May 26, 2022

An Exploration of Immersive Smoothed-Particle Hydrodynamics Data Visualisation in Astrophysics and Astronomy

Final thesis submission for the
Bachelor of Computer Science (Honours) programme.
Unit code: FIT4444 & FIT4448.

Final word count (Overleaf TeXcount): 7858 words.

Author: Nhan H.T. Luong[1] – **Student ID:** 29644364
Supervisors: Daniel J. Price[2] & Barrett Ens[1] & Maxime Cordeil[3]

[1]Faculty of Information Technology, Monash University
[2]School of Physics & Astronomy, Faculty of Science, Monash University
[3]School of Information Technology and Electrical Engineering, The University Of Queensland

MONASH University

Melbourne, Australia

Acknowledgements

I would like to express my deepest appreciation to my parents, who have allowed me to extend my time at university for another year to pursue my lifelong passion for science and research. Without their unwavering support, none of this would have been possible.

Special thanks to Prof. Daniel Price, Dr. Barrett Ens and Dr. Max Cordeil for agreeing to supervise a rather unconventional project, which enabled me to learn a great deal about topics in which I have long been interested.

I would also like to recognise the assistance that I received from my friends, Hoang-Anh Nguyen, Bach Ha, Thanh Le, and Linh Phan, who have helped and supported me along the way even though we're continents apart.

Important notes

The project was initially decided to produce a functional software package, however during development many unexpected challenges occurred and due to time limitations, we decided to pivot into an exploratory project. This is why the research goal stated in the Literature Review is different from the final paper.

In regards to the use of the pronoun *'we'* instead of *'I'*, in the thesis paper I used *'we'* to be consistent with the language used in journals and conferences. All work, including implementation, conducting the study, and writing, was done by me while the supervisors provided directions and guidance.

The targeted venue is ***The IEEE Visualization Conference (VIS)***.

CONTENTS

Literature Review

1 Introduction

Understanding how gases move and behave is essential to astrophysics research as it is the underlying mechanism behind most fundamental processes like the formation of planets, stars and galaxies (Clarke and Carswell, 2007). One common type of data for this research is mesh-free (Garg and Pant, 2017) particle-based data consisting of tabular information with each row corresponding to properties of a specific particle in the system, which often comes from astrophysics simulations (Jin et al., 2010; Navratil et al., 2007; Brown et al., 2021; Makino et al., 2003). To be able to visualise such data is essential for astronomical research.

Computer graphics introduced a vast range of possibilities not just for astronomical visualisation but scientific visualisation in general, one of which is to display data in three spatial dimensions. 3-D graphical representation of N-body simulations, planetary surfaces, and data cubes of galaxies from NRAO radio telescope can greatly benefit scientific analysis, as explored by Kent (2013). Traditionally, 3-D plots have been presented on 2-D computer screens which reduced the amount of information one can observe at a moment in time by one dimension. Virtual reality (VR) has been explored by researchers as a potential solution to this limitation for decades. One early instance of such investigation was by Bryson (1996), defining *Virtual reality* as an interface that allows for 3-dimensional interactivity with virtual objects and exploring the potential of real-time natural interactivity with abstract data sets. *Immersive analytics* research discussed the benefits of allowing the ability to fully immerse oneself in the visualisation (e.g. manipulate the data with intuitive hand gestures, walk and look inside the spatial structure of data, haptic feedback for better understanding of non-visible properties) for scientific visualisation (Chandler et al., 2015). This motivates a need to explore the possibility of bringing this technology to astronomical research and this literature review aims to investigate the state of the art of VR application in the domain, specifically for particle-based visualisation. There exist various definitions for VR which have changed over time and some variations will be seen in the papers referenced. The definition of VR that will be used throughout this review is closely related to the definition of immersive experience given by Bryson (1996) with personal VR head-mounted displays (HMDs) like the Oculus Rift[1], HTC Vive[2], and others.

There exists a large collection of scientific visualisation tools that are used extensively by astronomers and astrophysicists such as VTK[3], ParaView[4], VisIt[5] and 3D Slicer[6] but they lack *desirable* VR capabilities[7]. These programs are open-sourced and massively supported by the community, most of which are general-purpose visualisation programs developed in other fields such as medical research. Only a small number of programs were developed specifically for particle-based data. Over the years, there have also been a great number of projects on astro-visualisation on immersive displays such as CAVE, VR, and AR. The majority focused on public outreach as the primary goal. Science-focused VR projects in astronomy and astrophysics are far and few in between. They usually are not full-fledged software packages but only prototypes with low usability and extendability. This project addresses this by utilising the Immersive Analytics Toolkit developed for Unity[8] by Cordeil et al. (2019) with the purpose of scalable, responsive, and accessible data visualisation in VR. By building an astrophysics-focused open-source add-on to the toolkit, the goal of having a widely usable and cohesive platform to visualise particle-based data sets in VR is achievable thanks to the *relatively short*[9] software development time.

The review will first explore the broad range of visualisation technology in astronomy and astrophysics in Section 2.1, then to investigate the advantages of immersive analytics in Section 2.2, and which areas of astronomy and astrophysics had VR been applied to in Section 2.3. After the substantive literature review, the state of the art of VR visualisation in astrophysics and astronomy will be discussed in Section 3. Section 4 describes an initial plan for the project, including how IATK and SPLASH can help achieve the task in the given time limit, the key existing challenges that need to be solved, and a planned timeline to provide the solutions.

The review will not discuss in-depth details of the software applications mentioned such as their

[1] https://www.oculus.com/
[2] https://www.vive.com/eu/product/vive-pro/
[3] https://vtk.org/
[4] https://www.paraview.org/
[5] https://visit-dav.github.io/visit-website/index.html
[6] https://www.slicer.org/
[7] Such criteria will be discussed in later sections.
[8] A brief description of Unity is discussed in Sectioned 4.1.
[9] Compared to building a platform from scratch.

rendering techniques, implemented algorithms, performance metrics or other technical details. Rather, it will only focus on high-level aspects of such programs that are relevant to the topic at hand. Literature involving geographical rendering of data or environmental visualisation involved with aerospace research and planetary exploration will also not be included as it is inherently different from the topic of interest. Visualisation of imaging data formats such as FITS and the tools commonly used for such data types like SAOImageDS9 (Joye and Mandel, 2003) will also be omitted from the review.

2 Substantive Literature Review

Excellent surveys exist that investigate visualisation in astronomy, each with a different approach to categorising literature. Most recently, Lan et al. (2021) discussed literature from 2010 to 2020 while classifying the papers by data analysis tasks and visualisation techniques. Hassan and Fluke (2011) explored in-depth visualisation techniques for N-body simulations and spectral cubes visualisation from 1990 to 2010 and classified the literature by data types. Lipşa et al. (2012) conducted a survey of visualisation for the physical sciences, which included astronomy literature classified by the problem they aim to solve. All three articles addressed the same main challenges of dealing with the large amount of astronomical data with rapid growth size in the advent of more advanced telescopes, this out of scope for the project. This review will have a more general and interdisciplinary approach to the classification of literature with the purpose of constructing an argument to support the development of a new VR visualisation tool for astronomical data.

2.1 Visualisation software for astronomical data

As it is not practical to discuss every visualisation software used in astronomy and astrophysics, this section will discuss only ones that are prominent in the field with focus on programs supporting particle data. The goal is to get a sense of the existing technology, what is popular and what is commonly used for the project's purpose.

Regarding the problem of selecting software to visualise an astronomical data set, the approach researchers may take can generally be categorised as one of the following (Hassan and Fluke, 2011):

1. Use an existing application,
2. Modify or build extensions based on an existing application or
3. Build an entirely new application.

As it is unfeasible to create new software and packages for every project, the approaches listed are in order of popularity. From the reviews by Lan et al. (2021), Hassan and Fluke (2011), Lipşa et al. (2012), it can be seen that it is a common practise for astronomers and astrophysicists to use and build custom plugins for general-purpose scientific visualisation applications such as ParaView, VisIt, 3D Slicer, and others.

A well-known toolkit specialised for astrophysics simulations data is *yt* (Turk et al., 2010) made to visualise data from Enzo (Bryan et al., 2014) - a code for mesh-based fluid flows simulations, as opposed to particle-based simulation methods like Smoothed Particle Hydrodynamics (SPH) (Monaghan, 1992). Borkin et al. (2007) discovered major similarities between medical imaging and astronomical visualisation which lead to the use of 3D Slicer and Osirix[10] in astronomical research. Unfortunately, these programs also do not support particle-based data sets.

Another commonly used program is ParaView, a stable and highly optimised open-source application built on VTK that provides a massive set of features for interactive visualisation of large-scale scientific data. ParaView comes included with particle-based data support for formats like GADGET or cosmo. Example use cases of ParaView are Navratil et al. (2007) and Woodring et al. (2011) for visualising particle-based cosmological simulations. Web-based and multiplatform visualisation is also included in the software from the get-go. ParaView provides VR capabilities, although as explored by Sua et al. (2015) and Jensen et al. (2021), it still has some limitations and requires further development to be easily accessible with a straightforward workflow before it can be widely utilised for scientific visualisation.

Similar to ParaView are EnSight[11] and VisIt. They lack ParaView's high parallel performance, making the programs less appealing for large particle-based visualisation (Woodring et al., 2011). EnSight's main focuses are on engineering physics simulation with a highly interactive workflow, including real-time isosurface and streamline rendering for fast and efficient data analysis. Un-

[10] https://www.osirix-viewer.com/

[11] https://www.ansys.com/products/fluids/ansys-ensight

like ParaView, EnSight is not open-source which limits its customisability and therefore limits the potential usefulness for astrophysics. On the other hand, VisIt is an open-source visualisation tool built on VTK, like ParaView. The VisIt user interface does not include highly interactive 3-D viewports like EnSight but it does provide most functionalities needed to generate 3-D animated visualisations with tools to support data analysis like point/line/zone selection. VisIt has support for particle data but most of its emphasis for optimisation and development is on mesh-based data.

The paper by Kapferer and Riser (2008) also mentioned real-time visualisation programs MayaVi and IFrIT, the details are in line with other general-purpose tools previously mentioned.

A less feature-packed alternative that is also commonly used is TOPCAT[12] (Taylor, 2017), written in Java with a specific purpose of rendering tabular data. It was used by Taylor (2019) to visualise the second Gaia data set and due to the *relative simplicity*, it was used by Pössel (2020) to introduce astronomical data analytics to beginners (including high school students). TOPCAT has a rather simple Graphical User Interface (GUI) with ability to render contours, density maps, error bars, transparency, etc.

VisIVO (Becciani et al., 2006) is an open-source visualisation tool written in C++ and specifically designed with the purpose of intergrating the retrieval of data from Virtual Observatories (VOs), which are online databases of astronomical data sets aimed to provide publicly available data. Other programs in this review also support such feature but are omitted due to being of little relevance to the main topic. VisIVO supports both mesh-based and particle-based data with a user-friendly GUI akin to that of graphics software like Adobe Photoshop (Becciani et al., 2006).

Partiview (Levy, 2003, 2010), Splotch (Dolag et al., 2008), and SPLASH (Price, 2007) are all particle-based visualisation tools with different focuses. Partiview provides basic interactions with the plotted data like toggling visibility of groups of particles, variable colour and basic stereoscopic 3-D rendering. The program has a dated GUI with limited functionalities and most importantly unable to render particle data as continuous structures (Fairall, 2005a). It is mainly used by planetariums for public outreach (Abbott et al., 2003; Fairall, 2005b) or for educational purposes (Faherty et al., 2018).

[12]http://www.star.bristol.ac.uk/ mbt/topcat/

Splotch (Dolag et al., 2008) is a ray-traced visualiser designed for cosmological SPH simulations running on high-performance computing infrastructure, thus has no GUI. The purpose of ray-tracing is to add *perceived realism* to the visualisation.

SPLASH is also a commonly used visualisation tool for SPH data. Written in FORTRAN with a CLI, an interactive mode, ray-traced 3-D surface rendering, and a custom-built plotting library, it was created specifically for SPH data with the ability to render particles as a smooth continuum using SPH kernel, render multiple plots with support for cross sections and vector fields, simple production of animations, and support direct binary or ASCII parsing with an utility to convert data files.

There are also plotting utilities like GNUPlot[13] or S2plot (Barnes et al., 2006), a 3-D plotting library written in C. These have been widely used by the astronomy community due to their general-purpose plotting features and flexibility. Fluke et al. (2009) demonstrated the use of s2plot to create interactive 3-D web visualisation for astronomy and later showed the potential creating VR visualisation as well, albeit very rudimentary (Fluke and Barnes, 2018).

Emerging platforms like glue[14] and OpenSpace[15], with support for intercompatibility and cooperative workflow between different programs and a highly interactive GUI, are becoming some of the most accessible tools for astro-visualisation. Faherty et al. (2020) presents the incredible application of such software with advanced display domes in planetariums to support scientific data analysis, education, and public outreach.

Broader computer graphics programs usage for astronomical visualisation intended for cinematic and public outreach purposes should also be mentioned. Naiman et al. (2017), Borkiewicz et al. (2019a), Borkiewicz et al. (2019b) used cinematic visual effects application Houdini to present astronomical data in a more visually interesting way for non-experts. However, as these programs focus on visual fidelity and computational speed, mesh-based techniques are used instead of particle-based.

Kent (2015) explained the advantages of a general-purpose 3-D modelling program in visualising for astronomical research such as a beginner-

[13]http://www.gnuplot.info/

[14]https://glueviz.org/

[15]https://www.openspaceproject.com/

friendly workflow with minimal effort to produce visually appealing results, with built-in GPU utilisation for fast rendering and features to help create animation or physics-enabled simulations in just a few clicks. In addition, it also provides extensive support for Python scripting, enabling possibilities for custom features.

One such example is AstroBlend (Naiman, 2016), a package created specifically for astronomical visualisation in Blender. Blender is slowly becoming more popular in astronomy and other sciences, but due to the requirement of some understanding of Blender to use, it appeals still has not reached the wider astronomy community.

Overall, the landscape of visualisation applications for astrophysics and astronomy is diverse and well-established with emerging innovative software still to come. It can be seen that public outreach and education is also a prominent focus. With massively collaborative projects like OpenSpace and *glue* proving a two-way relationship between science and public outreach can be achieved with interdisciplinary research and development to make new scientific discoveries[16]. Regarding particle-based data visualisation, there are numerous programs that allow for such a task from general-purpose to specialised with varying ranges of flexibility, functionalities and usability. All of which are limited to a 2-D projection of 3-D graphs. Applications of novel displays to combat this limitation in astronomical research exist, and will be discussed in Section 2.3.

2.2 Immersive analytics for scientific visualisation

This section aims to explore the benefits that immersive displays, including VR HMDs, can bring to scientific visualisation. Combining immersive environments and scientific data analysis is not a recent idea, Bryson (1996) expressed that scientific visualisation can greatly benefit from non-traditional displays if challenges such as unnatural interface, inadequate display technology, inefficient data management and other technical limitations can be addressed. Scientific analysis is mainly involved with high-dimensional data (e.g. physics and astrophysics simulations), where the output is a multi-dimensional data set representing a state of a 3-D system. Immersive displays can greatly benefit the visualisation of this data type by making it more comprehensive with the addition of depth perception (Bryson, 1996). Since then, there has been advancements in all of these areas and with VR HMDs becoming more affordable, such an idea had become feasible to realise (Kline and Volegov, 2021; Cordeil et al., 2017b; Nagao et al., 2016; Donalek et al., 2014a).

