Interactive 3D Astronomy Visualization

Bhavish Mohee
Computer Science Honours
University of Cape Town
Cape Town, South Africa
MHXBHA001@myuct.ac.za

ABSTRACT

This review focuses and analyses the existing astronomical data visualizations tools, especially CARTA. It also covers the need for more efficient and comprehensible ways of visualizing data as modern radio telescopes become more accurate in producing large and low-noise quality images. Volume rendering is then presented to be a viable option for displaying multi-dimensional astronomical data. Additionally, the review shows how to improve the performance of the rendering process through better architecture and optimisations in order to achieve interactivity on the visualization software application.

CCS CONCEPTS

- **Computing Methodologies ~** Computer Graphics
- **Human-centred Computing** ~ Visualization
- **Human-centred Computing** ~ Interaction design

KEYWORDS

Volume Rendering, CARTA, Ray Casting, Visualization Tools, Data Formats

1 INTRODUCTION

Astronomy, being a visual science, requires presenting large sets of complex data into an efficient and comprehensive visualization allowing astronomers to analyse and draw conclusive statements from the information. Rosolowsky *et al.* [29] stated that visualization is the most dominant way through which astronomers can explore large amount of data, analyse and make conclusive discoveries. With the advent of modern telescopes and their underlying technologies, the image size cultivated from sky surveys is increasing exponentially due to their high amount of visibility data and low noise. Nowadays, facilities with high-end telescopes like MeerKAT, the Australian Square Kilometre Array Pathfinder (ASKAP), the Large Synoptic Survey Telescope (LSST) and the Square Kilometre Array (SKA) can produce high-resolution image cubes varying from gigabytes to terabytes of data which astronomers use for their analysis [4, 17].

As we moved into the era of Big Data and Technology, computer scientists realised the need for a better and more efficient form of visualization algorithm that does not require binary classification of the incoming data [39]. Borkin *et al.* [3] outlined the importance of developing and utilizing 3D visualization tools to their full potential in the Astronomical Medicine Project at the Initiative in Innovative Computing (IIC) at Harvard University. This project was set up to find new approaches for data exploration and analysis based on complex data sets in astronomy and medicine. Similar to how a physician might want to visualise an MRI scan and sections out a tumour out of the volume, astronomers want to explore the possibility of visualizing data generated in 3D from a radio telescope and identify new elements such as stars and constellations. Through amalgamation of astronomical and medical expertise, the Astronomical Medicine project is currently developing tools to address the common attributes of the imaging sciences. Hence, researchers began exploring volume rendering, a visualization technique that computes the volumetric colour and partial transparency for each 3D pixel to form the final image. The latter is significantly more powerful than the conventional rendering methods which extract geometrical surfaces from the volume data. Astrophysical, meteorological and geophysical measurements and computer simulations using models of sky surveys, stress and fluid flow naturally generate a volume data set. This literature review will focus on the importance of volume rendering and the optimisations required to enable real-time or near real-time interactions with the data [37].

2 CARTA

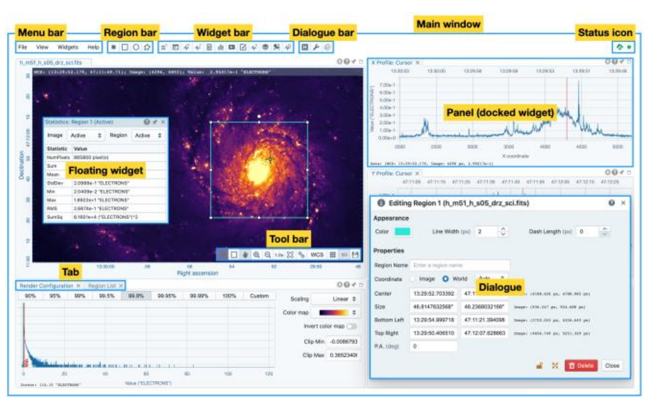

Figure 1: The Graphical User Interface (GUI) of CARTA with its widgets and functions labelled in yellow boxes. Retrieved from: https://carta.readthedocs.io/en/latest/about_gui.html

CARTA, the acronym for Cube Analysis and Rendering Tool for Astronomy, is an image visualization and analysis tool whose architecture was redesigned in 2018 by the CARTA development team [38]. The team includes the following institutes: The Academia Sinica Institute of Astronomy and Astrophysics (ASIAA), the Inter-University Institute for Data Intensive Astronomy (IDIA) and the National Radio Astronomy Observatory (NRAO). The aim of this new design was to level its architecture to efficiently comply with modern astronomical tools and consequently, improving its scalability with respect to the image input size while maintaining its interactive user experience format.

