Cavern Halos: Exploring Spatial and Nonspatial Cosmological Data in an Immersive Virtual Environment

Peter Hanula, Kamil Piekutowski, Carlos Uribe, Kyle Almryde,
Arthur Nishimoto, Julieta Aguilera, and G. Elisabeta Marai

Fig. 1. Collaboratively examining several halos and their substructures in the immersive environment.

Abstract - We present the design and implementation of an immersive visual mining and analysis tool for cosmological data. The tool consists of an immersive linked multiview display which allows domain experts to interact with visual representations of spatial and nonspatial cosmology data. Nonspatial data is represented as time-aligned merger trees, and through a pixel-based heatmap. Spatial data is represented through GPU-accelerated point clouds and geometric primitives. The user can select a halo and visualize a 3D representation of the raw particles, as well as of the halos at the particular time stamp. We have demonstrated the tool to a senior staff member of the Adler Planetarium and report their feedback. The tool can assist researchers in the interaction navigation and mining of large scale cosmological simulation data.

INTRODUCTION

Thanks to increasing advancements of high performance computing, cosmologists are able to model the formation of the universe via n-body simulation, from the Big Bang to the present. These simulations make up a core part of our understanding of the known universe. However, the volume of data generated by these simulations is enormous, with the largest simulation to date [12], having produced data on the order of petascale. The task of analyzing and disseminating this level of data has, unsurprisingly, proved to be a major challenge. The need for visualization tools which can accurately represent the complex interactions of these simulations in addition to scaling has only grown.

Major challenges involved in addressing this visual analysis problem include scalability, access to distributed data sources, integration of spatial and non-spatial data, and even data processing procedures such as ensuring that derived spatial data points are correctly aligned.

In this paper we present a novel approach towards the visualization of large-scale, n-body cosmological simulations of Dark Matter Particle and Halo data. We incorporate an immersive linked multiview display to facilitate the interactive exploration by domain experts of both spatial and nonspatial cosmological data. To that end, we outline our approaches to the following tasks: T1) Data integration and browsing; T2) Halo identification and visualization; and T4) Diving Deep into Halo Substructure.

1 RELATED WORK

Data visualization has been established as a tool for exploration and analysis of complex, large scale datasets. In particular,

---

{phanul2, kpieku2, curibe2, kalmry2, anish2,
gmarai}@uic.edu Electronic Visualization Laboratory
University of Illinois at Chicago

julietina@me.com
Adler Planetarium, Chicago,
IL

IEEE Scientific Visualization Conference (SciVis) 2015
25-30 October, Chicago, Il, USA
978-1-4673-9785-8/15/$31.00 ©2015 IEEE

previous works have visualized the Millennium Simulation Project's 10 billion dark matter model of the expansion of the universe. Due to the size of the Millennium dataset, many visualizations utilize snapshots of the data or pre-rendered animations [12]. Other approaches utilize multi-GPU mesh deformation to represent gravitational forces [11]. Continuous level of detail of point clouds have also been used to interactively explore billions of particles [10]. No reported works visualize the halo, particle, and merger tree data.

2 METHODS

We follow an immersive, linked views approach to visualize the cosmology data. We use a next generation CAVE2 [1] immersive environment, and the D3 API [2] and OmegaLib [3] framework for virtual reality to display tree data, respectively 3D particles and halos. The D3 nonspatial views are projected into the immersive environment. The spatial and nonspatial views are linked through interaction and a communication channel between D3 and OmegaLib.

The entry point of our application is a pixel heatmap of the merger tree forest, i.e. the collection of all merger trees in the simulation data. The heatmap intensities can be mapped to specific characteristics of each tree -- for example, the tree depth, breadth, the size or mass of the largest halo in the tree, or the end timestamp of each tree. From this overview of the tree data, the user can select a particular tree of interest.

A second view shows a 2D representation of the selected tree, with time mapped on the horizontal axis. From here, the user can select a particular time step, or a time lapse interval, and immersively explore, in the third view, the corresponding 3D particle set and halos. The time lapse visualization can be used to show all the merger-trees in 3D space.

Alternatively, the user may start directly with the immersive exploration of a particular timestep or time lapse interval and browse the corresponding 3D particle set and halos, then navigate to the merger tree data. We describe the initial data integration as well as each data encoding in the sections below.

2.1 Initial Data Integration and Browsing

Data Ingesting We used the 11G 128^3 particle dataset provided through the IEEE SciVis 2015 website. In the pre-processing stage, we ran a python [4] script to parse the raw particle and halo catalog datasets, using yt [5] and thingking [6] respectively. After integration and processing (described below), the spatial particle and halo data are displayed and manipulated in our immersive environment using OmegaLib [3].