A new area of research concerning the use of immersive displays for data visualisation advancements is *immersive analytics* (Chandler et al., 2015). While immersive analytics is not limited to visualisation and VR HMDs (Chandler et al., 2015), there is a considerable amount of research that has shown success in the area. Ware and Mitchell (2008) found an improvement of graph comprehension by using stereoscopic displays, which is the type of display used by, but not exclusive to, VR HMDs.

Ware and Franck (1996) found significant increase in the size of a 3-D graph a user can understand with stereopsis compared to traditional screens, however due to the technological limitation at the time with the data used being specifically designed for testing, a more recent investigation is required. Erra et al. (2018) found VR can have benefits for interacting with 3-D graphs but also showed that a great number of users experienced difficulty with navigation and control at first, as they had minimal experience with VR headsets and controllers.

Studies that focus on investigating a better user interface with more natural and intuitive interactions aim to address this roadblock and have shown success (Yang et al., 2021; Drogemuller et al., 2018; Besançon et al., 2017). Wagner Filho et al. (2018) found an advantage of HMDs is larger subjective perception of accuracy and engagement while users spent less effort to navigate and extract information using HMDs. A study by Besançon et al. (2017) is more closely related to this project's topic of interest which investigated novel interaction methods with fluid dynamics data sets. The research found that domain experts preferred the prototype over traditional input methods. However, the project was focused on 3-D hybrid interfaces rather than HMDs (the same interface can also be implemented in VR). The findings further proved that with a well-designed interface, proper 3-D interactions with data can be beneficial for scientists.

There are multiple existing open-source and commercial programs for VR/AR data visualisation tasks. Programs such as Virtualitics[17], iViz

[16] VIS2020 Talk on AstroVis: https://youtu.be/nB36qPnfoGc

[17] https://virtualitics.com/

(Donalek et al., 2014b), Flow Immersive[18] (DiBenigno et al., 2021), 3Data[19], BadVR [20], LookVR[21], Nanome[22] and KineViz[23] are full-fledged and well-maintained industry software that offer general-purpose data visualisation (except for Nanome, which is aimed at chemists and structural biologists). However, as they are commercialised, they do not offer a great degree of customisation and custom extension specific to our purpose. Whereas, the current project aims to provide a foundational open-source package that not only can be used for scientific research out of the box but also provides future customisation with minimal effort to meet the needs of specific astronomical research projects.

The aforementioned popular ParaView and its core, VTK, offer support for VR visualisation with OpenVR standards. The toolkit has been used successfully for engineering and medical visualisation, but there has been little support for particle-based rendering. Furthermore, the functionality is still in its infancy and requires more improvements to be made (O'Leary et al., 2017). The development of external plug-ins for VTK unfortunately requires in-depth knowledge of the toolkit and advanced software engineering experience, thus considered not suitable for this project.

Glance (Filonik et al., 2016) is a proposed framework and architecture to enable optimised information visualisation utilising the GPU. However, the project has only been seen as a proof-of-concept rather than a concrete toolkit and thus requires extensive knowledge of low-level programming to work with and implement new features.

VRIA (Butcher et al., 2018) is a web-focused platform for VR visualisation that aims to provide multi-platform support using web standards. The program also includes VRIA Builder which simplifies the work required to create new visualisations along with API to support further extension by the community. Similarly, VR-Viz (Saifee, 2018) utilises A-Frame, a WebVR framework, to achieve the same task. Due to the limitation of the target platform being web-based, at the moment it is not suitable for the type of visualisation of interest - which requires scalability to support the large size of astronomical data sets, especially particle-based ones.

[18]https://flowimmersive.com/
[19]https://3data.io/
[20]https://badvr.com/
[21]https://looker.com/
[22]https://nanome.ai/
[23]https://www.kineviz.com/

ImmVis (Pedroso and Costa, 2021) is a new package that provides data analysis features for VR visualisation by connecting Python data analytics tools (e.g. Numpy, scikit-learn, Pandas) to work with existing immersive analytics packages. It is not a visualisation platform but can potentially be of help when used with the outcome of this project to provide data analysis capabilities to VR visualisation.

The Blender Data Visualisation Plugin[24] provides data visualisation capabilities to Blender by providing simple-to-use preprogrammed user interface and excellent visual fidelity. However, the VR support and development workflow for the add-on is not straightforward and requires further learning and development from the users.

DxR (Sicat et al., 2018) is a promising toolkit aimed at providing authoring capability to VR/AR visualisation in Unity. One key strength of DxR is the focus of development to support users of varying level of programming skills. Modern scientific researchers are becoming more trained in programming and computing proficiency, although certainly not to the extent of professional computational scientists or programmers, especially non-computing science students. Having a VR visualisation tool that is accessible to this audience is highly desirable to support scientific discoveries. DxR provides simple-to-use GUI to create visualisations and also API for further community support, which allows flexibility and accessibility to programmers and nonprogrammers alike. However, it is clearly stated that the focus of DxR is on rapid prototyping and does not scale well with larger data sets, which presents an issue for astrophysics and astronomy.

The Immersive Analyics Toolkit (IATK) (Cordeil et al., 2019) is also an Unity package that is very similar to DxR but aims to be a more complete and cohesive user experience with the aforementioned benefits of DxR like low barrier of entry, support for API, more expressive grammar than other existing toolkits, such as support for fluid interaction with data similar to ImAxes (Cordeil et al., 2017a), and most importantly scalability to support larger data sets.

From this overview, IATK is found to be a suitable foundation to begin development of an astronomy-focused extension. The extension should provide the benefits of immersive analytics to support new scientific discoveries and analysis of the ever-growing data in astronomy and astro-

[24]https://github.com/Griperis/BlenderDataVis

physics research. More details of how IATK and Unity benefit the realisation of the vision of the project will be discussed in Section 4.1.

2.3 Use of VR in astronomical visualisation

In this section, previous explorations of nontraditional visualisation applications in astronomy and astrophysics will be discussed with the goal of determining whether an endeavour to achieve the project's goal have been attempted.

A basic application of VR for astrophysics visualisation is 360-degree video, which allows the use of platforms with panorama viewing support such as YouTube and Facebook. Russell (2017) created the first 360-degree visualisation of an astrophysics simulation by using SPLASH to generate not a fixed-angle view of the SPH simulation but rather, a full 360 degrees render around the camera position. Kent (2017) described the process to produce a 360-degree image of astrophysical data using Blender instead, which provides the advantage of flexible configuration of rendering and data representation of a 3-D modelling program. A side effect is that it requires more work to ensure scientific accuracy. The visualisation of a black hole from the movie Interstellar, constructed under the consultation of Kip Thorne, was a technical marvel for the movie industry. Verbraeck and Eisemann (2021) presented an intricate attempt at replicating such a feat but with more focus on shorter rendering time by using grid-based interpolation techniques to render the results. These projects present only a fixed-camera view of the visualisation with limited interactivity.

The E0102-VR project (Baracaglia and Vogt, 2020) showed observational astrophysics can benefit from a properly developed toolkit for VR visualisation, but its output only serves as a proof of concept and is not a ready to use platform. The project by Arcand et al. (2018) presents the Cassiopeia A supernova remnant by using observational data collected from observatories and telescopes to construct a 3-D model that can be ported to VR and AR environments using VTK and MinVR, allowing users to *walk* inside the structure. The project did not provide a cohesive and convenient workflow to produce future visualisation and also do not support direct data import. This results in additional effort required to prepare the data to be used in the VR environment. The 3D-MAP VR project (Orlando et al., 2019) is a well-known attempt of VR visualisation for astrophysics yet its workflow of requiring multiple programs to generate a visualisation made it less suitable for the current goal. Ferrand et al. (2016) created a Unity project to experiment with the potential of VR for astronomy, but it is not a visualisation package.

With the hardware becoming more accessible and the software more developed, scientists are starting to apply VR to visualisation of astronomical data. In general, there are few attempts of VR visualisation for astronomy and astrophysics. Existing projects are limited in terms of capability and have different goals, therefore the development of a dedicated VR visualisation toolkit for particle-based data with functionalities important for astronomical and astrophysical research is needed.

3 Summary of the State of the Art

The review shows that successful application of VR to astrophysics and astronomy visualisation is highly desirable. Yet, no current project exists that satisfies the criteria for a reliable, extendable, user-friendly VR visualisation toolkit for particle-based data sets with a cohesive and straightforward workflow[25] that is accessible to both experts and non-experts alike.

It is standard for general-purpose visualisation tools (VTK, ParaView, VisIt, yt, 3D Slicer, etc.) to be used for astronomy and astrophysics visualisations, although they are not specialised for particle-based data and some omit the support for particle-based data entirely. They also lack a built-in capability to create VR visualisations, except for VTK and ParaView, but these only offer minimal functionality for 3-D interaction. More specialised tools for particle-based data like Splotch, SPLASH, or Partiview offer helpful features but also do not provide support for VR visualisation.

Dedicated VR visualisation tools are better suited for the project's purpose. To the author's knowledge, there is no existing commercial VR visualisation tool specialised for astrophysics and astronomy (similar to how Nanome is specialised for chemistry and biology). As a result, the best solution is to extend an existing open-source VR visualisation project like VTK, Glance, VRIA, VR-Viz, DxR, or IATK. As discussed, VTK requires strong software development skills and experience with the framework, Glance requires ex-

[25] Here, *workflow* refers the process starting from the importing data to displaying the finalised visualisation. It does not include creation/wrangling/processing of the data.

tensive low-level programming, VRIA and VR-Viz are web-based and not scalable, while DxR is close to the requirements but also lacks scalability.

From existing research in immersive analytics and scientific visualisation, the ability to interact in an immersive way with the data (such as walking through or using ones hands to interact with the data in an intuitive way) to explore and understand the visualisation can be extremely beneficial. This is more important for amateur researchers or students whom have yet developed strong intuition for complex and large scale data sets. A great amount of scientific discoveries are made not just by experts but also students and apprentices, this further motivates the development of such functionalities to fully bridge the gap between VR visualisation and astronimical sciences. The closest existing software package is IATK, which meets almost all of the criteria except for specialised astrophysics and astronomy functionalities for particle-based data visualisation. The goal is made possible by its API support and Unity's low barrier of entry for development. By developing a simple, yet specialised, add-on for IATK, the project hopes to realise this vision and motivate the mainstream use of VR for astronomical and astrophysical research and education.

4 Research Project Plan

This section will touch on the software and algorithms that will be used as the foundation for the development of the project, the identified challenges and missing features that need to be addressed, as well as the proposed timeline for the project.

4.1 Unity and IATK as a foundation

Unity[26] is a 3-D game engine that provides all the necessary functionalities to streamline the process of developing an interactive 3-D experience, making it accessible for even a complete beginner to create a working product. Unity has recently been considered as a standard for VR and AR interaction development due to its ease of use (making 3-D environment usable in VR requires only a few clicks[27]), and extensive VR/AR support from the developers.

IATK provides a Scatterplot visualisation, which renders tabular data in 3-D with the ability to select data columns for the axes using the GUI, thus enable successful primitive rendering of particle-based data out of the box within minutes. The IATK GUI utilises the Unity *Inspector*, which enables seamless integration between the high-level user interface and the underlying code, with minimal programming required. This allows the visualisation workflow to be friendly with nonprogrammers. There is also a strong focus in development of IATK to support low-level APIs making it straightforward to implement new features for specific visualisations, such as astrophysics simulations, rendering of SPH data or spectral cubes visualisation. One of IATK's core design requirements is *scalability*, which is important for astronomical data sets. IATK has already provided a GPU-utilised rendering pipeline capable of handling millions of data points, which significantly reduces the amount of development time required for the computer graphics component of the project. However, the package still lacks key features for astrophysics visualisation of particle-based data. These will be discussed in detail in Section 4.3.

4.2 SPLASH algorithm for SPH 3-D surface rendering

One of the core objectives for the project is to enable SPH visualisation in VR. This subsection will summarise the main techniques used in SPLASH (Price, 2007) for 3-D surface rendering of the particle data that will be re-implemented in IATK. Please see the original text (specifically Section 4.3) for an in-depth explanation of the procedures and algorithms.

From a high level understanding, the 3-D surface in SPLASH is rendered using a process much like *splatting* in volume rendering, in which the particles are sorted in terms of distance away from the observer and are splatted on to the projection surface (the screen) with the appearance of the particles determined by the numerical solution for the 2-D interpolation of the SPH 3-D summation interpolant instead of a Gaussian kernel. This solution is given by equation 32 in the SPLASH paper:

$$\mathcal{A}(x,y) = \sum_j w_j h_j A_j F(\frac{r_{xy}}{h})$$

The individual variables and functions of the equation are explained in detail in the source paper. For simplicity, this results in a diametric Gaussian-like distribution, meaning a particle's pixels will become more transparent the further away they are from the particle's center.

[26]https://unity.com/

[27]https://docs.unity3d.com/540/Documentation/Manual/VROverview.html

SPLASH determines the final color of a pixel using ray tracing which takes each particle's intensity into calculation for the final intensity value of the pixel. For each particle at the pixel location (x, y), its contribution is determined by the intensity $I(x, y)$ which is the sum of the particle's intensity absorption and emission (equation 45 from the source paper):

$$I(x,y) = I_0(x,y)e^{-\tau_i(x,y)} + S_i(1 - e^{-\tau_i(x,y)})$$

To simplify, this can be understood as:

$$I(x,y) = I + S$$

With I being the absorption term and S being the emission term. In optics, the absorption of a particle's intensity over a distance D in a medium is usually of the form:

$$I = I_0 e^{-\kappa\rho D} = I_0 e^{-\tau} = \frac{I_0}{e^{\tau}}$$

Where I_0 is the original intensity of the source particle, κ is the *opacity* (or *absorbtion coefficient*) of the medium, ρ is the *density* of the medium, and τ is the *optical depth*. The expression given in the above equation is for ease of understanding. In practise, the calculation for τ will be the one implemented in SPLASH, given by equation 46 :

$$\tau_i(x,y) = \kappa m_i Y(x - x_i, y - y_i, h)$$

The emission term S relies on the source function S_i, of which the paper mentioned two variations: assigning RGB color values to each particle or keeping a monochromatic intensity for each particle. For this project, the monochromatic intensity is used and therefore S_i for each particle is the rendered quantity at the particle location. The intensity can then be mapped to a custom color scale.

By porting the developed numberical solutions and rendering procedure created by Price (2007) to IATK, surface rendering of SPH data in VR can be achieved with minimal development time.

4.3 Challenges, missing features, and proposed solutions

With the decision to use the existing IATK, most of the software development needed to realise the goal had already been done. However, there are major features and challenges key to an astro-visualisation program that need to be implemented and addressed.

First, a binary data file parser that supports loading large data files is needed. Currently, IATK comes with a CSV parser, which is not widespread for astro-data. While it is certainly possible to use an external script to convert the binary data to a CSV, it defeats the purpose of having a cohesive workflow for VR visualisation. Furthermore, astronomical data files are usually in orders of gigabytes (GBs), it is not feasible to load the entire table into memory at once and an alternative solution is needed. A possible solution can be to first load the header columns and then allow the user to select the columns to load the data for, which can significantly reduce the amount of redundant data being read.

Second, as the data can have very extreme dynamic range, the current IATK linear mapping of color gradients is not suitable. An adaptive custom mapping function or a dynamic scale needs to be implemented to effectively color the data.

Third, IATK currently renders the data as discrete points using a Scatterplot object, which is not useful for data like SPH simulations and therefore a method for surface rendering is also needed. This can be achieved by first quickly implementing each point's sprite as simple 2-D Gaussian distributions to get started, then improving it by porting the SPH kernel mentioned in the previous subsection.