Similar to CyberSKA [21] viewer, CARTA is based on a client-server architecture whereby the server-side provides high computing power and high data transfer rate on storage whereas the client-side is a browser interface on any form of personal computer or commercial machine that renders the data using GPU acceleration methods [22]. This server-client structure is also beneficial in the sense that it allows connection with multiple clients analyzing the same dataset. When an image is uploaded, CARTA breaks the image data into tiles with distinct resolutions and dynamically delivers the latter depending on the interactions performed by the user on the client-side. Since its latest update, all the functionalities were transformed into plugin frameworks [29] and so, CARTA became more robust and memory efficient as it can load in very large image files rapidly. Furthermore, CARTA allows the user to focus on areas of interest in an image file through a simple 'click and select' on the image viewer. CARTA then updates its profile to provide information on the selected subset of the image. CARTA also incorporates a very user-friendly and customizable GUI to satisfy the needs of its target users. Volumetric rendering is not yet supported by the current system but according to Wang *et al.* [38], the development team is working on evolving CARTA to cater for such advanced feature through efficient visualization with HDF5 images.

3 EXISTING VISUALIZATION TOOLS

This review is mainly focused on how the CARTA system operates but an adequate amount of knowledge and analysis of other contemporary visualization tools is also crucial to avoid duplication of already developed features in the field of astronomy [13]. Other major examples of open-source visualization tools commonly used in astronomy are KARMA, Blender, SAOImage and SlicerAstro.

3.1 KARMA

KARMA is a generic-type visualization tool made up of libraries and packages [11]. These packages have different power levels and are used according to the appropriate functionality required by the programmer. For example, the KARMA library includes a package that astronomers and researchers can use to read data formats such as FITS [39] and MIRIAD [30]. Unlike CARTA, KARMA does not operate on a server-client network but rather, all image inputs must be stored physically on the users' machine for processing. However, this is a major issue with the design as nowadays, the storage capacity of a typical personal computer cannot handle the whole of a standard survey from modern telescopes such as the MeerKAT Radio Telescope [17] which can consume terabytes worth of data. In terms of volumetric rendering capabilities, KARMA includes an XRAY package [1] which can be used to exploit 3D visualization exploration. This technology was developed around two decades ago but was undermined due to the unavailability of fast 3D graphic card and 3D displays. However, in this modern age, the XRAY package should take advantage of high-speed rendering GPUs to achieve its purpose.

3.2 BLENDER

Blender is a software package that supports high-resolution 3D graphics, modelling and animation [20]. In astronomy, the Blender AP is most commonly used to load data PYFITS along with python libraries available for astronomy. Therefore, Blender can act as a framework for astronomical data exploration by rendering volumetric data cubes. Blender uses mesh technology which contains data cubes to generate models representing the type of data that the user wishes to view. Since Blender's use is not only limited to astronomy, it is often utilized in other fields for its cinematic animations due to its expertise in lighting reflection on surfaces, agile camera control and rendering. In parallel, Blender's extensive use also allows it to import and export between multiple data formats such as FITS, JPEG, PNG and GIF which can be then assembled into a video output using FFmpeg utility on a Linux system. Image frames generated from the render process are achieved through the use of multiple threads on a Graphics Processing Unit (GPU) or multiple cores on a single processor and accelerated with Nvidia CUDA [26] and OpenCL [35] given the appropriate hardware.

3.3 SAOIMAGE DS9

SAOImage DS9 is a data visualization application used in astronomy [18]. DS9 supports FITS format images and can perform advanced image manipulation such as tiling, colourmaps, scaling, zoom and rotation through a series of coordinate systems. Similar to CARTA, DS9 support web-based servers and can be used as a local frontend to analyse large data sets. However, as this process is carried out locally, the image files to be analysed needs to be stored on the user's machine and loaded into the main memory. This brings forward a major disadvantage as very large data cubes cannot be used on this application through a typical personal computer. DS9 also has the functionality to bin FITS files into a 3D data cube but it can be viewed one slice at a time along specified axes or as a movie of sequencing images controlled interactively.