To read merger tree data we wrote a custom parser. To enable better data flexibility, we also wrote an algorithm to create a proper JSON format [7]. In our approach, the branches expand backwards with respect to time: the algorithm starts at the final stage of the halo creation and moves back in time by recursively searching for the parents of each node.

Data Integration and Processing The raw particle data and the halo data are specified in different coordinates, and need to be cross-registered before displaying. For the raw particle dataset, we used a modified algorithm to convert the proper kpc coordinates to Mpc/h comoving, as follows:

$$pos\_cMpc = (pos\_kpc + width/2.0) * h\_100 * kpc\_to\_Mpc / cosmo\_a + redShift * Delta$$

where *pos_kpc* represents the position coordinate to convert, *width* represents the simulation width, *h_100* represents the Hubble parameter, *kpc_to_Mpc* represents the conversion rate, *cosmo_a* represents the scale factor for a particular snapshot, *redShift* represents the redshift value for a particular snapshot, and *Delta* = -31.2715 is an offset constant specified in the sample dataset.

After parsing the necessary initial parameters (XYZ coordinates, acceleration, velocity, and potential for raw particle data; XYZ coordinates, mass, radius, and circular velocity for halo data), we converted the data to the format used by OmegaLib [3]. The Point Cloud module of OmegaLib was used to load and handle point cloud data. We extended the module to accommodate the parsed parameters mentioned above, and the opacity field, alpha, was replaced with an identifier to differentiate raw particles from halos. In total, 178 files were generated (89 for the raw particle dataset + 89 for the halo dataset), giving two files per timestep ranging from the twelfth timestep to the hundredth. These files were then loaded into memory in order to be further processed by our python program.

To take full advantage of GPU acceleration, we further normalized particle parameters such as potential, acceleration, or velocity to a common range of [0:1,000,000]. For the Halo dataset we normalized the Halo Mass, Circular Velocity, and other parameters to a shorter range of [0: 300].

Data Browsing When the application starts, the 100th timestep for the particle and halo data is loaded and displayed by default. Other timesteps can be loaded instantly. The merger forest is also displayed, as well as the first tree in this forest. Browsing, 3D viewing, zooming, filtering, and selection across the spatial and nonspatial views are enabled via the D3 API, respectively OmegaLib.

2.2 Particle Data, Halo and Merger Tree Visualization

Particles and Halos The raw particle and halo data are displayed as point clouds, respectively as transparent spheres. The sphere radius is mapped to Rvir (Halo radius), a parameter provided by the Halo catalog dataset. The raw particles are color-mapped by potential (an indicator of the power of a halo), while halos are colored a semi-transparent white, which enables them to stand out. The halo transparency further allows users to examine the particle substructure of a halo. Both the particle data and the halo data can be toggled off or on.

Users can choose between three color schemes: 1) cyan to purple particles with yellow halos (focus on general flow from a min to max data set for raw particles, with a clear contrast of halos), 2) gray-scale particles with red halos (focus on halos and on the particles that make up the halo), or 3) multi-hue sequential colored particles with white halos (easier to distinguish the different ranges of data sets, and easier to examine the substructure of a halo). The third color scheme is the most clear and it is displayed by default.

The raw particle dataset can also be color-mapped by velocity and acceleration, while halos can be colored by any parameter specified by the user via an interactive menu. By default, all halos are the same color, making it easier to distinguish between particles and halos. However, when the particles are toggled off, the halo colors can be mapped to halo quantities of interest, for example, the halo mass.

Time-lapse Visualization We further implemented a 3D time-lapse function, where the user can specify a start timestep, an end timestep, and the step size. The application overlaps the selected timesteps to show the flow and path of the halos and/or particles over time. The time lapse creates a static 3D representation of the merger trees. The representation can also be animated to show the halo formations at each timestep. While the animation is playing, the user can freely move through the environment and zoom in on a desired halo formation.

We use the OmegaLib's GLSL shaders to render points. Because we use GPU-acceleration, the timesteps can be rendered instantly to help better visualize the data. The GLSL approach should also help when working with larger datasets.

Time-aligned Tree Visualization The 3D tree representation above can give users a good sense of space relationships. However, the trees are also time-aligned. The time aspect can be difficult to capture via composited static renderings or via animations. The sheer number of trees can also get quickly overwhelming.