Another missing feature is to support plots of vector quantities for the particles. A naive solution is to simply render an arrow for each existing particle yet this is not computationally feasible for VR visualisation due to a strict requirement of at least 60 frames per second to avoid inducing motion sickness and dizziness to users (Zhang, 2020). Therefore, a function to selectively plot vectors is essential.

Finally, analysis of astrophysics simulations heavily relies on states at multiple time steps of the same system, thus requires the ability to seamlessly load and switch between multiple tables of data. This does not necessarily have to be done in real-time, but being able to select a time step to display with a delay in between is a desirable functionality.

4.4 Timeline

The project's roadmap has been agreed to be comprised of monthly *sprints*. With each major issue being addressed by a dedicated month. The proposed timeline is as follows:

Project Timeline

Aug 2021	Literature review
Sept 2021	Literature review submission & custom file loading
Oct 2021	Rudimentary particle splatting & adaptive color mapping function
Nov 2022	Surface rendering of SPH data & naive vector plots
Jan 2022	Begin thesis write-up
Feb 2022	Smarter vector plots
Mar 2022	Multiple tables loading for changing time steps
Apr 2022	Additional features (haptics, etc.) and optimisation
Jun 2022	Thesis submission

5 Conclusion

Research in astronomy and astrophysics requires effective visualisation software to handle the ever-increasing amount of data and make new scientific discoveries. One of the most common types of data sets is particle-based data. A vast collection of both general-purpose and specialised software is available for domain researchers to use for authoring and data analysis tasks. These standard programs lack the ability to visualise data interactively in Virtual Reality. The awareness of potential benefits of VR to scientific visualisation is not a new discovery, yet due to technical limitations, the topic had not been explored thoroughly by researchers until recently. A number of VR visualisation packages have been developed, but most do not fit the requirements of a competent particle-based visualisation tool for astronomical research. We aim to address this gap by utilising IATK to create an accessible and cohesive workflow to enable VR visualisations of particle-based astronomical data sets. By using an open-source platform, further extensions are possible and strongly encouraged with the hopes of making accessible VR visualisation standard for astronomical research in the future.

References

Abbott, B. P., Gawiser, E., Emmart, C. E., and Wyatt, R. J. (2003). Hayden Planetarium's Digital Universe: The Extragalactic Atlas. In *American Astronomical Society Meeting Abstracts*, volume 203 of *American Astronomical Society Meeting Abstracts*, page 118.07.

Arcand, K. K., Jiang, E., Price, S., Watzke, M., Sgouros, T., and Edmonds, P. (2018). Walking through an exploded star: Rendering supernova remnant cassiopeia a into virtual reality.

Baracaglia, E. and Vogt, F. P. A. (2020). E0102-VR: Exploring the scientific potential of Virtual Reality for observational astrophysics. *Astronomy and Computing*, 30:100352.

Barnes, D. G., Fluke, C. J., Bourke, P. D., and Parry, O. T. (2006). An Advanced, Three-Dimensional Plotting Library for Astronomy. *Publications of the Astronomical Society of Australia*, 23(2):82–93.

Becciani, U., Comparato, M., Costa, A., Gheller, C., Larsson, B., Pasian, F., and Smareglia, R. (2006). Visivo: an interoperable visualisation tool for virtual observatory data. *Proceedings of the International Astronomical Union*, 2(14):622–622.

Becciani, U., Comparato, M., and Gheller, C. (2006). VisIVO: a VO enabled tool for Scientific Visualization and Data Analysis . *Memorie della Societa Astronomica Italiana Supplementi*, 9:427.

Besançon, L., Issartel, P., Ammi, M., and Isenberg, T. (2017). Hybrid tactile/tangible interaction for 3d data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):881–890.

Borkiewicz, K., Christensen, A., Kostis, H.-N., Shirah, G., and Wyatt, R. (2019a). Cinematic scientific visualization: The art of communicating science. In *ACM SIGGRAPH 2019 Courses*, SIGGRAPH '19, New York, NY, USA. Association for Computing Machinery.

Borkiewicz, K., Naiman, J. P., and Lai, H. (2019b). Cinematic visualization of multiresolution data: Ytini for adaptive mesh refinement in houdini. *The Astronomical Journal*, 158(1):10.

Borkin, M., Goodman, A., Halle, M., and Alan, D. (2007). Application of Medical Imaging Software to 3D Visualization of Astronomical Data. In Shaw, R. A., Hill, F., and Bell, D. J., editors, *Astronomical Data Analysis Software and Systems XVI*, volume 376 of *Astronomical Society of the Pacific Conference Series*, page 621.

Brown, A. G. A., Vallenari, A., Prusti, T., de Bruijne, J. H. J., Babusiaux, C., Biermann, M., Creevey, O. L., Evans, D. W., Eyer, L., and et al. (2021). Gaia early data release 3. *Astronomy & Astrophysics*, 650:C3.

Bryan, G. L., Norman, M. L., O'Shea, B. W., Abel, T., Wise, J. H., Turk, M. J., Reynolds, D. R., Collins, D. C., Wang, P., Skillman, S. W., Smith, B., Harkness, R. P., Bordner, J., hoon Kim, J., Kuhlen, M., Xu, H., Goldbaum, N., Hummels, C., Kritsuk, A. G., Tasker, E., Skory, S., Simpson, C. M., Hahn, O., Oishi, J. S., So, G. C., Zhao, F., Cen, R., and and, Y. L. (2014). ENZO: AN ADAPTIVE MESH REFINEMENT CODE FOR ASTROPHYSICS. *The Astrophysical Journal Supplement Series*, 211(2):19.

Bryson, S. (1996). Virtual reality in scientific visualization. *Commun. ACM*, 39(5):62–71.

Butcher, P., John, N. W., and Ritsos, P. D. (2018). Towards a framework for immersive analytics on the web. In *Posters of the IEEE Conference on Visualization (IEEE VIS 2018), Berlin, Germany.*

Chandler, T., Cordeil, M., Czauderna, T., Dwyer, T., Glowacki, J., Goncu, C., Klapperstueck, M., Klein, K., Marriott, K., Schreiber, F., and Wilson, E. (2015). Immersive analytics. In *2015 Big Data Visual Analytics (BDVA)*, pages 1–8.

Clarke, C. and Carswell, B. (2007). *Principles of astrophysical fluid dynamics.* Cambridge University Press.

Cordeil, M., Cunningham, A., Bach, B., Hurter, C., Thomas, B. H., Marriott, K., and Dwyer, T. (2019). Iatk: An immersive analytics toolkit. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 200–209.

Cordeil, M., Cunningham, A., Dwyer, T., Thomas, B. H., and Marriott, K. (2017a). Imaxes: Immersive axes as embodied affordances for interactive multivariate data visualisation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, page 71–83, New York, NY, USA. Association for Computing Machinery.

Cordeil, M., Dwyer, T., Klein, K., Laha, B., Marriott, K., and Thomas, B. H. (2017b). Immersive collaborative analysis of network connectivity: Cave-style or head-mounted display? *IEEE Transactions on Visualization and Computer Graphics*, 23(1):441–450.

DiBenigno, M., Kosa, M., and Johnson-Glenberg, M. C. (2021). Flow immersive: A multiuser, multidimensional, multiplatform interactive covid-19 data visualization tool. *Frontiers in Psychology*, 12:1782.

Dolag, K., Reinecke, M., Gheller, C., and Imboden, S. (2008). Splotch: visualizing cosmological simulations. *New Journal of Physics*, 10(12):125006.

Donalek, C., Djorgovski, S. G., Cioc, A., Wang, A., Zhang, J., Lawler, E., Yeh, S., Mahabal, A., Graham, M., Drake, A., Davidoff, S., Norris, J. S., and Longo, G. (2014a). Immersive and collaborative data visualization using virtual reality platforms. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 609–614.

Donalek, C., Djorgovski, S. G., Cioc, A., Wang, A., Zhang, J., Lawler, E., Yeh, S., Mahabal, A., Graham, M., Drake, A., Davidoff, S., Norris, J. S., and Longo, G. (2014b). Immersive and collaborative data visualization using virtual reality platforms. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 609–614.

Drogemuller, A., Cunningham, A., Walsh, J., Cordeil, M., Ross, W., and Thomas, B. (2018). Evaluating navigation techniques for 3d graph visualizations in virtual reality. In *2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA)*, pages 1–10.

Erra, U., Malandrino, D., and Pepe, L. (2018). Virtual reality interfaces for interacting with three-dimensional graphs. *International Journal of Human–Computer Interaction*, 35:75 – 88.

Faherty, J. K., Desir, D., Coker, K., Nelson, O., Vasquez, C., and Smithka, I. (2018). Visualizing Gaia Data with Science Teachers at AMNH. In *American Astronomical Society Meeting Abstracts #231*, volume 231 of *American Astronomical Society Meeting Abstracts*, page 156.01.

Faherty, J. K., SubbaRao, M., Wyatt, R., Ynnerman, A., deGrasse Tyson, N., Geller, A., Weber, M., Rosenfield, P., Steffen, W., Stoeckle, G., Weiskopf, D., Magnor, M., Williams, P. K. G., Abbott, B., Marchetti, L., Jarrrett, T., Fay, J., Peek, J., Graur, O., Durrell, P., Homeier, D., Preston, H., Müller, T., Vos, J. M., Brown, D., Godfrey, P. G., Rice, E., Gagliuffi, D. B., Bock, A., Hedberg, J., Rosen, D., and Emmart, C. (2020). Ideas: Immersive dome experiences for accelerating science.

Fairall, A. (2005a). Seeing Beyond the Naked Eye in a Planetarium. In *In IAU Commission 55: Communicating Astronomy with the Public 2005*, pages 118–127.

Fairall, A. (2005b). Seeing Beyond the Naked Eye in a Planetarium. In *In IAU Commission 55: Communicating Astronomy with the Public 2005*, pages 118–127.

Ferrand, G., English, J., and Irani, P. (2016). 3d visualization of astronomy data cubes using immersive displays.

Filonik, D., Bednarz, T., Rittenbruch, M., and Foth, M. (2016). Glance: Generalized geometric primitives and transformations for information visualization in ar/vr environments. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*, pages 461–468.

Fluke, C. J. and Barnes, D. G. (2018). Immersive Virtual Reality Experiences for All-Sky Data. *Publications of the Astronomical Society of Australia*, 35:e026.

Fluke, C. J., Barnes, D. G., and Jones, N. T. (2009). Interchanging Interactive 3D Graphics for Astronomy[1]. *Publications of the Astronomical Society of Australia*, 26(1):64–74.

Garg, S. and Pant, M. (2017). Meshfree methods: A comprehensive review of applications. *International Journal of Computational Methods*, 15:1830001.

Hassan, A. and Fluke, C. J. (2011). Scientific visualization in astronomy: Towards the petascale astronomy era. *Publications of the Astronomical Society of Australia*, 28(2):150–170.

Jensen, M. B., Jacobsen, E. I., Frisvad, J. R., and Bærentzen, J. A. (2021). Tools for Virtual Reality Visualization of Highly Detailed Meshes. In Gillmann, C., Krone, M., Reina, G., and Wischgoll, T., editors, *VisGap - The Gap between Visualization Research and Visualization Software*. The Eurographics Association.

Jin, Z., Krokos, M., Rivi, M., Gheller, C., Dolag, K., and Reinecke, M. (2010). High-performance astrophysical visualization using splotch. *Procedia Computer Science*, 1(1):1775–1784. ICCS 2010.

Joye, W. A. and Mandel, E. (2003). New Features of SAOImage DS9. In Payne, H. E., Jedrzejewski, R. I., and Hook, R. N., editors, *Astronomical Data Analysis Software and Systems XII*, volume 295 of *Astronomical Society of the Pacific Conference Series*, page 489.

Kapferer, W. and Riser, T. (2008). Visualization needs and techniques for astrophysical simulations. *New Journal of Physics*, 10(12):125008.

Kent, B. R. (2013). Visualizing scientific graphics for astronomy and astrophysics. In *ACM SIGGRAPH 2013 Posters*, SIGGRAPH '13, New York, NY, USA. Association for Computing Machinery.

Kent, B. R. (2015). 3D Visualization of Astronomical Data with Blender. In Taylor, A. R. and Rosolowsky, E., editors, *Astronomical Data Analysis Software an Systems XXIV (ADASS XXIV)*, volume 495 of *Astronomical Society of the Pacific Conference Series*, page 157.

Kent, B. R. (2017). Spherical panoramas for astrophysical data visualization. *Publications of the Astronomical Society of the Pacific*, 129(975):058004.

Kline, J. L. and Volegov, P. L. (2021). Toward 3d data visualization using virtual reality tools. *Review of Scientific Instruments*, 92(3).

Lan, F., Young, M., Anderson, L., Ynnerman, A., Bock, A., Borkin, M. A., Forbes, A. G., Kollmeier, J. A., and Wang, B. (2021). Visualization in astrophysics: Developing new methods, discovering our universe, and educating the earth.

Levy, S. (2003). Interactive 3-D visualization of particle systems with Partiview. In Makino, J. and Hut, P., editors, *Astrophysical Supercomputing using Particle Simulations*, volume 208, page 343.

Levy, S. (2010). partiview: Immersive 4D Interactive Visualization of Large-Scale Simulations.

Lipşa, D. R., Laramee, R. S., Cox, S. J., Roberts, J. C., Walker, R., Borkin, M. A., and Pfister, H. (2012). Visualization for the physical sciences. *Comput. Graph. Forum*, 31(8):2317–2347.

Makino, J., Fukushige, T., Koga, M., and Namura, K. (2003). GRAPE-6: Massively-Parallel Special-Purpose Computer for Astrophysical Particle Simulations. *Publications of the Astronomical Society of Japan*, 55(6):1163–1187.

Monaghan, J. J. (1992). Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574.

Nagao, K., Ye, Y., Wang, C., Fujishiro, I., and Ma, K.-L. (2016). Enabling interactive scientific data visualization and analysis with see-through hmds and a large tiled display. In *2016 Workshop on Immersive Analytics (IA)*, pages 1–6.

Naiman, J. P. (2016). AstroBlend: An astrophysical visualization package for Blender. *Astronomy and Computing*, 15:50–60.

Naiman, J. P., Borkiewicz, K., and Christensen, A. J. (2017). Houdini for astrophysical visualization. *Publications of the Astronomical Society of the Pacific*, 129(975):058008.

Navratil, P., Johnson, J., and Bromm, V. (2007). Visualization of cosmological particle-based datasets. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1712–1718.

O'Leary, P., Jhaveri, S., Chaudhary, A., Sherman, W., Martin, K., Lonie, D., Whiting, E., Money, J., and McKenzie, S. (2017). Enhancements to vtk enabling scientific visualization in immersive environments. In *2017 IEEE Virtual Reality (VR)*, pages 186–194.

Orlando, S., Pillitteri, I., Bocchino, F., Daricello, L., and Leonardi, L. (2019). 3dmap-VR, a project to visualize three-dimensional models of astrophysical phenomena in virtual reality. *Research Notes of the AAS*, 3(11):176.

Pedroso, F. and Costa, P. (2021). Immvis: Bridging data analytics and immersive visualisation. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 1: IVAPP,*, pages 181–187. INSTICC, SciTePress.

Pössel, M. (2020). A Beginner's Guide to Working with Astronomical Data. *The Open Journal of Astrophysics*, 3(1):2.

Price, D. J. (2007). Splash: An interactive visualisation tool for smoothed particle hydrodynamics simulations. *Publications of the Astronomical Society of Australia*, 24(3):159–173.

Russell, C. M. P. (2017). 360-degree videos: a new visualization technique for astrophysical simulations. In Eldridge, J. J., Bray, J. C., McClelland, L. A. S., and Xiao, L., editors, *The Lives and Death-Throes of Massive Stars*, volume 329, pages 366–368.