3.4 SLICERASTRO

SlicerAstro is one of the extension plugins for the more general-purpose application 3DSlicer [28]. 3DSlicer was originally developed for the visualization and analysis of medical images but was then extended to other fields like astronomy due to its quality visualization capabilities [7]. SlicerAstro is designed to focus on astronomy in terms of functionalities and abilities, especially on neutral hydrogen (HI) cloud and galaxies. The SlicerAstro provides a proper environment for both 2D and 3D visualization as the software allows both CPU and GPU rendering based on the Visualization Toolkit (VTK) [12, 31]. SlicerAstro supports image files of FITS data format and allows interactive controls on the dataset in all three dimensions. This includes generating histograms and profiles on specified subsets of the data. Despite its excellent visualization abilities, SlicerAstro has a major constraint that requires the image file to be stored on the user's machine for processing and needs to be loaded into the main memory. This is considered problematic as modern-day image files generated from sky surveys by modern telescopes are increasing drastically in size and a typical personal computer might not have enough storage capacity to process these new sets of data.

4 VOLUME RENDERING

Volume rendering is a computationally intensive task that involves displaying sampled data in three dimensions [40]. It is used to visualize a 3D sample data set on a 2D display. The data used is in the form of a cube which is made up of stacked slices of 2D images of the same dimensions in a regular pattern. Previous techniques of visualizing 3D data consisted of using computer graphics to convert the volume data sample into a set of geometric primitives which are connected together to form object surfaces contained in the data. This process is called Indirect Volume Rendering [24] and is often used in the game development industry to create 3D objects in an environment. However, this approach cannot be set forward to be implemented in astronomy as in this vast field, all objects do not have well-defined surfaces and this method only renders the surface area of objects, leaving the interior volume of objects hollow. In this situation, astronomy is analogous to the medical Magnetic Resonance Imaging (MRI) scan where, a doctor needs to visualize the data in a 3D format as well as the inside of organs, tissues and tumours [3, 10]. Therefore, this approach is a complete mismatch to astronomical exploration application.

In contrast, Direct Volume Rendering attempts to use the data to construct on a 3D grid by assembling 3D pixels, most commonly known as voxels [15]. Voxels are each made up of an array of elements of volume that forms a notional 3D space. These elements comprise of an opacity and a colour value which is calculated by the Red Green Blue Alpha (RGBA) transfer function [33] and maps a distinct numerical value from the opacity and colour of each voxel. This approach is computationally more expensive but fits in the astronomical exploration context as the interior volumes are also rendered and the output has a better signal-to-noise ratio than using other previously mentioned approaches. One of the most effective ways to implement direct volume rendering is through a process called ray casting.

4.1 RAY CASTING

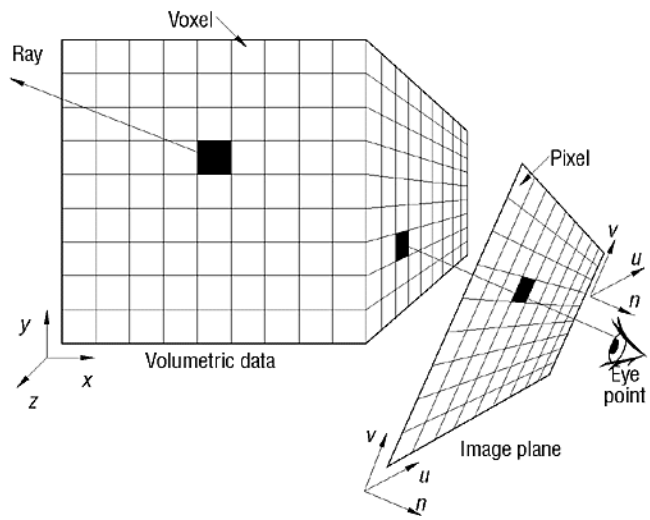

Figure 2: A visual representation of the ray casting process whereby a ray traverses through each voxel until the final image is produced on the 2D plane. Retrieved from: https://doi.ieeecomputersociety.org/10.1109/4434.656777