To circumvent these issues, and to further facilitate the navigation of the tree forest, we have created two additional encodings for the merger trees. The first is the pixel heatmap described earlier. The second encoding is a horizontally aligned graph representation of the tree data, in which halos are represented by nodes, and merging or splitting operations are represented by arcs. These D3 nonspatial representations are linked to the spatial representations via the SAGE2 library, which allows us to run large screen applications in the immersive environment.

Exploring the Trees and Halo Substructure Exploration of the tree data and of the halo substructure is enabled through interaction with the representations above. Selecting a tree from the heatmap overview generates the time-aligned view of that tree. Selecting a halo in the tree loads the corresponding timestep of the 3D view, and allows the user to navigate instantly to the corresponding halo. The 3D view enables the exploration of the halo particle substructure, which is visible through the transparent halo shell.

3 EVALUATION AND RESULTS

Experimental setup and results This application was designed under the CAVE2 [1] hybrid immersive environment, which consists of 72 stereoscopic displays driven by a 36 node cluster. Each node has 64 GB of system memory and a nVidia GTX 680 graphics card.

For this application we have used a 11 GB sample dataset. We were able to load 89 timesteps for raw particle data, and 89 timesteps for halos (starting at timestep 12, since the first 12 timesteps are very similar). We have used the 170 MB particle dataset, which includes 2M particle points per timestep in addition to the halo tree data; the number of trees ranges from 63 to 7K. Loading the data, including the startup for the OmegaLib application, takes about a minute. Raw particle data requires about 400 - 550 ms longer to load than the halo data.

Our application attains a reasonable rate of 60 fps. Running intensive tasks, such as the time lapse function where multiple timesteps can be overlapped, can lead to a drop down to 20-30 fps. If we toggle the raw particle dataset off, and leave the halo dataset on, the application attains again 60 fps, even when every timestep is turned on.

Domain expert feedback To further evaluate the usefulness of our visual encodings and of the overall application, we have demonstrated the tool to a senior domain expert from the Adler Planetarium, who has significant experience in immersive environments. Figures from the analysis reported below are included at the end of this manuscript, as well as in the accompanying video.

With respect to the particle set and halo visualization, the expert found the visual encodings were effective and ``pretty good''. The interaction and flow were found to be ``very smooth''. The expert remarked the density and distribution of particles inside halos ``showed well the power of the halo''. The expert further appreciated the ability to analyze the relationship between mass and size by turning the particles off, since ``halos could be very compact and still have high mass``.

The expert was particularly interested in merging halos, and found the time lapse visualization useful in that respect. The expert further suggested a non-spherical encoding for the halos, since the halo shape is probably ``more amoeba-like'', as well as explicitly encoding the ``flow'' direction for the time lapse visualization. The expert was surprised the simulation did not include temperature data, and concluded this was a cold dark matter simulation. With respect to the merger trees, the expert expressed preference for the 2D, time-aligned representation of the tree, as opposed to an alternative encoding which traced the tree paths in 3D space.

Overall, the domain expert was impressed with the application, which was found to be "very nice'', and was keen to show it to colleagues at the Planetarium. Navigation in the virtual environment came natural. The ability to examine details within context without getting lost, despite the scale of the data was appreciated. Finally, the expert brainstormed for ways to port the application to the Adler Planetarium, where OmegaLib is already installed and a 3D display is in use.

4 DISCUSSION AND CONCLUSION

Our results and domain expert evaluation show our halo visualization application is useful and effective, despite the large scale of the data. The tool integrates smoothly spatial and nonspatial encodings to enable data exploration. Immersive visualization further enables users to navigate smoothly these large datasets.

In terms of scalability and portability, our approach benefits from the computing hardware behind the CAVE environment. However, we did not attempt datasets larger than the sample 11GB dataset. Similarly, we did not explicitly tackle the multi-dimensional challenges of the multi-variate parameters associated with each halo.

In conclusion, we introduced a novel immersive tool for the exploration of large scale cosmological simulation data. Our approach integrates spatial and nonspatial data through a linked views approach, and supports implicit whole body navigation in this large space.

ACKNOWLEDGMENTS

We thank Jillian Aurisano, Alessandro Febretti, Luc Renambot, Lance Long and the entire Electronic Visualization Lab staff for the many visualization tutorials this summer, and for their endless patience and support. This work was supported by NSF CAREER IIS-1541277, by NSF IIS-1340866 (CAREER Research Experiences for Undergraduates supplement) and by a CRA-W REU award. The development of the CAVE2 and OmegaLib was supported by NSF awards CNS-0821121 and CNS-0959053. We thank Galen Ballew for his help with photo and video editing.