Saifee, M. (2018). VR-Viz: Visualization system for data visualization in VR. Master's thesis, Aalto University. School of Arts, Design and Architecture.

Sicat, R., Li, J., Choi, J., Cordeil, M., Jeong, W.-K., Bach, B., and Pfister, H. (2018). Dxr: A toolkit for building immersive data visualizations. *IEEE transactions on visualization and computer graphics*, 25(1):715–725.

Sua, S., Chaudhary, A., O'Leary, P., Geveci, B., Sherman, W., Nieto, H., and Francisco-Revilla, L. (2015). Virtual reality enabled scientific visualization workflow. In *2015 IEEE 1st workshop on Everyday Virtual Reality (WEVR)*, pages 29–32. IEEE.

Taylor, M. (2017). Topcat: Desktop exploration of tabular data for astronomy and beyond. *Informatics*, 4(3):18.

Taylor, M. B. (2019). TOPCAT and Gaia. In Teuben, P. J., Pound, M. W., Thomas, B. A., and Warner, E. M., editors, *Astronomical Data Analysis Software and Systems XXVII*, volume 523 of *Astronomical Society of the Pacific Conference Series*, page 43.

Turk, M. J., Smith, B. D., Oishi, J. S., Skory, S., Skillman, S. W., Abel, T., and Norman, M. L. (2010). yt: A MULTI-CODE ANALYSIS TOOLKIT FOR ASTROPHYSICAL SIMULATION DATA. *The Astrophysical Journal Supplement Series*, 192(1):9.

Verbraeck, A. and Eisemann, E. (2021). Interactive black-hole visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):796–805.

Wagner Filho, J. A., Rey, M. F., Freitas, C. M. D. S., and Nedel, L. (2018). Immersive visualization of abstract information: An evaluation on dimensionally-reduced data scatterplots. In

2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pages 483–490.

Ware, C. and Franck, G. (1996). Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Trans. Graph.*, 15(2):121–140.

Ware, C. and Mitchell, P. (2008). Visualizing graphs in three dimensions. *ACM Trans. Appl. Percept.*, 5(1).

Woodring, J., Heitmann, K., Ahrens, J., Fasel, P., Hsu, C.-H., Habib, S., and Pope, A. (2011). Analyzing and Visualizing Cosmological Simulations with ParaView. *The Astrophysical Journal Supplement*, 195(1):11.

Yang, Y., Cordeil, M., Beyer, J., Dwyer, T., Marriott, K., and Pfister, H. (2021). Embodied navigation in immersive abstract data visualization: Is overview+detail or zooming better for 3d scatterplots? *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1214–1224.

Zhang, C. (2020). Investigation on motion sickness in virtual reality environment from the perspective of user experience. In *2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pages 393–396.

Final Thesis Paper

An Exploration of Immersive Smoothed-Particle Hydrodynamics Data Visualisation in Astrophysics and Astronomy

Nhan H.T. Luong
Monash University

Daniel J. Price
Monash University

Barrett Ens
Monash University

Maxime Cordeil
The University of Queensland

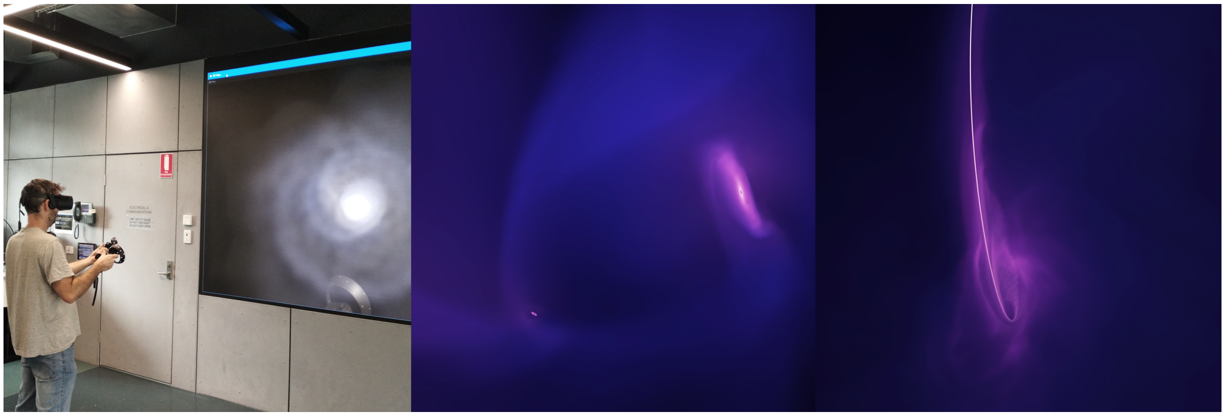

Figure 1: Left: User study using VR prototype to explore the common-envelopes data set [27]. Middle: The stellar flyby data set [6] rendered using the VR prototype. Right: A tidal disruption event simulation (Liptai et al., to be published) rendered using the VR prototype.

Abstract

Researchers have long suggested that novel display technologies can greatly benefit scientific analysis, and research in this area has recently gotten a lot of attention thanks to the widespread availability of affordable virtual reality (VR) headsets. Visualisations of simulation data are extensively used in astronomy and astrophysics research, which motivated us to look into the viability of using VR to visualise smoothed-particle hydrodynamics data, a particle-based fluid modelling technique. We implemented two rendering techniques: grid-based volume rendering and particle billboards. The technical challenges encountered, the decisions made and the overall performance were described in detail. A user study was conducted to obtain feedback from researchers and students in this area to better understand the design requirements of a functional visualisation tool for SPH research. The results of the user study were overwhelmingly positive and the users were able to understand the 3-D structure of the data much better than with traditional software. The results showed that in the future, the VR visualisation tool can play an important role in their work if some technical challenges can be solved and more quantitative analysis features are added. There are still barriers to overcome before such software can be implemented, but we have found that there is a strong motivation for further work to address the remaining problems and create a widely useable SPH VR visualisation tool.

Index Terms: Human-centered computing—Visualization—Visualization application domains—Scientific visualization; Computing methodologies—Computer graphics—Graphics systems and interfaces—Virtual reality

1 Introduction

Effective visualisations are essential in astronomical research in order to analyse complex data and make innovative scientific discoveries. With the advent of low-cost virtual reality (VR) devices, research into how to use the technology to improve scientific data visualisation and visual analytics has received a lot of attention. The implementation of novel display technologies for visual data analysis tasks has previously been shown to provide empirical benefits. This inspired us to look at the possibility of using VR technology in astronomy and astrophysics simulations, specifically a fluid simulation method called smoothed-particle hydrodynamics.

Immersive analytics [9] introduces the research domain of applying immersive display technology for data visualisation and analytics. Although using immersive displays to aid scientific visualisation is not a novel concept [8], the issues of low quality interface design, limited display hardware technology, and computational capability have just recently been able to be resolved. It is well known that a high level of interactivity and immersion of novel displays is beneficial for scientific visualisation [30], with multiple studies showing performance improvement in data analysis tasks using immersive environments [15] [17] [47] [50].

Simulation of gases is at the core of astronomical research. These fluid simulations are often done with either a Eulerian mesh-based [26] or a Lagrangian mesh-free approach. Each paradigm has its own merits and is often chosen based on the goal of a project [2] [5]. Smoothed-particle hydrodynamics (SPH) [31] [32] is a widely-used mesh-free Lagrangian method for its capability to simulate highly dynamic fluids with free surfaces. SPH is also commonly found in other domains such as planetary sciences, civil engineering, and computer graphics. Effective visualisation of SPH data is essential to the analysis of a physical system.

The simulations themselves are not often visualised in real-time in astronomical research. Rather, the main aspect of data analysis is visualisation of the data produced by SPH codes such as PHANTOM [39] or GADGET-2 [45]. There are a variety of tools for visualising this type of data, including general-purpose scientific plotting applications like TOPCAT[1], as well as more specialised SPH software like Splotch [13] and, most notably, SPLASH [36]. However, they are limited to traditional 2-D displays and none have been found that allow immersive visualisation of SPH data. As a result, the scope of this project is not to simulate fluids in real time, but to focus on visualisation of generated SPH data.

Recently, the Inter-university Institute for Data Intensive Astronomy[2] developed iDaVIE-v [10], a software to visualise astronomical data cubes in virtual reality. This further motivated the creation of a similar tool for SPH data, as iDaVIE-v can only visualise data cubes in FITS format, which are grid-based data. Along with iDaVIE-v, projects like glue[3] and OpenSpace[4] demonstrate that the astronomy community recognises the value of novel platforms and is beginning to develop tools to help domain specialists and the general public better understand astronomical data.

The project is intended to be an exploratory study to determine the feasibility, technological challenges, and potential benefits of employing VR head-mounted displays (HMD) to perform SPH data exploration, with the goal of laying the groundwork for future research. The following are necessary to address this:

- A 3-D SPH rendering system that can run on a VR HMD,
- A user study to explore potential of VR for SPH data analysis.

More specifically, the main questions to answer qualitatively through the user study are:

- Does VR visualisation improve user understanding of the 3-D structure of the data?
- Is haptic feedback helpful in exploring the data?
- What new features would make the VR application better suited for the user's work?

For our contribution, we designed and implemented two renderer prototypes, identified technical challenges and aspects that future work should be aware of, and described the process in detail for future work to be based upon and improve. We want the project to serve as a time saver for future work where more effort can be put into optimisation and further advancements in development. Additionally, we conducted a user study to gain experts' insights on potential of VR for SPH data analysis, what is essential in a tool for a researcher to perform SPH data exploration, and whether future efforts should be put towards creating such tool.

Section 2 will discuss previous works and their limitations or differences to the current project, with Section 2.1 focusing on traditional SPH visualisation and Section 2.2 dedicated to Immersive analytics software. Section 3 serves as a methodology discussion and will describe design decisions, observable variables, and justifications for the chosen methods. Section 4 introduces the SPH interpolation process, which is necessary to understand how it can be visualised correctly. Section 5 discusses the technical implementation of 3-D rendering methods for SPH data and VR interactions along with the performance evaluation of each method. Section 6 focuses on the details of the user study, including the general design in Section 6.1, the tasks involved in Section 6.2, the participant groups in Section 6.3 and finally the final results from the study in Section 6.4. Section 7 will contain general discussions of the results of the user study, technical feasibility, limitations, and suggestions for future improvements. Lastly, Section 8 concludes the work in general with a discussion of potential for future work.

[1]http://www.star.bristol.ac.uk/ mbt/topcat/

[2]https://www.idia.ac.za/

[3]https://glueviz.org/

[4]https://www.openspaceproject.com

2 Related work

In this section, prior work regarding SPH visualisation and immersive data visualisation will be discussed. It is important to gain an overview of the software landscape for SPH visualisation and identify the gap of VR support in existing SPH visualisation tools. Furthermore, research related to immersive analytics is explored to further motivate the need for the application of VR in SPH visualisation. Software made with the purpose of immersive data visualisation will also be examined to identify suitable existing work that can be built upon and missing components that need to be implemented.

2.1 SPH visualisation software

In astrophysics and astronomy, SPH simulations are commonly visualised using SPLASH [36]. It includes a large suite of scientific plotting capabilities and, most importantly, the ability to render 3-D SPH data accurately. Other programs that can also be used for SPH visualisation are ParaView [3], IFrIT [20] and Splotch [13]. ParaView's 3-D SPH interpolator is limited to a subvolume of the data, IFrIT offers limited rendering options for SPH data, and Splotch provides rendering capability similar to SPLASH, yet lacks interactivity. A more recent solution is to combine the SPH simulation code DualSPHysics [12] with the Blender 3D graphics programme for visualisation in a single workflow [19] but the rendering method is focused on isosurface visualisation, which is more useful for simulation of water, oil, or other liquids. In general, there is no VR-capable tool for SPH visualisation at the time of writing.

The most aligned with the project's purpose is iDaVIE-v [10] which provides scientific data analysis functionalities in VR for astronomical data cubes in FITS format. However, as this is a tool for discrete data, support for SPH visualisation isn't available.

2.2 Immersive data visualisation

The idea of enhancing scientific visualisation with novel display technologies has existed for nearly three decades [8]. The 3-D nature of VR displays has been proven to be extremely beneficial for scientific data analysis which is mostly involved with high-dimensional data [48]. This is even more apparent with physics and astrophysics simulations, where the high-dimensional data represent the states of a 3-D physical system over time. With the ever more affordable VR headsets, depth perception, haptic feedback, and intuitive user interactivity have become more accessible to developers. Many recent studies have shown positive improvements in data analysis tasks using VR [33] [23]. Most notably, the area of *immersive analytics* has been created to dedicate research efforts to the use of immersive displays to enhance data visualisation [9].

Both commercial and open-source programmes exist for data visualisation purposes. Virtualitics[5], FlowImmersive[6], 3Data[7] and many more are commercially available software with dedicated development teams with large sets of functionalities and well-designed user interfaces. However, none are suitable for SPH data and due to the commercial nature, cannot be modified. ParaView [3] and VTK have OpenVR support but limited capability for particle-based rendering. Combined with its complex architecture, the OpenVR API for VTK core is difficult to modify to accommodate the research goal.

The Immersive Analytics Toolkit [11] is an open-source framework specifically designed for data visualisation and analytics using VR. The current iteration of the framework only provides general information visualisation types thus a custom visualisation pipeline must be implemented to support SPH rendering. Fortunately, the framework is designed with API support for creation of custom

[5]https://virtualitics.com/

[6]https://flowimmersive.com/

[7]https://3data.io/

visualisation at its core, making it highly suitable for a baseline for development of a VR SPH visualisation prototype.

3 Project Aims

As discussed in previous sections, there are inherent benefits when visualising scientific data with 3-D displays. On traditional 2-D screens, so-called *three-dimensional visualisations* are rendered in virtual 3-D space but projected on to a 2-D viewing plane thus omitting an entire spatial dimension of information in the data. To overcome this, techniques are developed to encode the information of the third dimension using another visual variable like colour. This is not a limitation with 3-D displays, where the third dimension of information is available through depth perception. This allows for representation of data with even higher dimensionality by using other visual variables for the fourth or even fifth dimension of the data set.

To answer the question of feasibility, we decided to implement and evaluate the performance of two rendering methods for SPH: volume rendering and particle billboards. Volume rendering is a popular method for visualising 3-D volumetric data in scientific visualisation and has previously been used to render SPH data [41] [29]. Particle billboard [42] [46] technique was chosen for its similarity to the ray-traced *faithful* SPH rendering procedure [36] if implemented correctly, which will be discussed in detail in Section 4. To evaluate the feasibility of each method, the following criteria are chosen:

- Pre-processing time,
- Run-time performance (frametime or frames per second),
- Visual fidelity (information loss),
- Interactivity.

To evaluate the *usefulness* of the technology, we conducted a user study to answer the qualitative questions presented in Section 1. Although some quantitative questions will be present in the user study, most have been decided to be qualitative and open-ended. A questionnaire will be used to assess the usability of traditional visualisation software (SPLASH) and the novel VR prototype, identify the benefits and drawbacks of each platform, and learn what participants want to see in a functional SPH visualisation software, as well as a short informal interview with participants about their overall experience with the study.

The decisions were made based on the aim of the study, which is to focus on the targeted users, to understand what they find beneficial and what is not about the technology, and to decide its potential. We are not focusing on evaluating the specifics of the prototype shown. For the analysis of the user study results, open coding [22] is chosen because there is no established codebook for this specific scenario.