Ray casting is a process which computes the projection of a volume of data onto a 2D viewing display [14, 22]. For every pixel, ray casting computes a ray originating from that pixel to project it into the data volume. The ray accumulates an aggregate colour and opacity which is then assigned to that pixel. Since this operation is computationally expensive when considering a large volume of data, ray casting must be conducted in parallel. Therefore, ray casting is best executed with a GPU. Nowadays, with the advent of technology, GPUs are much more powerful in terms of parallel executions than CPUs. GPUs can now be utilized as massively parallel, general-purpose computational co-processors using software development kits like CUDA [26] and OpenCL [35]. Kaehler *et al.* [19], Becciani *et al.* [2], Szalay *et al.* [36] and Jin *et al.* [16] all elaborated on or carried out experiments to demonstrate the prime advantages of using GPUs to render information in the astronomical space in different setup and parameters. For example, Szalay *et al.* [36] observed a speed up of 23% when GPUs were used for visualization and rendering. Hassan *et al.* [15] took it a step further by experimenting with different computer systems and GPUs as well as different volume data forms. The architecture used for optimizing results was to use a cluster of interconnected nodes representing GPU card and CPU cores, a server machine for task-scheduling and a client machine which is a station for viewing the output. This allows the client with limited resources and computing power to still run and interact with the application as very little processing is required from the client.

3D visualization is best suited for data exploration for astronomers but the final output, however, is displayed on a 2D plane. This issue was resolved through the use of Virtual Reality (VR) tools as they can provide a unique and immersive experience of the data [8]. In December 2020, Marchetti *et al.* [25] announced the iDaVIE-v project which is an extension of the IVL iDaVIE [34] software. iDaVIE-v is a data visualization software which supports virtual reality on any commercial VR headsets currently on the market and different operating systems such as Microsoft Windows, MacOS and Linux. The primary goal of developing iDaVIE-v was to analyze Neutral Hydrogen (HI) Radio data cubes in virtual reality and to be able to perform intuitive interactions with the elements in the environment.

4.2 OPTIMISATIONS

Volume rendering takes a lot of computing power to execute especially for large data files produced by radio telescopes like Australia's ASKAP and South Africa's MeerKAT. Despite the multithreading process to execute volume rendering, some optimizations need to be performed for a better real-time interactive application. The primary purpose of optimization is to skip as much volume as possible without creating a significant on the actual representation of the data. In medicine, especially CT scans or MRIs, one of the most common ways of optimization is empty space skipping and early ray termination [32]. This process includes identifying and avoid rendering transparent regions of the volume as well as those regions which are blacked out when a ray travels through a sufficiently dense material. However, this technique involves skipping voxels which also means losing some information. Consequently, the output file is not of the best possible quality.

Better alternatives that can solve the issue of information loss are volume segmentation and pre-integrated volume rendering. Volume segmentation is a procedure which sections a smaller volume out of the entire dataset. Volume rendering is then performed on the selected volume only which is the actual focus of analysis while the remaining piece is deemed uninteresting or can be analyzed at some other point in time. On the other hand, pre-integrated volume rendering [6, 23] is an operation which pre-computes the required data before attempting volume rendering. The pre-processing done on the data actually accelerates the rendering process and also improves overall quality without affecting performance.

5 DATA FORMATS

In astronomy, relevant information can be stored in files of different data formats. These data formats each store information in a different way which can alter performance and efficiency of the whole visualization and data exploration process. The most commonly used ones nowadays are FITS and HDF5 formats.

5.1 FITS FORMAT

The FITS data format originally standardized in 1981, was formularized in such a way that it is very generic and therefore, it can represent a large amount of astronomical data [39]. The original FITS file format contained two sections, namely a human-readable header and the data component. The header provides relevant information about the data embedded in the files while the data component contains the actual real-world data. This data has to be correctly interpreted by software applications to view and manipulate the information presented. Initially, the FITS format was only capable of recording a single set of multi-dimensional data which was a major issue as multiple FITS files were needed to view multiple sets of related multi-dimensional data. Fortunately, this concern was soon tackled with by the use of extensions [27]. Multiple extensions such as IMAGE, TABLE and BINTABLE were implemented to resolve some of the issues with the format. With the aid of extensions, FITS file was able to record a plethora of multi-dimensional datasets as well as organize data in a table form to allow more flexibility and efficacy of storing the data structures. However, the FITS format stores data products in a sequential pattern and therefore, modern technology such as GPUs cannot take advantage by processing the file in a concurrent fashion and achieve faster execution.