REFERENCES

[1] A. Febretti, A. Nishimoto, T. Thigpen, J. Talandis, L. Long, J. D. Pirtle, T. Peterka, et al. "CAVE2: A Hybrid Reality Environment for Immersive Simulation and Information Analysis." In *IS&T/SPIE Electronic Imaging*, 864903–864903. International Society for Optics and Photonics, 2013, http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1660803.

[2] M. Bostock, J. Heer, V. Ogievetsky, and community, Data-Driven Documents, last stable release 2015. Available at https://github.com/mbostock/d3/wiki

[3] A. Febretti, A. Nishimoto, V. Mateevitsi, L. Renambot, A. Johnson, and J. Leigh, "Omegalib: A multi-view application framework for hybrid reality display environments," in *Virtual Reality (VR), 2014 iEEE*, 2014, pp. 9–14.

[4] G. van Rossum and F.L. Drake (eds), Python Reference Manual, PythonLabs, Virginia, USA, 2001. Available at http://www.python.org

[5] M. Turk, B. Smith, J. Oishi, S. Skory, S. Skillman, T. Abel, M. Norman,'yt: A MULTI-CODE ANALYSIS TOOLKIT FOR ASTROPHYSICAL SIMULATION DATA', The Astrophysical Journal Supplement Series, 192:9 (16pp), January 2011

[6] Matthew J. Turk, Samuel W. Skillman and Michael S. Warren (2014). ThingKing: Memory Mapping the World Wide Web. Zenodo. [doi:10.5281/zenodo.10773]http://dx.doi.org/10.5281/zenodo.10773

[7] ECMA International,The JSON Data Interchange Format 1st Edition, France, 2013. http://www.ecma-international.org/publications/standards/Ecma-404.htm

[8] V. Springel, S. D. White, A. Jenkins, C. S. Frenk, N. Yoshida, L. Gao, J. Navarro, R. Thacker, D. Croton, J. Helly, and others, "Simulating the joint evolution of quasars, galaxies and their large-scale distribution," *arXiv preprint astro-ph/0504097*, 2005.

[9] A. Febretti, K. Richmond, P. Doran, and A. Johnson, "Parallel Processing and Immersive Visualization of Sonar Point Clouds," in *IEEE Symposium on Large Data Visualization*, Paris, France, 2014.

[10] R. Kaehler, O. Hahn, and T. Abel, "A novel approach to visualizing dark matter simulations," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 12, pp. 2078–2087, 2012.

[11] R. Fraedrich, J. Schneider, and R. Westermann, "Exploring the Millennium Run - Scalable Rendering of Large-Scale Cosmological Datasets," *IEEE Transactions on Visualization & Computer Graphics*, vol. 15, no. 6, pp. 1251–1258, Dec. 2009.

[12] S. Skillman, M. Warren, M. Turk, R. Wechsler, D. Holz, P. Sutter, "Dark Sky Simulations: Early Data Release", arXiv preprint arXiv:1407.2600, 2014.

[13] A. Febretti, A. Nishimoto, V. Mateevitsi, L. Renambot, A. Johnson, and J. Leigh, "Omegalib: A Multi-View Application Framework for Hybrid Reality Display Environments." In *Virtual Reality (VR), 2014 iEEE*, 9–14. IEEE, 2014, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6802043.

Figures:

P1020728.jpg
Pixel heatmap (top) and time-aligned tree representation (below). Each heatmap square represents a tree in the tree forest. In the tree representation, nodes represent halos, and the arcs represent merging and splitting operations.

P1020720.jpg
Immersive examination of the substructure of the biggest halo at time step 100. Halos are represented as transparent white spheres, and particles as color-mapped point clouds.

P1020639.jpg
Collaboratively examining several halos and their substructures in the immersive environment.

P1020627.jpg
The last timestep in the Halo data set. A red-blue color map encodes the mass of each halo.

