two-handed cuboid selection techniques [5], a hybrid cuboid selection [13], lasso techniques [12, 16, 17], and automatic source detection techniques [1, 8, 15]. Exploring these methods has illustrated the value studying and developing selection techniques holds. Data is arguably one of the most important parts of current society and is at the base of most if not all of the systems our world has in place. Developing methods of visualisation and interaction with this data that align with the format of the data is imperative. Three-dimensional data is at the forefront of this idea with it being relatively new. The space for volumetric visualisation and selection techniques has endless possibilities, only left to the creativity of researchers to explore. While this review only dealt with current methods for volumetric selection, the ideas and results drawn from analysing these methods can be applied to develop better methods in the future. Fine-tuning automatic selection techniques holds great potential for success in this field. Moreover, testing techniques for integrating multiple volumetric selection techniques could allow for further improvement in the efficiency and accuracy of selection.

REFERENCES

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

[2] Lonni Besançon, Mickael Sereno, Lingyun Yu, Mehdi Ammi, and Tobias Isenberg. 2019. Hybrid Touch/Tangible Spatial 3D Data Selection. *Computer Graphics Forum* 38 (2019). https://doi.org/10.1111/cgf.13710

[3] 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

[4] Arie Kaufman et al. 1994. Research issues in volume visualization. *IEEE Computer Graphics and Applications* 14, 2 (1994), 63–67. https://doi.org/10.1109/38.267473

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

[6] David Ebert et al. 1996. Minimally-immersive interactive volumetric information visualization. *Proceedings of the IEEE Symposium on Information Visualization 1996*, 66–67. https://doi.org/10.1109/INFVIS.1996.559220

[7] Holger Grees et al. 2000. 2D and 3D visualisation of head and neck tumours from spiral-CT data. *European Journal of Radiology* 33, 3 (2000), 170–177. https://doi.org/10.1016/S0720-048X(99)00121-7

[8] Paolo Serra et al. 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

[9] Thomas 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

[10] Tony Grossman and Ravin Balakrishnan. 2006. Design and Evaluation of Selection Techniques for 3D Volumetric Displays. *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology* (2006). https://doi.org/10.1145/1166253.1166257

[11] 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

[12] John Lucas, Doug Bowman, Jian Chen, and Chad Wingrave. 2005. Design and Evaluation of 3D Multiple Object Selection Techniques. *Virgina Tech* (2005). https://doi.org/10919/31769

[13] Robert Montano, Cuong Nguyen, Rubaiat Habab Kazi, and Sriram Subramanian. 2020. Slicing-Volume: Hybrid 3D/2D Multi-target Selection Technique for Dense Virtual Environments. *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2020), 53–62. https://doi.org/10.1109/VR46266.2020.1581198507712

[14] Roberto Souza, Luís Tavares, Letícia Rittner, and Roberto Lotufo. 2016. An Overview of Max-Tree Principles, Algorithms and Applications. *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, 15–23. https://doi.org/10.1109/SIBGRAPI-T.2016.011

[15] Paul Teeninga, Ugo Moschini, Scott C. Trager, and Michael H.F. Wilkinson. 2016. Statistical attribute filtering to detect faint extended astronomical sources. *Mathematical Morphology - Theory and Applications* 1, 1 (2016). https://doi.org/10.1515/mathm-2016-0006

[16] Lingyun Yu, Konstantinos Efstathiou, Petra Isenberg, and Tobias Isenberg. 2016. CAST: Effective and Efficient User Interaction for Context-Aware Selection in 3D Particle Clouds. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 886–895. https://doi.org/10.1109/TVCG.2015.2467202

[17] 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