COSMIC EXPLORER: A VIRTUAL REALITY ENVIRONMENT FOR EXPLORING COSMIC DATA

Deyang Song and Michael L. Norman
National Center for Supercomputing Applications
605 E. Springfield Ave.
Champaign, IL 61820

March 19, 1993

Abstract

Supercomputer simulations now can produce multi-level, multi-scale, large data sets that require new techniques in scientific visualization and higher levels of hardware performance. We are exploring the use of VR technology in this connection. Cosmic Explorer is a virtual reality environment for visualizing numerical and observational cosmology data. We have implemented multi-scale visualization techniques that will work on large, multi-level time-dependent data sets. Virtual reality provides natural ways to navigate in an immersive environment, and scientific visualization requires overviewing. Our techniques bridge the two in a VR system involving the BOOM and the DataGlove. Combining modern visualization hardware, software, and VR technology, we are able to create a system that lets users explore the virtual space created by numerical simulations with ease and naturalness.

1 Introduction

Cosmic Explorer is a Virtual Reality (VR) environment for scientists to visualize in realtime their time-dependent cosmology data generated from either numerical simulations [5] or observations [7]. It allows scientists to interact with their data and to navigate in the virtual space they create. The Cosmic Explorer is motivated by Carl Sagan's imaginary spaceship in the PBS series *Cosmos*, in which he explores the far corners of the universe. In the Cosmic Explorer, the user can explore the formation of the galaxies, clusters, and superclusters by means of numerical simulations and VR technology. The numerical simulations produce very large data sets representing the cosmic structures and events [5, 9]. It is important for the scientist not only to be able to produce images from this data but to be able to animate events and view them from multiple perspectives.

There are two fundamental properties of a VR system: immersion, the illusion of being inside a computer-generated world, and navigation, the ability to move around inside the virtual world [13]. The Cosmic Explorer demonstrates human-oriented paradigms for examining large data sets that contain spatial and temporal information. By allowing a human to explore these data sets as they might explore a physical place, a user can take advantage of human experience rather than technological prowess to comprehend the data.

There are several successful applications of VR to scientific visualization. Most noticeably are several research projects at UNC [1] and NCSA [15], NASA's Virtual Windtunnel [2, 3],

and EVL's CAVE [4]. These systems have shown beyond doubt the power of VR technologies. However, scientific visualization poses a difficult condition for VR applications. VR applications require realtime graphics update rate (the rendering speed) in response to the changes of user's viewpoint. But scientific data sets are often enormously large (many gigabytes), and take up a large percentage of the CPU time and memory space. The gap between the strict requirement of realtime response by the VR environment and the time and memory consuming visualization systems makes it difficult to apply VR technology to scientific visualization. For example, the time-dependent galaxy formation data set we used had a size of 300 megabytes. New simulations are generating even larger data sets. So it is impossible to cache all the data in main memory. Furthermore, with the advancement in numerical technologies, simulations now can generate multi-level data whose scales span orders of magnitude in space and time. The Cosmic Explorer has attempted to tackle these problems using various techniques.

Brooks in [1] suggested using progressive refinement techniques to deal with large data set and enforce the realtime requirement of the VR system. In Cosmic Explorer, we use a selective refinement technique that also proves useful. We keep a down-sampled data set in memory and load in individual frames (each frame contains data for one time step) from the full data set on demand. Animating the down-sampled smaller data set gives an overview of the simulation results. In our case, the user can observe how galaxies form, move, and cluster together. During the animation, the user can interactively select a slice or a sub-volume within which full-resolution data is visualized. In this way the user can closely inspect the local properties of the original full data set.

Another experiment we did with the Cosmic Explorer is the multi-scale visualization techniques. We have developed multi-level numerical simulations based on nested grids [12], which produce data sets that are hierarchical and span multiple scale levels in space and time. For example, at the highest level, we can simulate the galaxy cluster formation. At a lower level, we can simulate the formation of an individual galaxy. It is a new challenge for visualization systems to provide effective ways for scientists to navigate in such a multi-level virtual space. Our experiment with the VR systems shows that VR provides a good solution to these problems.

Section 2 gives a brief overview of the hardware and software components of the Cosmic Explorer, followed by discussions of some salient features that the Cosmic Explorer provides in dealing with large data sets. Section 3 discusses some implementation issues related to VR and presents future work on our system.