4 Brief background on SPH visualisation

This section is dedicated to summarising the SPH rendering process presented in SPLASH [36], with the purpose of providing the necessary background to the topic. All novel work will be discussed in Section 5. It is important to note that an SPH particle is not analogous to a certain type of physical particle of the fluid (such as an atom or a molecule) but is a volume of the fluid. As a result, in astrophysical simulations the particles' spatial sizes are not uniform. This poses a challenge for rendering optimisation. Few programmes have implemented faithful SPH rendering. SPLASH describes the procedure in detail and a simplified version is discussed below. The 3-D surface is rendered using a process similar to *particle splatting* [1] [25], in which the particles in 3-D space are sorted by distance from the observer and projected onto the 2-D output plane.

At any given point r in an SPH volume, the quantity A is given by taking the sum of the contribution of the particles r_j at position j having density ρ_j, mass m_j and smoothing length h_j:

$$A(r) = \sum_j m_j \frac{A_j}{\rho_j} W(r - r_j, h_j). \tag{1}$$

The function $W(r,h)$ is called the kernel function, which is defined as a cubic spline [31]. Note that A can be any property of the fluid particles like density, mass or temperature.

In SPLASH, the continuous medium can be interpolated by applying the cubic spline kernel function W over the discrete particles:

$$W(q) = \frac{1}{\pi} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3, & 0 \le q < 1; \\ \frac{1}{4}(2-q)^3, & 1 \le q < 2; \\ 0, & q \ge 2. \end{cases} \tag{2}$$

The following equations use a camera-aligned coordinate system, with the z-dimension representing the *line-of-sight*. A dimensionless 2-D kernel F is the integral of W along the line-of-sight with $q_z = \frac{z}{h}$ and $q^2 = q_{xy}^2 + q_z^2$:

$$F(q_{xy}) = \int_{-\sqrt{4-q_{xy}^2}}^{\sqrt{4-q_{xy}^2}} W(q)\,dq_z. \tag{3}$$

As a result, the screen-space particle quantity can is the *column integrated kernel* Y, computed by using F with $q_{xy} = \frac{r_{xy}}{h}$:

$$Y(r_{xy}, h) = \frac{1}{h^2} F(q_{xy}). \tag{4}$$

This results in the 2-D contribution of a particle having a diametric Gaussian-like distribution for its contribution value, with the standard deviation approximating h, the SPH smoothing length. The final contribution is then multiplied by the particle's opacity coefficient to determine the final alpha contribution of this pixel.

In short, Figure 2 demonstrates the visual intuition for how the kernel impacts the rendering of a particle. Here, the particle is isolated from others in the data set, and its opacity is set to be maximum (`1.0`). In practise, each particle's opacity is multiplied by an opacity coefficient.

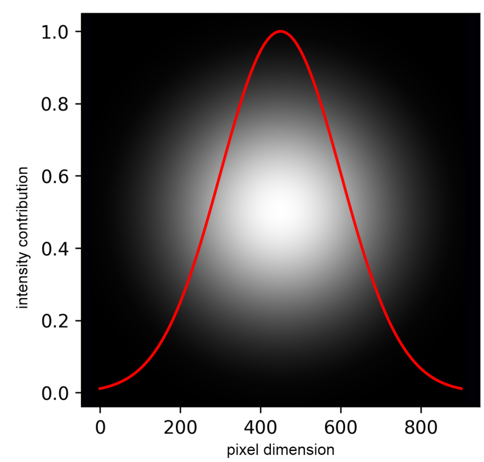

Figure 2: The rendered quantity for a particle using the Y kernel function is mapped to pixel intensity contribution.

Next, a ray-tracing procedure determines each pixel's intensity by accumulating particles' intensity absorptions and emissions at location (x,y):

$$I(x,y) = I_0(x,y)e^{-\tau_i(x,y)} + S_i(1 - e^{-\tau_i(x,y)}). \quad (5)$$

This is analogous to a simpler model with I being the *absorption* term and S being the *emission* term:

$$I(x,y) = I + S. \quad (6)$$

With I_0 being the original intensity of the source particle, a simplified model for particle's intensity absorption over distance D can be constructed using the *opacity* κ, the *medium density* ρ and the *optical depth* τ:

$$I = I_0 e^{-\kappa\rho D} = I_0 e^{-\tau} = \frac{I_0}{e^{\tau}}. \quad (7)$$

In practice, SPLASH calculates τ using:

$$\tau_i(x,y) = \kappa m_i Y(x - x_i, y - y_i, h). \quad (8)$$

A monochromatic output texture is constructed using the source function S_i and can then be mapped to a custom colour scale. Using this method, SPLASH allows the ability to visually encode particle properties such as *column density*, *temperature*, and *mass* using colour. This process can be seen in Figure 3, which shows a SPLASH visualisation of the Common-envelopes simulation coloured by log(column density) [27].

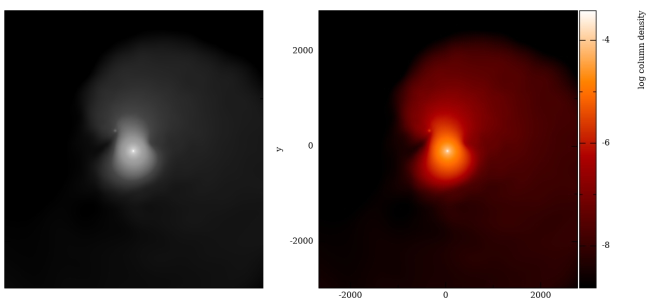

Figure 3: Left: Monochromatic pixel intensity texture generated using the ray-traced optical model. Right: Colour mapped final output.

This method produces outstanding results in terms of visual quality and perfect accuracy. However, it is too slow to be used for the VR environment, therefore a faster method is required. We will discuss our approach to overcome this hurdle in Section 5.

5 Immersive visualisation of SPH Data

This section aims to describe our technical contribution including what we implemented, the methods we designed, the challenges found, and what future projects should be aware of when creating such a system. Both prototypes discussed were made using the Unity engine, but the methods are discussed in a generalised way that can be implemented outside of the Unity system. This is because if one were to implement a specialised system from scratch without the constraints of a game engine, many aspects can be optimised and potentially yield much better results.

We implemented two primary rendering techniques, volume rendering and particle billboards, for SPH rendering. For each method, the core implementation idea is explained and the challenges, benefits, and performance are documented. The Unity3D[8] game engine is chosen as the primary development platform to utilise the built-in VR support which allows easy deployment to VR devices. The volume rendering prototype was built upon the `unity-volume-rendering`[9] package which necessitated a data sampling technique to spatially discretise the SPH data and populate the 3-D grid for a raycaster to visualise the volume quantity. The particle billboard prototype is a modification of the existing Immersive Analytics Toolkit [11] that already provides the essential functionalities like VR-capable scatterplot billboard rendering and tabular data ingestion pipeline with in-depth internal API support for custom visualisations. The billboard shader had been modified to simulate SPH interpolation and an API was created to accommdate the ingestion of SPH data. Overall it was found that the volume rendering method required long pre-processing time to adapt SPH data on to a 3-D grid, and for the desired framerate, the output is of nonsatisfactory visual quality. The particle billboard requires next to no precomputation time, however, suffers from poor performance scalability with data size and produced results with good visual quality, but only manages to barely achieve usable framerates on top-of-the-line hardware. According to the user study, adopting VR SPH visualisation in astronomical research has a lot of promise and clear benefits if the technical hurdles can be addressed in the future.

5.1 Design goals

In order for a prototype to be considered satisfactory for the user study, the following criteria must be met:

- Visual clarity: Minimal information loss, minute intricacies in the data set should still be visible in the visualisation.
- Reasonable pre-processing time: less than 1 minute for a data set of 1 million data points.
- Safe to use performance[10]: average frametime of at most 11 milliseconds or more than 90 frames per second.
- Interactivity: Users are able to freely navigate in 3-D space, pick up and rotate the data using their hands. Haptic feedback should also be implemented to encode an additional dimension of the data set.

Each method's viability and potential are also explored in terms of whether the requirements are met and, if not, whether they may be solved through better implementation or optimisation.

5.2 Implemented rendering methods for VR

5.2.1 Volume rendering

Volume rendering [14] is a common technique to visualise 3-D volumetric data. The volume is discretised into cubic volume elements; the quantity of volume elements used is the volume resolution. The technique usually involves using a ray-casting, ray marching, or marching cube algorithm to sample the volume and compute a colour value for the pixel output. It is common for the volume to be a uniform 3-D grid, however, it can be an irregular grid [49] for optimisations such as the octree grid [24] [43] [28]. It can be seen from past works that volume rendering is not uncommon to be used for SPH data rendering [41] [29], and specialised procedures have been developed for SPH [18] [21].

The `unity-volume-rendering`[11] package was chosen as a starting point. It provides direct volume rendering of medical imaging data using Unity's `Texture3D` data structure. The work described in this section is our own modifications applied to the package.

[8]https://unity.com/

[9]https://github.com/mattatz/unity-volume-rendering

[10]It is standard for VR experiences to be at least 90FPS to avoid invoking motion sickness. Sensitive or new users may still suffer from these side effects; however, it should be reduced to a safe degree if the performance standard is met.

[11]https://github.com/mattatz/unity-volume-rendering

A grid mapping method must be implemented in order for SPH data to be visualised using volume rendering. For this purpose, we designed a GPU-accelerated procedure for generating a `Texture3D` object using SPH data. To maintain the aspect ratio of the SPH volume within a cube, 2 *ghost particles*[12] at positions (M,M,M) and (N,N,N) are added to ensure the spatial size of the data is equal in all 3 dimensions. This produces a cube with side length $D = M - N$ in the original data set's unit for spatial dimensions where:

$$\begin{cases} M = max(max(\texttt{data}[\texttt{x}]), max(\texttt{data}[\texttt{y}]), max(\texttt{data}[\texttt{z}])); \\ N = min(min(\texttt{data}[\texttt{x}]), min(\texttt{data}[\texttt{y}]), min(\texttt{data}[\texttt{z}])). \end{cases} \tag{9}$$

A barebone reconstruction of the data using a 1-D Gaussian $G(d)$ with $\mu = 0$ and $\sigma = h_{cells}$ within a grid of resolution R^3 can be described using **Algorithm 1**.

Algorithm 1 Populating `Texture3D`

for each particle **do**
    `Volume` $\leftarrow$ `Texture3D`(R)
    p_x, p_y, p_z $\leftarrow$ particle's position in volume grid
    $k \leftarrow 2 * 3 * h_{cells}$
    for x from $p_x - k\backslash 2$ to $p_x + k\backslash 2$ **do**
        for y from $p_y - k\backslash 2$ to $p_y + k\backslash 2$ **do**
            for z from $p_z - k\backslash 2$ to $p_z + k\backslash 2$ **do**
                d $\leftarrow$ distance from (x,y,z) to (p_x,p_y,p_z)
                `Volume`$[x,y,z]$ $\leftarrow$ `Volume`$[x,y,z]$ + $G(d)$
            end for
        end for
    end for
end for

Here k and h_{cells} are in number of cells. For a particle with original smoothing length value h_0, we can compute h_{cells}:

$$h_{cells} = \frac{h_0 \cdot R}{D}. \tag{10}$$

Figure 4 shows the output of the procedure on the HD 97048 data set [34] with a 256 resolution volume cube.

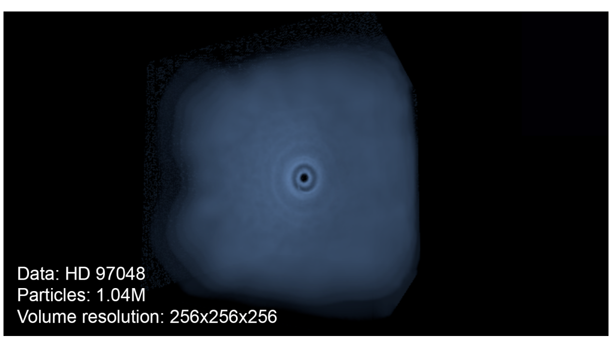

Figure 4: Volume rendering of HD 97048 data set [34] using the `unity-volume-rendering` prototype. Volume resolution: 256x256x256.

The procedure is parallelisable and is implemented using a compute shader. The current implementation is to parallelise the outer most loop, but a better method to parallelise all four loops at once is highly desirable. In practise, the spherical nature of the kernel is used to optimise the mapping procedure, by only performing the computations on an *octant*[13] in the cube and pasting the results into the remaining octants, which should reduce the total runtime to an eighth of the original process.

The method results in an interpretable representation of the volume, but it lacks most of the fine visual details in the data, making it not suitable for visual analysis. More advanced methods exist to perform this task [18] [35] [7], which may yield better results. However, due to time constraints, these methods were not tested; therefore, it is difficult to discuss their relative performance within the context of the Unity engine. Figure 5 shows the visual quality achieved with the volume rendered at 128 and 256 volume resolutions. For this specific data set (a cropped version of the HD 97048 data set), the engine crashes after the volume resolution exceeds 300; therefore, a resolution of 256 is close to the maximum quality that we can achieve with this method for this data set. The results show that even at the highest working resolution, the visual quality is not satisfactory with a great amount of information lost from the input data.

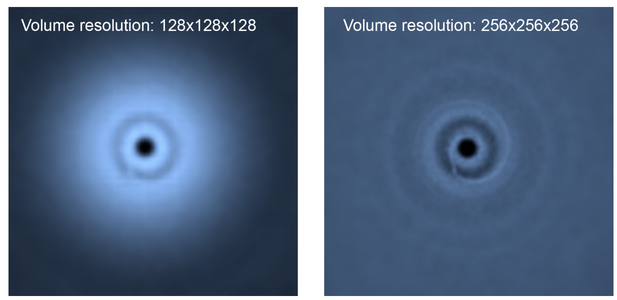

Figure 5: Comparison of the visual quality of two different volume resolutions. Left: Volume resolution = 128. Right: Volume resolution = 256.

Using the 3-D texture, haptic feedback can be easily implemented by looking up the quantity of the current cells intersecting with the controller pointer's position. Implementation of the haptic feedback functionality is discussed in more detail in Section 5.3. The volume rendering prototype can also be integrated with Scaptics [40] to utilise its existing haptic infrastructure.

5.2.2 Particle billboards

The `Immersive Analytics Toolkit` (IATK) [11] is an open-source VR data visualisation package for Unity with a robust framework for interactivity and performance. More importantly, the architecture provides easy to use API to create custom data visualisations out of the box. With publicly available source code, it is also suitable for deep modifications to create new visualisation types.

In computer graphics, a *billboard* refers to a 2-D image constantly rotated in 3-D space that always faces directly at the camera. This is often used for particle systems in video games to simulate a volume such as smoke, gas, or fire. It is most often used to represent a sphere using only a circle sprite. This is perfect for SPH data, as each particle in SPH can be thought of as a perfect spherical distribution of volume and, therefore, can be accurately represented as the billboard of a 2-D projection of such volume. A demonstration of this concept is shown in Figure 6.

The current IATK pipeline provides a CSV file parser with API to create custom parsers for other formats and a GPU-accelerated particle billboard renderer. The graphics pipeline for IATK `Scatterplot` utilises a mesh and shaders to parallelise the

[12] A ghost particle refers to a particle with 0 smoothing length h that does not contribute to the data set in any form other than to ensure the ranges in all 3 spatial dimensions are equal.

[13] Dividing the kernel cube using three mutually perpendicular planes intersecting at the centre of the cube gives eight equal parts, with each called an *octant*.