5.2 HDF5 FORMAT

The HDF5 format is a more recent model designed to effectively eliminate the issues with ancient formats like FITS [9]. The data model of HDF5 does not conform to the simplicity of a relational model but rather installs the data products in a hierarchical structure. An HDF5 file supports an unlimited variety of datatypes and is intended as an efficient input file for complex and large volume data. HDF5 datasets contain array variables whose elements are structured as a multi-dimensional array. HDF5 also includes the rank and the maximum extent of the data which represents the number of dimensions involved and the totality of the data elements, respectively. The HDF5 format can perform chunking of the dataset and thereafter, adheres to pipelining which is applied to each chunk of the HDF5 dataset to improve. Therefore, HDF5 allows users to exploit parallelism while using a single global semaphore for internal data protection in order to achieve better input and output performances. Comrie *et al.* [5] developed an HDF5 schema to demonstrate better efficiency of using the HDF5 format compared to the FITS format on SKA-scale image cube visualization. The paper successfully demonstrated that large HDF5 files take less time to load in CARTA than FITS files of similar sizes. It also highlighted the significant increase in channel animation frame rate when using HDF5 files animation compared to the traditional FITS file animation.

6 CONCLUSIONS

In this literature review, some of the existing data visualization tools used in astronomy were analysed in terms of their functionalities and capabilities. Modern visualization techniques such as volume rendering were also reviewed as recent developments in radio astronomy telescopes demand more effective and faster computation for analysing the large data cubes produced. The different optimisations which can be implemented to the volume rendering process and their impact on overall performance were outlined. Lastly, the data formats of the input and output files used for data visualization also demonstrated to have an influence on performance as well. Consequently, the entirety of this review needs to be taken into consideration when developing or upgrading a data visualization tool to comply with modern technological systems.

In the light of the above, an interactive 3D data visualization is achievable with CARTA due to its client-server architecture which allows a client with limited resources to view and interact with the data on a browser whereas the server does all the heavy computations ideally by a cluster of GPUs for faster quality rendering. The rendered images are then compressed and transferred over to the client for display. Optimizations such as volume segmentation and pre-integrated volume rendering could be a great integration as it improves the execution time without reducing the quality of the image produced. Finally, the HDF5 file format would add to the overall performance gain due to its hierarchical structure which can be accessed in parallel. All these improvements would greatly contribute in enabling the application to be interactive as the operations are performed in real-time or near real-time.

REFERENCES

[1] Barnes, D. et al. 2006. An Advanced, Three-Dimensional Plotting Library for Astronomy. Publications of the Astronomical Society of Australia. 23, 2 (2006), 82-93.

[2] Becciani, U. et al. 2010. VisIVO - Integrated Tools and Services for Large-Scale Astrophysical Visualization. Publications of the Astronomical Society of the Pacific. 122, 887 (2010), 119.

[3] Borkin, M. et al. 2007. Application of Medical Imaging to the 3D Visualization of Astronomy Data. Proceedings of IEEE Visualization Conference. (2007).

[4] Camilo, F. 2018. African star joins the radio astronomy firmament. Nature Astronomy. 2, 7 (2018), 594-594.

[5] Comrie, A. et al. 2020. Development and application of an HDF5 schema for SKA-scale image cube visualization. Astronomy and Computing. 32, (2020), 100389.

[6] Engel, K. et al. 2001. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware - HWWS '01. (2001).

[7] Fedorov, A. et al. 2012. 3D Slicer as an image computing platform for the Quantitative Imaging Network. Magnetic Resonance Imaging. 30, 9 (2012), 1323-1341.

[8] Ferrand, G. et al. 2016. 3D VISUALIZATION OF ASTRONOMY DATA CUBES USING IMMERSIVE DISPLAYS. (2016).

[9]Folk, M. et al. 2011. An overview of the HDF5 technology suite and its applications. Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases - AD '11. (2011).

[10] Fuchs, H. et al. 1989. Interactive visualization of 3D medical data. Computer. 22, 8 (1989), 46-51.

[11] Gooch, R. 1996. Karma: a Visualization Test-Bed. Astronomical Society of the Pacific Conference Series. 101, (1996), 80.

[12] Hanwell, M. et al. 2015. The Visualization Toolkit (VTK): Rewriting the rendering code for modern graphics cards. SoftwareX. 1-2, (2015), 9-12.

[13] Hassan, A. and Fluke, C. 2011. Scientific Visualization in Astronomy: Towards the Petascale Astronomy Era. Publications of the Astronomical Society of Australia. 28, 2 (2011), 150-170.