2 Architecture and Main Features

The Cosmic Explorer consists a Silicon Graphics workstation, a stereo viewing device called BOOM (Binocular Omni-Orientational Monitor) from Fake Space Labs [10], a DataGlove from VPL Research [19], and a Polhemus 3Space tracker. The BOOM is used as an output device. It produces stereo images and gives the viewer an illusion of being in a three-dimensional space. Its position and orientation also indicates the user's viewpoint. The DataGlove is used by the user to interact with the Cosmic Explorer. The Polhemus tracker is attached to the DataGlove and is used to track the position of the DataGlove.

The software system has five components: the data manipulation part, the hand interaction part, the viewpoint control part, the particle displaying part, and the animation control part. The data manipulation part loads the selected time steps of down-sampled particle data into the memory, and color-encodes the particles according to their attributes such as date of birth or

mass of the particle. The hand interaction part implements a direct-manipulation style interface [14] that recognizes sequences of hand gestures made by the user and interprets them according to a state transition diagram, where the user's hand gestures are encoded as events. We have built a small set of gesture commands that are natural, direct and easy to remember.[1] The hand interaction part also implements a hand motion function that allows global interactions by the user. The viewpoint control part implements a nonlinear viewpoint control function that spans large virtual space distance. It gives users the ability to be immersed in the data set and also to have an overview of it. The particle displaying part draws galaxies as resizable points in the space and renders them twice, once for the left eye and once for the right eye. The animation part implements VCR-like control functions which control what direction and how fast the animation goes.

The Cosmic Explorer provides the following capabilities for interaction and navigation.

Slicing the Universe

The user can select an orthogonal slice of variable thickness by making a series of hand gestures. The software will display from the full data set galaxies which are located only in that slice. The slice can be pushed back and forth along the primary axis perpendicular to the slice. Slicing here serves two purposes. First, it eliminates the galaxies outside the slice, thus it can show clearly features in that slice. This is how the observational cosmologists observe and present their data. Secondly, it allows the display of the full-resolution data, which shows more features than present in the down-sampled data set. A user-selected slice is shown in Figure 3, which reveals a filamentary structure that is hardly recognizable in Figure 2.

Multi-scale visualization

If the user finds a region of interest, he can select a sub-volume and put himself in that sub-volume. The sub-volume now becomes the whole virtual space and all the interactions allowed at the top level can be performed here. The galaxies displayed in this sub-volume are taken from the full data set which are stored in files on the hard disk. Following this scheme a user can travel from cluster to cluster at one moment, and the next moment he can travel into a cluster and observe galaxies forming inside a cluster. Figure 4 is a snapshot during the sub-volume selection. It also shows how visual cues (highlighting in this case) can assist the user in selecting features of interest.

The slice and sub-volume specification and movement is done by making a series of gestures and hand movements. For example, selecting a slice with a thickness involves the following steps:

1. making a slicing gesture and turning the hand to indicate one of the three orthogonal slicing planes,
2. moving the hand while maintaining a slicing gesture to specify the location of the slice,
3. switching to another gesture to indicate fixing the slice location and starting to define the thickness,

[1] Whole-hand devices like DataGlove have been studied by Sturman, et al., in [17]. They distinguish between postures and gestures, with the former meaning stationary positions and latter referring to motions. Here we use the word "gesture" to represent both.

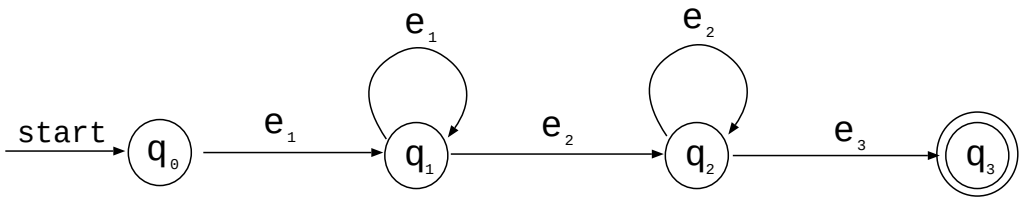

Figure 1: Event-based state transition diagram. There are four states: q_0: before slicing takes place; q_1: first plane being defined; q_2: second plane being defined; q_3: slicing plane has been defined. There are also three events produced by hand gestures: e_1: slicing gesture is made; e_2: thickness gesture is made; e_3: thickness gesture is released.