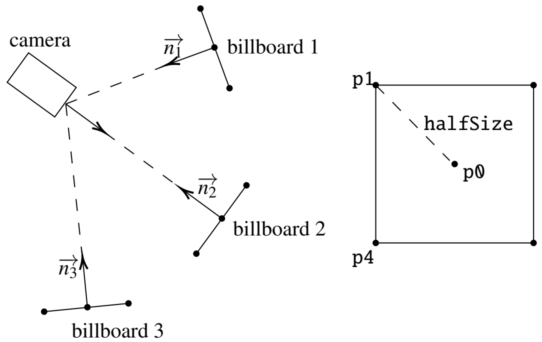

Figure 6: Visual demonstration of a particle billboard. Left: A billboard's normal vector $\vec{n_i}$ is pointing towards the camera's position. Right: The four points `p1`, `p2`, `p3`, and `p4` are calculated using the dimensionless particle `p0`'s position, the normal vector $\vec{n_i}$, and the billboard's `halfSize` parameter.

drawing of data points. Using the Unity engine's path tracing render pipeline and particle billboards procedure of IATK, we can change the data point to represent SPH particles and simulate the SPH rendering procedure implemented in SPLASH. The key difference is that IATK maps the colour scale to the data point-wise, while in SPLASH the colour mapping is implemented in post-processing with ray-tracing.

We designed, implemented and examined two different methods to interpolate the SPH kernel: *Method 1.* Implicit kernel interpolation and *Method 2.* Approximation with 2-D Gaussian sprites.

Method 1 works by passing the particle's attributes to the geometry shader and calculating the effective half-size of the billboard using the given h_0 value, and within the fragment shader, the particle quantity distribution is calculated per pixel and set as the pixel's alpha value. *Method 2* is similar but instead of dynamically calculating the alpha distribution, it is approximated by a pre-computed 2-D Gaussian kernel texture, which is then mapped to the billboard in the fragment shader. We chose to forgo *Method 1* as it is less efficient. The 2-D Gaussian texture of resolution r is generated by **Algorithm 2**.

Algorithm 2 Generating 2-D Gaussian texture

`Image` $\leftarrow$ `Texture2D`(r)
$\sigma \leftarrow \frac{r}{3}$
for each pixel (x,y) **do**
  d $\leftarrow$ distance from (x,y) to center
  `Image`$[x,y] = G(d)$
end for

To create the billboard with accurate size, the smoothing length in Unity spatial units h_U has to be calculated. Assume that the `IATK Visualisation BigMesh` object is a cube of side length L in Unity spatial units:

$$h_U = \frac{h_0 \cdot L}{D}. \tag{11}$$

The half-size of the billboard refers to half the diagonal of the billboard square with a side length of $3 \cdot h_U$. In order to provide the particle shader with the correct particle value, the shader programme input has been modified to utilise the fourth channel of the `NORMAL` shader input [14] to store the particle's h_0 value. Using this method, four additional values can also be given to the shader programme using the `TANGENT` channel if needed for other functionalities.

[14] https://docs.unity3d.com/Manual/SL-ShaderSemantics.html

IATK provides the ability to dynamically change the blend mode of the shader; however, for SPH data the traditional transparency blend mode `Blend SrcAlpha OneMinusSrcAlpha` is most relevant. For a pixel (x,y) of particle i, with the alpha value $\alpha_i(x,y)$ and colour c_i, its final intensity $I_i(x,y)$ is calculated by:

$$I_i(x,y) = \alpha_i(x,y) \cdot c_i + (1 - \alpha_i(x,y)) \cdot \sum_{j}^{i-1} c_j. \tag{12}$$

This is different from how SPLASH describes the blending process, but returns a satisfactory result with no information loss. A visual comparison of the SPLASH renderer and the particle billboard renderer can be seen in Figure 7 using the HD 97048 simulation data set (1.04 million particles) with the SPLASH render taken from the Figure 3 of the source paper [34]. The difference in colour mapping is caused by a different colour scale and particle-wise colouring technique instead of the screen-wise technique used in SPLASH. In terms of rendering speed, for the same plot at `900x900` resolution: SPLASH rendered the frame in 0.92 seconds, while the particle billboard rendered the frame in 8.5 milliseconds.

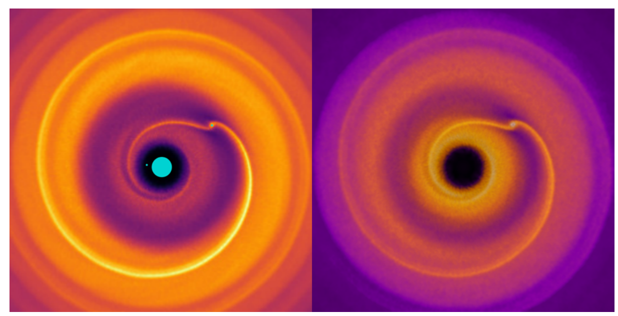

Figure 7: Visual clarity comparison. Left: SPLASH plot from source paper. Right: Particle billboard prototype. Note: The blue marker is not part of the render but an annotation by the author.

To maximise performance, the visualisation is not created using the IATK GUI but rather via a custom API script. This is because the GUI has per frame computations for responsive UI features. For the purpose of the prototype, the generation of the visualisation is static.

5.3 Interactivity

A key component of the VR prototype is the ability to use haptic feedback to encode an additional dimension of data during exploration. Works on the Scaptics project [40] have shown the benefit of haptic feedback in 3-D data visualisations can help users find regions of interests better than with traditional displays.

We developed an easy-to-implement, yet effective method to map SPH data variables to haptic feedback in Unity. For the volume rendering prototype, this process is trivialised by simply using the controller's position inside the data volume to accumulate quantities inside the cells within the *scanning* radius and trigger controller vibration accordingly. The IATK-based particle billboard prototype required more effort to enable this feature, due to the *dimensionless* nature of the data points in 3-D space as the volume interpolation happens on the GPU later in the rendering pipeline. This can be solved by creating an invisible volume cube, similar to a volume rendering `Texture3D`, and mapping our chosen variable using the same process described in Section 5.2.1. Due to the cube not being used for rendering, we can use a much lower resolution cube to encode the haptic feedback data. In the user study, we used a resolution $\mathtt{R}_h = 256$.

In practise, we found that the process of simply accumulating quantities within a cell is not meaningful for some variables like velocity. An additional option was created to let user decide whether they want the data to be mapped accumulatively or be averaged cell by cell. An additional array of length $\mathtt{R}_h^3$ is created to store the total number of particles that contributed to the quantity in a cell, and then the final quantity is divided by this sum to obtain the quantity averaged in the cell. This can be modified to retrieve a regional average as well, which might be more meaningful to find variables such as local average temperature or local average velocity magnitude.

This *haptics* cube is then situated within the Unity scene at the world origin, the same initial position as the visualisation object when the scene is initiated. A controller script is then added to read the controller's position within the visualisation object coordinate system and fetch the corresponding cell regions in the haptics cube. In Unity this can be done using the built-in world-object coordinate system methods. For implementations outside of the Unity engine, this process can be understood as: given the visualisation object $\mathtt{V}$ initialised at world position $(0,0,0)$, the transformation M is applied (user moving and rotating the visualisation), we can apply the transformation M^{-1} to the controller position (x_c, y_c, z_c) to get the relative position within the haptics cube (x_h, y_h, z_h). The initialised haptics cube has side lengths of 1, the controller's cell position is given by $(x_h \cdot \mathtt{R}_h, y_h \cdot \mathtt{R}_h, z_h \cdot \mathtt{R}_h)$.

To map between cell values to controller vibration, the cell values are normalised between 0 and 1. This value can then be mapped linearly to the vibration intensity or by using a sigmoid function. In the user study, the transfer function for vibration intensity I_v and input cell quantity q_c is implemented using a custom S-curve[15]:

$$I_v = \frac{1}{1 + (\frac{q_c}{1-q_c})^{-\beta}}. \tag{13}$$

Other interactivity features included in the prototype were the ability to pick up, rotate, and place the rendered volume in 3-D space to introduce intuitive 3-D navigation. Zooming was intentionally left out as the performance heavily depends on minimising the effect of overdrawing of particles, with frametime scaling exponentially with total pixel count times number of particles. An indicator is shown when the user is holding down the button for haptic scanning functionality, with green showing low haptic quantity and red showing high haptic quantity. A colour scale and legend for mapped colour and mapped haptic dimension are also included for user preference, this can be seen in Figure 8.

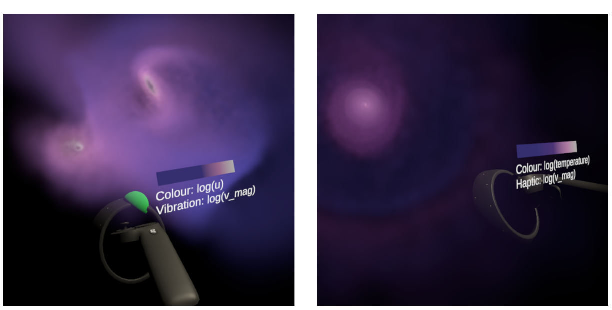

Figure 8: Screenshots of user's view in VR prototype. Left: The flyby data set, with user scanning for haptic returning low velocity magnitudes. Right: The common-envelopes data set.

5.4 Performance evaluation and analysis

The performance benchmarks were tested on a GIGABYTE AERO 15 laptop with an Intel i7-11800H 2.30GHz CPU, 16 GB of DDR4 RAM and a NVIDIA GeForce RTX 3080 Laptop GPU with 8 GB of VRAM. The HP Reverb VR1000-220a headset was used for all performance tests and user study. The prototypes were set to be rendered at 1080 by 1080 resolution per eye using SteamVR upscaling functionality. This is due to balance between visual clarity and usable framerates. The headset's native resolution of 2160 by 2160 pixels per eye proved to be too demanding for the renderer.

Volume rendering evaluation. A series of tests was carried out to time the GPU computation time for the generation of the volume cube at different resolutions. For each resolution, 10 test runs were recorded and averaged. The data set used is HD97048 data set without particles larger than 128 cubes in radius as the large particles exponentially increase the runtime while obscuring important details. The highest resolution tested was 512 before running out of memory. The results for the tests are displayed in Figure 9. Note that this is only the GPU computation time, in practice, CPU buffer initialisation is also a factor.

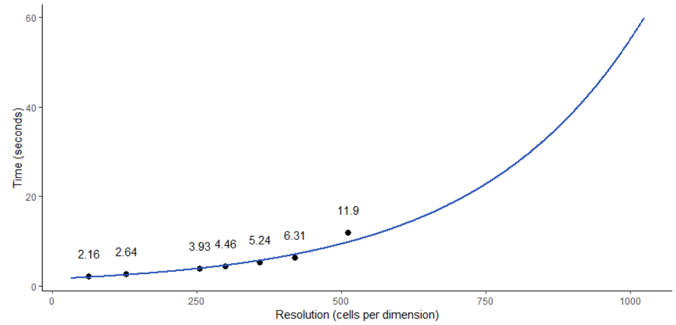

Figure 9: Total GPU `Texture3D` generation time in seconds over different volume cube resolutions. Data: HD 97048 data set without particles larger than 128 cubes in radius.

Given the strict performance budget of VR, direct volume rendering acts as a good starting point. However, the technique has poor scalability. The time complexity of the 3-D texture generation is $O(n \cdot k^3)$ for the data set with n particles of kernel size k and the space complexity is $O(gridResolution^3)$. Furthermore, the current linear transfer function is not desirable for astrophysics SPH data sets. A non-linear transfer function is compulsory to achieve desired visual intelligibility of SPH data with volume rendering, but often more computationally expensive [16].

In general, to avoid crash, the volume rendering prototype failed to meet the visual clarity criteria, which deemed it unsuitable for the user study.

Particle billboard evaluation. The particle billboard prototype's haptic cube generation time testing process involves running the generation for each data set 10 times and the average times are displayed in Table 1.

Data set	Time (seconds)
Common-envelopes (50k rows)	1.31
HD 142527 (878k rows)	4.35
Stellar flyby (50k rows)	1.82

Table 1: Pre-computation (haptic cube generation) time for billboard prototype.

Real-time performance benchmark for the billboard prototype was performed using a sample use case scenario, with the user exploring the data naturally and the SteamVR frametime graph was recorded. The scenario involves 3 phases: *Phase 1.* The user is moving around inside the data with the data volume covering the entire view, *Phase 2.* The user is looking closely at the densest region in the data with

[15]This function was suggested by user Ismam Huda on https://stats.stackexchange.com/q/289477

the region taking up most of the view, *Phase 3*. The user moved outside of the data, and walked around while looking at it from a distance. The recorded graph is shown in Figure 10. We can see that the performance is right on the limit of our performance budget with sporadic spikes up to 21ms frametime when the user is looking around inside the data, while the performance is halved when the user is looking closely at the densest region with average frametime above 20 ms and peaked at 27ms. Lastly, when the user is exploring the data at arms length away or is generally outside of the data volume, the performance is on average twice as recommended.

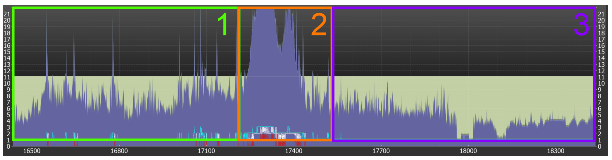

Figure 10: Recorded frametime graph of demo data exploration process on the stellar flyby data set. The pale green region shows the desired frametime (under 11 milliseconds). Phase 1: User are looking inside the data. Phase 2: User are looking very closely to the densest region in the data. Phase 3: User is walking around the outside of the data with the entire data set in view at all time.

The main challenge with particle billboards is *overdraw*. This is a common problem with transparent object rendering in computer graphics and is usually unavoidable when rendering semi-transparent volumes. Overdrawing refers to a pixel's value being updated more than once, or more than it needs to depending on the context. For our scenario, assume that each particle contributes 0.01 alpha to a pixel. We would only need to draw this pixel 100 times before it becomes saturated. However, if our data set contains 1 million particles, the pixel has to be drawn 10,000 times more than required. This is the main bottleneck for performance.

The relationship between data resolution and performance is erratic. It can be generally be assumed that with more particles in the data set, the higher the average frametime. The general time complexity is $O(W \cdot H \cdot n)$ for a display resolution of W by H rendering a data set with n particles, each filling the entire screen. However, data resolution itself is not a good indicator of performance. This is because the performance heavily depends on the number of pixels being drawn on the screen, so a data set with more smaller particles may still perform better than a data set with fewer but larger particles. Distribution of particles' h value should also be taken into account when estimating performance of a data set.

The particle billboard meets all design goals except for the performance criterion, which heavily depends on the use case and the data set. We decided the performance is good enough for the user study, with some specifically chosen sets of data. This introduced another variable we want to observe from the user study: whether the users are bothered by the framerates and do they feel nauseous using the prototype for an extended amount of time.

6 User Study

The user study looks into the prospects of using virtual reality platforms in astronomical research involving SPH data as part of the scientific analysis process, as well as possibilities for broader use cases like public outreach and education.

6.1 Study design

The study is designed in a *naturalistic* manner [4], to simulate a real-world use case, where a participant will perform data exploration using SPLASH and VR. The purpose is to organically identify flaws of the platform and elicit suggestions for improvements and extensions. This can potentially yield more informative data, as the researcher is not from the astrophysics domain, thus limited in ability to design a controlled procedure that includes important external variables in a usual SPH data exploration process. However, there are some drawbacks to this approach that must be acknowledged and addressed [44]:

1. The participant's behaviour may be affected by the fact that they are being observed.
2. The participant may try to do what they think the researcher expects to see as positive.
3. The recorded observation can be influenced by the researcher's prior experience, beliefs, and expectations.

The process has been designed to be as neutral as possible, with participants encouraged to express critical opinions throughout the study and reminded that input on platform flaws is greatly desired. Open-ended short questions are included in the feedback questionnaire at the end to eliminate observer bias in observation recording. To eliminate observer bias, the responses of the questions will be cross-referenced with the recorded observations. Participants can still perform subtle subconscious actions that can only be documented by the observer. Observer bias can influence these results.