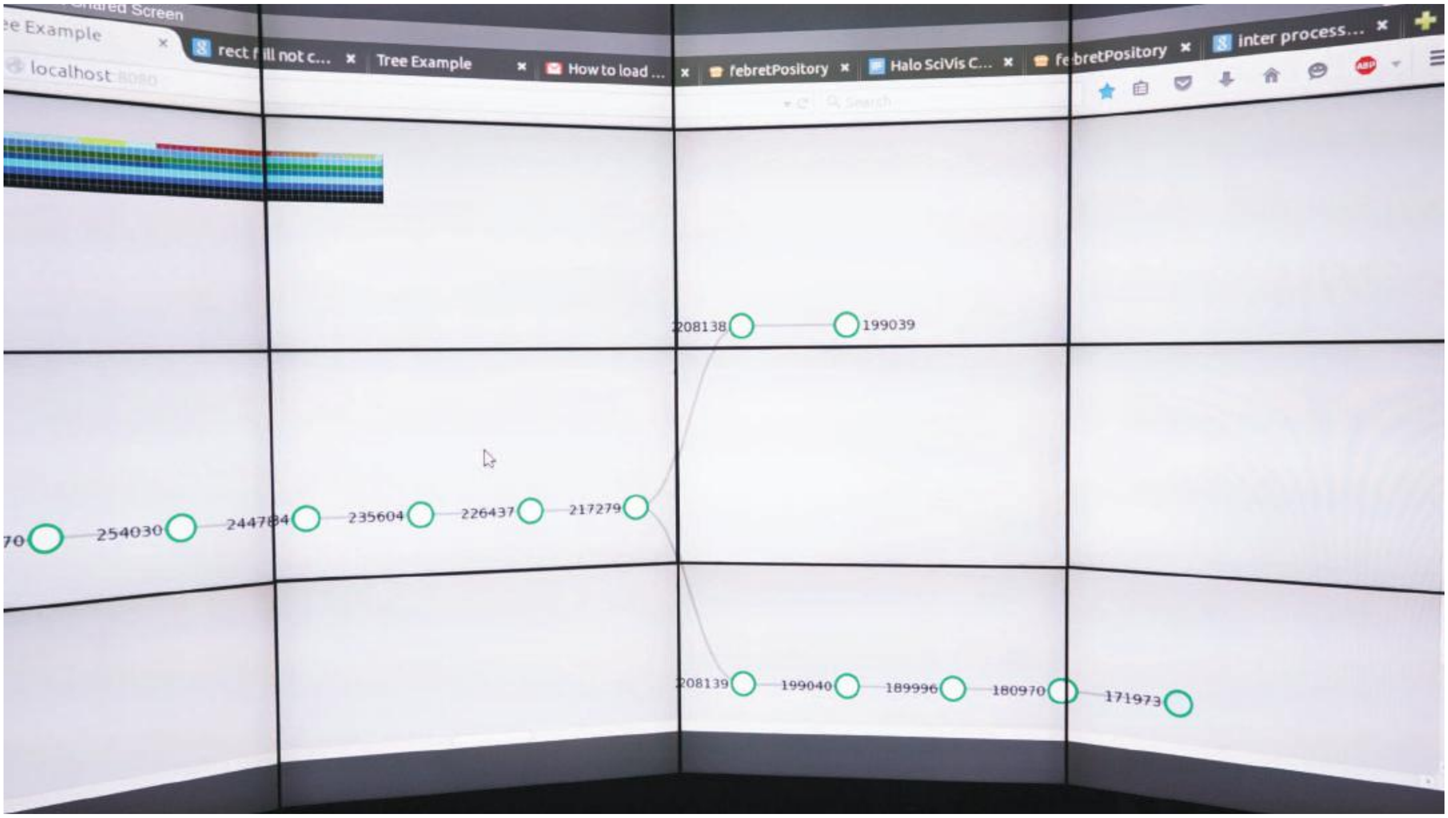
P1020574.jpg
Time lapse visualization of 89 timesteps, which shows the paths taken over tie by halos. Note how halos merge, get created, or disappear through time.

P1020568.jpg
Halos merging together to form a giant halo in a time lapse visualization of 89 timesteps. Note the halos in the lower right hand corner that were created halfway through the timestep sequence and do not merge, but move linearly through time.

P1020554.jpg
Zoom-in operation on a massive halo, along with the smaller halos surrounding it. Note the halo substructures, along with the raw particles that surround the halos. Halos are a transparent white, while raw particles are colored by potential.

P1020551.jpg
Zoomed-out view of many halo formations at once (all timesteps). With 3D glasses, we can see depth of the halos and the paths they form, as well as where halo formations start and eventually end. Each tile of the immersive display is the size of a regular desktop display, indicating how little information and context could fit on a single display.

P1020470.jpg
Single timestep view of several bigger halos which are starting to cluster.


Shared Screen
ee Example
rect fill not c...
Tree Example
How to load ...
febretPository
Halo SciVis C...
febretPository
inter process...
localhost
70
254030
244784
235604
226437
217279
208138
199039
208139
199040
189996
180970
171973