[14] Hassan, A. et al. 2011. Interactive visualization of the largest radioastronomy cubes. New Astronomy. 16, 2 (2011), 100-109.

[15] Hassan, A. et al. 2012. A Distributed GPU-Based Framework for Real-Time 3D Volume Rendering of Large Astronomical Data Cubes. Publications of the Astronomical Society of Australia. 29, 3 (2012), 340-351.

[16] Jin, Z. et al. 2010. High-performance astrophysical visualization using Splotch. Procedia Computer Science. 1, 1 (2010), 1775-1784.

[17] Jonas, J. 2018. The MeerKAT Radio Telescope. Proceedings of MeerKAT Science: On the Pathway to the SKA — PoS(MeerKAT2016). (2018).

[18] Joye, W. and Mandel, E. 2003. New Features of SAOImage DS9. Astronomical data analysis software and systems XII. 295, (2003), 489.

[19] Kaehler, R. et al. 2006. GPU-Assisted Raycasting for Cosmological Adaptive Mesh Refinement Simulations. Volume Graphics. (2006), 103-110.

[20] Kent, B. 2013. Visualizing Astronomical Data with Blender. Publications of the Astronomical Society of the Pacific. 125, 928 (2013), 731-748.

[21] Kiddle, C. et al. 2011. CyberSKA. Proceedings of the 2011 ACM Workshop on Gateway Computing Environments-GCE'11. (2011), 65.

[22] Kruger, J. and Westermann, R. 2003. Acceleration techniques for GPU-based volume rendering. IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control. (2003).

[23] Lum, E. et al. 2004. High-Quality Lighting and Efficient Pre-Integration for Volume Rendering. UC Davis: Institute for Data Analysis and Visualization. (2004).

[24] Lurig, C. et al. Deformable surfaces for feature based indirect volume rendering. Proceedings. Computer Graphics International (Cat. No.98EX149).

[25] Marchetti, L. et al. 2020. iDaVIE-v: immersive Data Visualization Interactive Explorer for volumetric rendering. (2020).

[26] Pawan, H. and Narayanan, P. 2007. Accelerating large graph algorithms on the GPU using CUDA. International conference on high-performance computing. (2007).

[27] Ponz, J. et al. 1994. The FITS image extension. Astronomy and Astrophysics Suppl.. 105, (1994), 53-55.

[28] Punzo, D. et al. 2017. SlicerAstro: A 3-D interactive visual analytics tool for HI data. Astronomy and Computing. 19, (2017), 45-59.

[29] Rosolowsky, E. et al. 2015. The Cube Analysis and Rendering Tool for Astronomy. (2015).

[30] Sault, R. et al. 1995. A Retrospective View of MIRIAD. Astronomical Society of the Pacific Conference Series. 77, (1995), 433.

[31] Schroeder, W. et al. 2000. Visualizing with VTK: a tutorial. IEEE Computer Graphics and Applications. 20, 5 (2000), 20-27.

[32] Sherbondy, A. et al. Fast volume segmentation with simultaneous visualization using programmable graphics hardware. IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control.

[33]Shiaofen Fang et al. 1998. Image-based transfer function design for data exploration in volume visualization. Proceedings Visualization '98 (Cat. No.98CB36276). (1998), 319-326.

[34] Sivitilli, A. et al. 2021. Virtual Reality and Immersive Collaborative Environments: the New Frontier for Big Data Visualization. (2021).

[35] Stone, J. et al. 2010. OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems. Computing in Science & Engineering. 12, 3 (2010), 66-73.

[36] Szalay, T. et al. 2008. GPU-Based Interactive Visualization of Billion Point Cosmological Simulations. (2008).

[37] Taylor, R. 2015. Frelled: A realtime volumetric data viewer for astronomers. Astronomy and Computing. 13, (2015), 67-79.

[38] Wang, K. et al. 2020. CARTA: Cube Analysis and Rendering Tool for Astronomy. Astronomical Society of the Pacific Conference Series. 527, (2020), 213.

[39] Wells, D. and Greisen, E. 1979. FITS - a Flexible Image Transport System. (1979).

[40] Westover, L. 1989. Interactive volume rendering. Proceedings of the 1989 Chapel Hill workshop on Volume visualization - VVS '89. (1989).