6.2 Participants

We recruited two main groups of participants: *A. Students and researchers in Astronomy and Astrophysics*, and *B. Interested users from outside of Astronomy and Astrophysics*. Group A is the primary research target; by gathering input from students and professionals inside the domain, limitations and functionalities specific to SPH data analysis can be identified which will help future design and technical decisions of VR SPH visualisation tools. Group B can provide insights on how such technology can help in public outreach. The specific area of expertise for the second group might not be relevant to the research purpose, but is still recorded for future analysis. The study was conducted with the participation of 16 volunteers from each group. The distribution is shown in Table 2.

Area	Count
Astronomy and Astrophysics (A)	12
Information Technology (B)	2
Business and Commerce (B)	2

Table 2: Area of expertise distribution.

Specifically, the academic experience distribution within Astronomy and Astrophysics group is shown in Table 3. Such distribution may provide a balanced overview of both beginner, intermediate, and professional user requirements.

Academic level	Count
Undergraduate	4
PhD	4
Post-PhD	4

Table 3: Group A academic level distribution.

For participants whose expertise is outside of the domain: three were working in industry at graduate/entry level, and one currently pursuing a PhD.

6.3 Tasks

The participants are asked to explore 3 data sets [37] [27] [6] using the desktop environment with SPLASH, then using the VR environment with the IATK-based prototype. The participant can choose to finish the exploration when they feel like they have a good understanding of the 3-D structure of the data and the spatial distribution

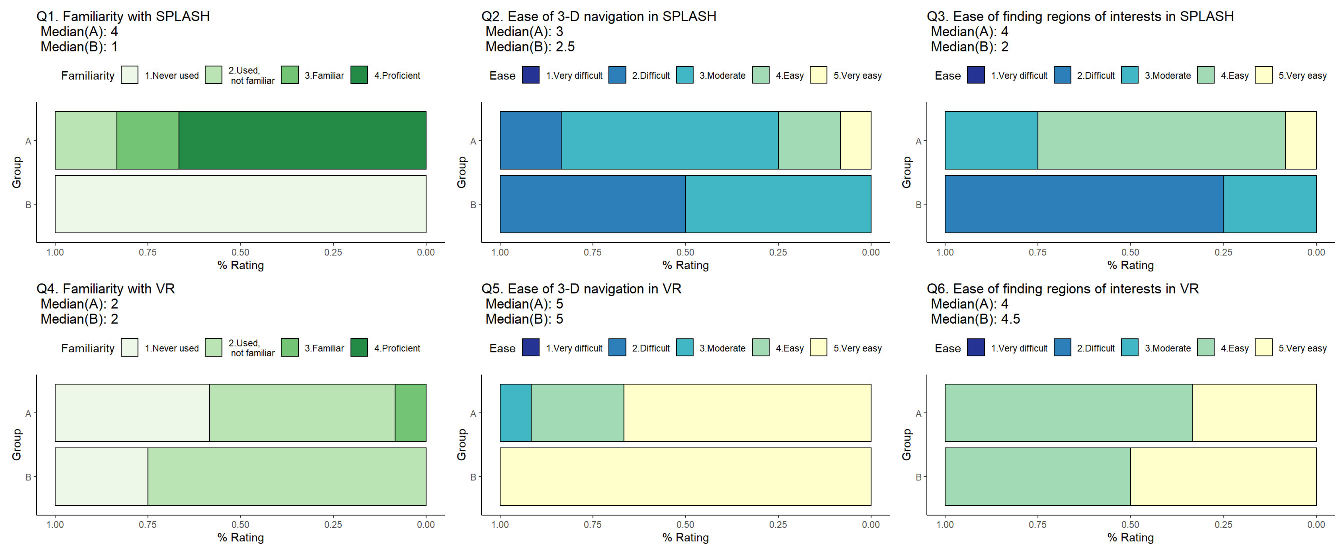

Figure 11: Responses from Question 1-6 from Group A and B.

of physical quantities in the data set. Participants will first be introduced to each environment, how to use them, the basic controls and functionalities. For participants who have no experience with SPLASH, a printed instruction sheet is also provided. A simple VR practice scene is included for participants unfamiliar with VR environment and controls, this includes instructions for movements and basic object interactions (pick up, rotate, scanning).

The participants will be exploring the following three data sets:

- The common-envelopes simulation data set [27]. The VR prototype mapped colour to log(temperature) and haptic feedback to log(local average velocity magnitude).
- The HD 142527 protoplanetary disc simulation data set [38]. The VR prototype mapped colour to log(density) and haptic feedback to log(local average velocity magnitude).
- The stellar flyby simulation data set [6]. The VR prototype mapped colour to $log(u)$and haptic feedback to log(local average velocity magnitude).[16]

In the VR environment, participants are also introduced to the haptic functionality. The participant can use the controller to *scan* for a mapped quantity in the volume such as temperature or velocity. To move on to the next data set, participants will notify the attending researcher once they are confident in their understanding of the data. After the VR activities, a short unstructured interview is conducted where the participant is asked about their overall experience, what they enjoyed best and least, whether they feel limited in terms of functionalities and what can be improved upon. Lastly, the participants are asked to fill out a questionnaire to evaluate usability and effectiveness of the desktop and VR platforms. Open-ended short questions regarding user feedback similar to the verbal interview are also included but with the purpose of gathering more detailed explanations.

[16] u can be understood as being equivalent to temperature. Simulation data produced by PHANTOM assigns u to be temperature with account for the gas radiation term.

6.4 Results

6.4.1 Quantitative responses

The first six questions of the questionnaire ask participants to describe their experience with SPLASH and VR and the ease of navigation and exploration for each software. The response distribution is shown in Figure 11.

The participants are then asked to identify the preferred platform for understanding 3-D structure of the data and identifying regions of interest (densest regions, hottest regions, regions with highest local velocity magnitude):

7. Which environment helped you understand the 3-D structure of the data better?
8. Which environment helped you identify regions of interest in the data better?

The preference of each platform is then tallied for each participant group is shown in Table 4.

Question	SPLASH **(A)**	VR **(A)**	SPLASH **(B)**	VR **(B)**
7	0	12	0	4
8	2	10	0	4

Table 4: Platform preference for each participant group.

For Question 9, the participants were asked if they thought with further improvements, VR visualisation can be used in their work. For Group B, the question was altered to ask if they think with further improvements, VR visualisation can be used to help them understand astronomical data. 15 participants responded **Yes**, and 1 participant from Group B responded **Maybe**. When asked to clarify in the informal interview, the participant mentioned that they were not sure how technically feasible it was to further develop the VR platform therefore wasn't confident enough to give a definitive answer.

Questions 10 and 11 asked participants to gauge how helpful each platform was in providing the information they needed to understand the data. Questions 12 and 13 asked the participants to rate how confident they were about understanding the data after the exploration using each platform. The response distribution is shown in Figure 12.

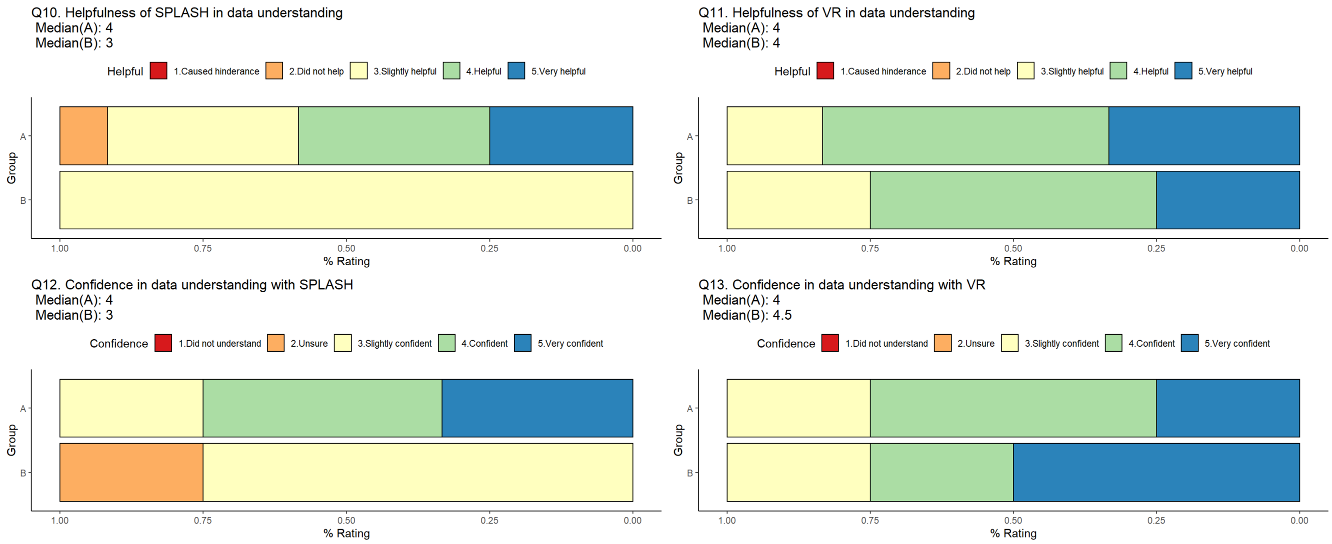

Figure 12: Responses from Question 10-13 from Group A and B.

6.4.2 Qualitative responses

No user study of this exact use case has been conducted previously, as such, open coding was chosen for content analysis. There are three main topics for which participants were asked to provide input in the questionnaire:

9. Identify the benefits and drawbacks of the desktop program (SPLASH).
10. Identify the benefits and drawbacks of the VR prototype.
11. Suggest potential improvements for the VR prototype.

Using open coding, the hierarchy coding frames are created. The observer's recordings are also analysed using open coding and merged with the codes and themes found from the questionnaire. The coding frames for Question 9 and 10 are shown in Figure 13 and Figure 14, respectively.

Most participants agreed that the main drawbacks of SPLASH are unintuitive navigation and hard to understand 3-D structure of data using 2-D projections: *"Difficulty understanding 3d structure using a cross section or projection plot."* (P8), *"It can be used anywhere and can extract data directly for analysis but with 2D representation it is not so good to give a global view of the data."* (P9). However, these aspects are often highly rated for the VR prototype: *"It is extremely good at giving you an idea of the 3D structures and discovering things that you don't imagine."* (P10).

Many participants did not use the haptic functionality for most of the exploration, but those who did use it extensively gave positive feedback: *"I really love the haptic way to feel to data, that allows real 'feel' of the data!"* (P9).

However, the aspects where SPLASH excels are commonly noted as lacking for the VR prototype. These are mainly to do with scientific plotting functionalities: quantitative functionalities and changing plot parameters. These features are not present in the VR prototype, which lead users to feel like it is more for general exploration, than analysis: *"VR gives a much more insightful feel about the data which makes it easier to decide the direction of analysis, but it is hard to do the actual analysis in VR."* (P9).

One participant gave an interesting input: *"I also felt the main benefit was in the controls and not necessarily the VR. I feel it would have had the same effect if I was using the VR controls on a 2D monitor."* (P10), raising a new question regarding whether the main benefit was from the real-time 3-D renderer or the VR platform.

In general, the participants mostly agreed that the VR platform helped them better understand the 3-D distribution of the data, but it is limited in terms of quantitative functionalities and, most importantly, can only show low resolution data sets[17]. None commented on the framerate. Only 6 out of 16 mentioned the benefits of haptic feedback functionality in the questionnaire. One participant mentioned feeling slight dizziness after the VR activity, the rest mentioned they felt fine in this regard when asked by the researcher after the activity.

The key themes found for Question 11 are (listed by the number of occurrences in descending order):

- Quantitative scales, legends, markers.
- Ability to change plotting parameters.
- Support for higher resolution data.
- Animated plots, scrolling data through time.
- Annotations.

Most notable findings include: 14 participants performed majority of the process standing still while only 2 utilised movement in physical space; only half of the participants paid attention to the use of the haptic functionality, the rest spent most of their time exploring the data visually; 5 participants tried to *zoom in* using the two-hand pulling/pinching gesture, much like on a touch screen. This can serve as future reference for interaction design.

7 Research results discussion

From the user study it can be seen that the main limitations of the desktop platform are unintuitive controls, difficult navigation, and hard to understand and visualise the 3-D structure of the data. These are also the most notable advantage of the VR platform. Overall, participants from both groups agreed that their understanding of the 3-D structure of the data benefits from using the VR platform.

The haptic feedback feature was not often used, and most of the participants spent most of their time exploring the data visually. Participants who spent a great amount of time using the haptic feature

[17]Resolution of a SPH data set refers to the number of particles used in the simulation

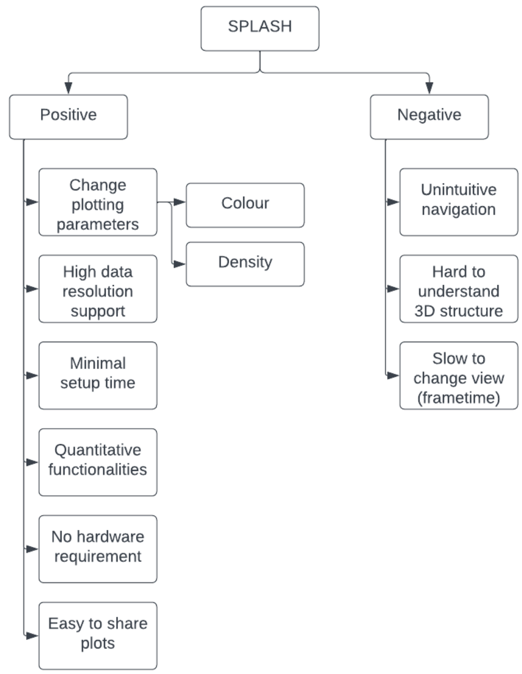

Figure 13: Hierarchical coding frame diagram for user feedback for SPLASH.

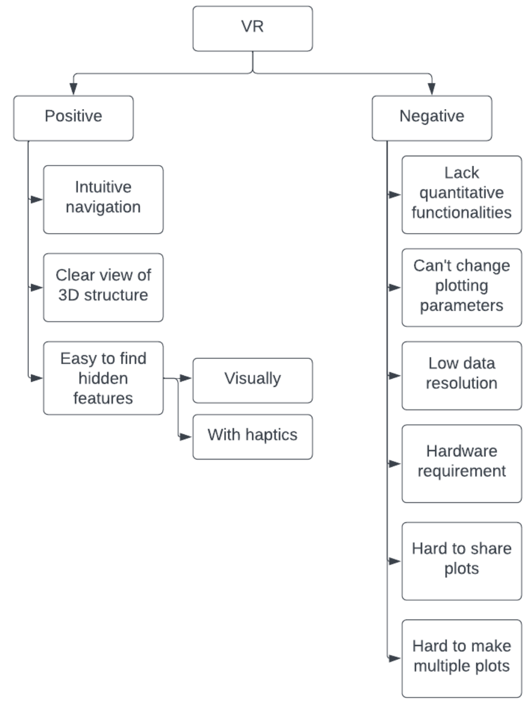

Figure 14: Hierarchical coding frame diagram for user feedback for VR.

found it helpful. We believe that haptic feedback is an important and helpful feature but were under-utilised due to rudimentary design and functionality.

Most requested features by the participants can easily be implemented except for two that pose technical challenges: higher resolution data and animated plots. Higher resolution data support requires rendering optimisation and more performant hardware. It is challenging to optimise the particle billboarding method because of its quality-focused nature, but we believe that with more work it can be improved. For future work, if an early stopping mechanism for the Z-buffer can be implemented where a pixel draw call is discarded if the pixel value is saturated, this can greatly improve performance. For animated plots, a caching mechanism can be developed to pre-generate multiple data sets, and implement function calls to show/hide the data sets in order. This can be tricky due to memory constraints, but it is possible. Overall, we find the results promising and further development of a full-fledged VR analysis tool for SPH data is feasible.

Although current performance is not desirable, no participants have mentioned it, unless asked. When asked, some mentioned they did notice some stuttering but didn't mind because this often occurs when the user is looking closely at a region of interest, and only minimal movements are occurring therefore less affected by low framerate. The performance should still be further optimised to guarantee usability.

Group B participants shared similar results with Group A with respect to improvements in data understanding using VR. They also suggested more qualitative functionalities and quality-of-life improvements, which can be easily implemented with more development time. Quantitative results showed that Group B benefited more from VR visualisation compared to Group A, possibly due to their lack of experience with SPLASH and scientific visualisation programmes.

We found that the study design decisions made lead to more diverse responses and greatly helped ideation for further developments. However, we believe more controlled studies, both qualitative and quantitive, should be conducted to provide firmer grasp of what aspects are more important than others and develop methods to accurately quantify SPH data understanding.

8 Conclusion

As an exploratory study, we managed to answer conclusively that a fully functional VR visualisation tool for SPH should be developed. Our results showed that VR helps users in both groups better understand 3D data and identified features that researchers would like to have for the final product. One research question we weren't able to fully answer was regarding haptic feedback, requiring further investigation to be conducted. We discovered another intriguing question of whether a simple desktop tool with real-time SPH rendering can already provide benefits to astrophysicists with their research, which requires further exploration. We also found that volume rendering is not a promising method for VR SPH visualisation, and particle billboard is a better option to pursue if one were to develop the software, albeit requiring effort into optimisations.

References

[1] B. Adams, T. Lenaerts, and P. Dutré. Particle splatting: Interactive rendering of particle-based simulation data. 2006.

[2] O. Agertz, B. Moore, J. Stadel, D. Potter, F. Miniati, J. Read, L. Mayer, A. Gawryszczak, A. Kravtsov, Nordlund, F. Pearce, V. Quilis, D. Rudd, V. Springel, J. Stone, E. Tasker, R. Teyssier, J. Wadsley, and R. Walder. Fundamental differences between SPH and grid methods. *Monthly Notices of the Royal Astronomical Society*, 380(3):963–978, 08 2007. doi: 10.1111/j.1365-2966.2007.12183.x

[3] J. Ahrens, B. Geveci, and C. Law. Paraview: An end-user tool for large-data visualization. In *The Visualization Handbook*, 2005.

[4] M. V. Angrosino. *Naturalistic observation*. Routledge, 2016.

[5] I. Babuška, U. Banerjee, and J. E. Osborn. Meshless and generalized finite element methods: A survey of some major results. In M. Griebel and M. A. Schweitzer, eds., *Meshfree Methods for Partial Differential*

Equations, pp. 1–20. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[6] E. M. A. Borchert, D. J. Price, C. Pinte, and N. Cuello. On the rise times in FU Orionis events. , 510(1):L37–L41, Feb. 2022. doi: 10.1093/mnrasl/slab123

[7] J. Borrow and A. J. Kelly. Projecting sph particles in adaptive environments. 2021. doi: 10.48550/ARXIV.2106.05281

[8] S. Bryson. Virtual reality in scientific visualization. *Commun. ACM*, 39(5):62–71, May 1996. doi: 10.1145/229459.229467

[9] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, K. Marriott, F. Schreiber, and E. Wilson. Immersive analytics. In *2015 Big Data Visual Analytics (BDVA)*, pp. 1–8, 2015. doi: 10.1109/BDVA.2015.7314296

[10] A. Comrie, A. Sivitilli, F. Vitello, T. Jarrett, and L. Marchetti. iDaVIE-v: Immersive Data Visualisation Interactive Explorer for Volumetric Rendering. Zenodo, Mar. 2021. doi: 10.5281/zenodo.4614116

[11] M. Cordeil, A. Cunningham, B. Bach, C. Hurter, B. H. Thomas, K. Marriott, and T. Dwyer. Iatk: An immersive analytics toolkit. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 200–209, March 2019. doi: 10.1109/VR.2019.8797978

[12] A. J. C. Crespo, J. M. Domínguez, B. D. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro, and O. García-Feal. DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH). *Computer Physics Communications*, 187:204–216, Feb. 2015. doi: 10.1016/j.cpc.2014.10.004

[13] K. Dolag, M. Reinecke, C. Gheller, and S. Imboden. Splotch: visualizing cosmological simulations. *New Journal of Physics*, 10(12):125006, Dec. 2008. doi: 10.1088/1367-2630/10/12/125006

[14] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, p. 65–74. Association for Computing Machinery, New York, NY, USA, 1988. doi: 10.1145/54852.378484

[15] A. Drogemuller, A. Cunningham, J. Walsh, M. Cordeil, W. Ross, and B. Thomas. Evaluating navigation techniques for 3d graph visualizations in virtual reality. In *2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA)*, pp. 1–10, 2018. doi: 10.1109/BDVA.2018.8533895

[16] K. Engel, M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware, HWWS '01, ACM*, pp. 9–16, 2001.

[17] U. Erra, D. Malandrino, and L. Pepe. Virtual reality interfaces for interacting with three-dimensional graphs. *International Journal of Human–Computer Interaction*, 35:75 – 88, 2018.

[18] R. Fraedrich, S. Auer, and R. Westermann. Efficient high-quality volume rendering of sph data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1533–1540, 2010. doi: 10.1109/TVCG.2010.148

[19] O. García-Feal, A. J. C. Crespo, and M. Gómez-Gesteira. VisualSPHysics: advanced fluid visualization for SPH models. *Computational Particle Mechanics*, Jan. 2021. doi: 10.1007/s40571-020-00386-7

[20] N. Gnedin. IFrIT: Ionization FRont Interactive Tool. Astrophysics Source Code Library, record ascl:1304.019, Apr. 2013.

[21] R. T. Inácio, T. H. C. Nobrega, D. D. B. Carvalho, and A. v. Wangenheim. Interactive simulation and visualization of fluids with surface raycasting. In *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 142–148, 2010. doi: 10.1109/SIBGRAPI.2010.27

[22] S. H. Khandkar. Open coding. *University of Calgary*, 23:2009, 2009.

[23] J. L. Kline and P. L. Volegov. Toward 3d data visualization using virtual reality tools. *Review of Scientific Instruments*, 92(3), 3 2021. doi: 10.1063/5.0040468

[24] A. Knoll. A survey of octree volume rendering methods. In *VLUDS*, 2006.

[25] A. Knoll, R. K. Morley, I. Wald, N. Leaf, and P. Messmer. *Efficient Particle Volume Splatting in a Ray Tracer*, pp. 533–541. Apress, Berkeley, CA, 2019. doi: 10.1007/978-1-4842-4427-2_29

[26] C. B. Laney. *Computational gasdynamics*. Cambridge university press, 1998.

[27] M. Y. M. Lau, R. Hirai, M. González-Bolívar, D. J. Price, O. De Marco, and I. Mandel. Common envelopes in massive stars: towards the role of radiation pressure and recombination energy in ejecting red supergiant envelopes. , 512(4):5462–5480, June 2022. doi: 10.1093/mnras/stac049

[28] B. Liu, G. J. Clapworthy, F. Dong, and E. C. Prakash. Octree rasterization: Accelerating high-quality out-of-core gpu volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 19(10):1732–1745, 2013. doi: 10.1109/TVCG.2012.151

[29] N. Lukac, D. pelic, and B. alik. Hybrid visualization of sparse point-based data using gpgpu. In *2014 Second International Symposium on Computing and Networking*, pp. 178–184, 2014. doi: 10.1109/CANDAR.2014.76

[30] K. Marriott, J. Chen, M. Hlawatsch, T. Itoh, M. A. Nacenta, G. Reina, and W. Stuerzlinger. *Immersive Analytics: Time to Reconsider the Value of 3D for Information Visualisation*, pp. 25–55. Springer International Publishing, Cham, 2018. doi: 10.1007/978-3-030-01388-2_2

[31] J. J. Monaghan. Smoothed particle hydrodynamics. , 30:543–574, Jan. 1992. doi: 10.1146/annurev.aa.30.090192.002551

[32] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703–1759, jul 2005. doi: 10.1088/0034-4885/68/8/r01

[33] K. Nagao, Y. Ye, C. Wang, I. Fujishiro, and K.-L. Ma. Enabling interactive scientific data visualization and analysis with see-through hmds and a large tiled display. In *2016 Workshop on Immersive Analytics (IA)*, pp. 1–6, 2016. doi: 10.1109/IMMERSIVE.2016.7932374

[34] C. Pinte, G. van der Plas, F. Ménard, D. J. Price, V. Christiaens, T. Hill, D. Mentiplay, C. Ginski, E. Choquet, Y. Boehler, G. Duchêne, S. Perez, and S. Casassus. Kinematic detection of a planet carving a gap in a protoplanetary disk. *Nature Astronomy*, 3:1109–1114, Aug. 2019. doi: 10.1038/s41550-019-0852-6

[35] M. Piochowiak, T. Rapp, and C. Dachsbacher. Stochastic volume rendering of multi-phase sph data. *Computer Graphics Forum*, 40(1):97–109, 2021. doi: 10.1111/cgf.14121

[36] D. J. Price. SPLASH: An Interactive Visualization Tool for Smoothed Particle Hydrodynamics Simulations. Astrophysics Source Code Library, record ascl:1103.004, Mar. 2011.

[37] D. J. Price, N. Cuello, C. Pinte, D. Mentiplay, S. Casassus, V. Christiaens, G. M. Kennedy, J. Cuadra, M. Sebastian Perez, S. Marino, P. J. Armitage, A. Zurlo, A. Juhasz, E. Ragusa, G. Laibe, and G. Lodato. Circumbinary, not transitional: on the spiral arms, cavity, shadows, fast radial flows, streamers, and horseshoe in the HD 142527 disc. , 477(1):1270–1284, June 2018. doi: 10.1093/mnras/sty647

[38] D. J. Price, N. Cuello, C. Pinte, D. Mentiplay, S. Casassus, V. Christiaens, G. M. Kennedy, J. Cuadra, M. Sebastian Perez, S. Marino, P. J. Armitage, A. Zurlo, A. Juhasz, E. Ragusa, G. Laibe, and G. Lodato. Circumbinary, not transitional: on the spiral arms, cavity, shadows, fast radial flows, streamers, and horseshoe in the HD 142527 disc. , 477(1):1270–1284, June 2018. doi: 10.1093/mnras/sty647

[39] D. J. Price, J. Wurster, T. S. Tricco, C. Nixon, S. Toupin, A. Pettitt, C. Chan, D. Mentiplay, G. Laibe, S. Glover, C. Dobbs, R. Nealon, D. Liptai, H. Worpel, C. Bonnerot, G. Dipierro, G. Ballabio, E. Ragusa, C. Federrath, R. Iaconi, T. Reichardt, D. Forgan, M. Hutchison, T. Constantino, B. Ayliffe, K. Hirsh, and G. Lodato. Phantom: A Smoothed Particle Hydrodynamics and Magnetohydrodynamics Code for Astrophysics. , 35:e031, Sept. 2018. doi: 10.1017/pasa.2018.25

[40] A. Prouzeau, M. Cordeil, C. Robin, B. Ens, B. H. Thomas, and T. Dwyer. Scaptics and highlight-planes: Immersive interaction techniques for finding occluded features in 3d scatterplots. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300555

[41] F. Reichl, M. Treib, and R. Westermann. Visualization of big sph simulations via compressed octree grids. In *2013 IEEE International Conference on Big Data*, pp. 71–78, 2013. doi: 10.1109/BigData.2013.6691717

[42] G. Schaufler. Dynamically generated impostors. In *Proc. GI Workshop on Modeling, Virtual Worlds, Distributed Graphics, 1995*, pp. 129–135, 1995.

[43] W. Song, S. Hua, Z. Ou, H. An, and K. Song. Octree based represen-

tation and volume rendering of three-dimensional medical data sets. In *2008 International Conference on BioMedical Engineering and Informatics*, vol. 1, pp. 316–320, 2008. doi: 10.1109/BMEI.2008.287

[44] J. Sparling and I. Chong. *Natural Observation*, pp. 992–993. Springer US, Boston, MA, 2011. doi: 10.1007/978-0-387-79061-9_1916

[45] V. Springel. The cosmological simulation code GADGET-2. , 364(4):1105–1134, Dec. 2005. doi: 10.1111/j.1365-2966.2005.09655.x

[46] T. Umenhoffer, L. Szirmay-Kalos, and G. Szijártó. Spherical billboards and their application to rendering explosions. In *Proceedings of Graphics Interface 2006*, GI '06, p. 57–63. Canadian Information Processing Society, CAN, 2006.

[47] J. A. Wagner Filho, M. F. Rey, C. M. D. S. Freitas, and L. Nedel. Immersive visualization of abstract information: An evaluation on dimensionally-reduced data scatterplots. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 483–490, 2018. doi: 10.1109/VR.2018.8447558

[48] C. Ware and P. Mitchell. Visualizing graphs in three dimensions. *ACM Trans. Appl. Percept.*, 5(1), jan 2008. doi: 10.1145/1279640.1279642

[49] J. Wilhelms, A. Van Gelder, P. Tarantino, and J. Gibbs. Hierarchical and parallelizable direct volume rendering for irregular and multiple grids. In *Proceedings of Seventh Annual IEEE Visualization '96*, pp. 57–63, 1996. doi: 10.1109/VISUAL.1996.567606

[50] Y. Yang, M. Cordeil, J. Beyer, T. Dwyer, K. Marriott, and H. Pfister. Embodied navigation in immersive abstract data visualization: Is overview+detail or zooming better for 3d scatterplots? *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1214–1224, 2021. doi: 10.1109/TVCG.2020.3030427

Appendices

A Code for user study prototype

The entire code base for the prototype used can be found here: https://drive.google.com/file/d/12_kMjtugl7OcHnY7qwQ2S-W7uGortN89/view?usp=sharing

The majority of the code base is prior work from IATK. My contribution includes everything inside the `Assets/SPH-study/` directory, and modifications made to `Assets/IATK/` include:

- `Shaders/GSBillboard.shader`: SPH interpolation shader.
- `Scripts/Controller/AbstractVisualisation.cs`.
- `Scripts/Controller/Visualisation.cs`
- `Scripts/View/BigMesh.cs`.
- `Scripts/View/View.cs`.

B Executable builds user study prototype

The executable VR prototypes used for the user study can be found here: https://drive.google.com/file/d/1dvwP31mL_WWmk0if2sVkCN1-8yfcVdKD/view?usp=sharing

C SPH data sets used for user study

The processed data sets used for the user study can be found here: https://drive.google.com/file/d/1dvwP31mL_WWmk0if2sVkCN1-8yfcVdKD/view?usp=sharing

D User study questionnaire

The questionnaire used for the user study can be found here: https://drive.google.com/file/d/1iUF2KcajJvZTC2UFjOBf9xvv3kg7I1sP/view?usp=sharing

E User study questionnaire response

The questionnaire responses from participants can be found here: https://docs.google.com/spreadsheets/d/1sg4i988Eghy0QwkqucRAxXn1NwPH6-x-/edit?usp=sharing&ouid=112464434066765549966&rtpof=true&sd=true

F Ethics form application approval letter

The ethics form approval letter can be found here: https://drive.google.com/file/d/1LrlMlsTmONLLFdkuOtx0KUg2GszivABS/view?usp=sharing