4. moving the hand while keeping the second gesture to adjust the thickness,

5. changing to a null gesture to indicate the end of the slice definition.

By making a grab gesture, the user can push the slice along the major axis perpendicular to the slicing plane. By this means the user can quickly find interesting features such as filaments in a sheet (see Figure 3).

We use event-based state transition diagrams [8] to interpret hand gesture commands. In an event-based state transition diagram, a node represents a state, an arc represents a transition upon the occurrence of an event. Actions can be associated with either events or states. Hand gestures are encoded as events. The state transition diagram controls the execution of the program. Event buffering is not allowed because of the real-time requirement. There are only limited number of gestures that can be recognized uniquely [17] (about 10 in our case). But by using state transition diagrams, we can handle quite complex operations, because they recognize combinations of hand gestures. The state transition diagram that interprets slicing command is shown in Figure 1. Note that it only takes two gestures to make a slice in this case. It is more direct and economical than other conventional means such as buttons, menus and sliders.

Viewpoint and hand position control

The nonlinear motion control techniques include viewpoint control and hand motion control. These are essentially functions that map the locations in the physical space into ones in the virtual space. They provide a solution to the problem of large scale viewpoint movement (in virtual space) and full-range hand motion control.

The viewpoint control feature allows the user to move into the simulation box and pull out quickly through a nonlinear view transformation function, which is a combination of a linear function inside the data cube and an exponential function outside the cube. If one is interested in a particular cluster of galaxies, one can move close to it and examine it in more detail, because inside the data cube, equal amount of physical displacement of the BOOM translates into equal amount of viewpoint movement in virtual space. If one is interested in getting a global view of the data, one can pull the BOOM away from the data cube. Once the viewpoint gets outside the data cube, a small amount of BOOM displacement translates into a large viewpoint movement. All these can be done by just moving the the BOOM head without making selections for speed control [18] or scale factors [3].

Notice that the user is looking into the BOOM head all the time, and the glove is moving along with the BOOM head. Suppose the user is standing at one corner of the data box and

wants to reach the other end, he cannot do it within a fixed world coordinate system. In some experimental systems, the user must walk to the point where he wants to do an operation, say placing a camera [18]. Our hand position mapping function allows the user to reach any point in the virtual space with a sweep of his hand. So, for example, no matter where the user is standing, he can always select a slice and push it to anywhere. Here, any point in the virtual space is reachable within an arm's length (in physical space). Details of the nonlinear motion control techniques are described in [16].

3 Implementation, Limitations, and Future Work

We have used the Cosmic Explorer to visualize a time-dependent data set generated by a numerical cosmology simulation [5], which simulates galaxy formation in the universe starting from the early age shortly after the Big Bang up until the current time. There are 128 time frames of galaxy data, each frame containing about 17000 galaxies. The galaxies are displayed as resizable points. The color of each galaxy is encoded by the date of its birth. The animation speed can reach 20 frames/second.

The Cosmic Explorer is our experiment of using VR technologies to provide an alternative and possibly better visualization environment for scientists. We have implemented in it several novel techniques including multi-scale visualization techniques and nonlinear motion control techniques. A VR system is much superior to a screen-based tool in many respects. But it also has a higher demand for computing power from the machine driving the whole system. For example, double rendering delays system response time. For a screen-based interactive visualization system, an update rate of 2-3 frames/second would still be tolerable, whereas in a VR system fresh rates below 10 frames/second would be unacceptable to a user.

Another inherent problem with the VR system is the accuracy of positioning. Using a human hand to locate a position in space is less accurate than using a mouse, because a human hand jitters all the time. By the same token, holding a viewpoint still with a human head is not always possible. We have used several techniques, such as casting shadows, using perspective viewing, and highlighting selected regions (see Figure 4), to present visual cues to help the user with 3D positioning.

In the future development of the Cosmic Explorer, we will continue our research on multi-scale visualizations. Some navigational metaphors can be used for multi-scale visualization [11, 18]. 3D widgets [6] should be added to the interface in place of the state transition diagram, which is a fast way to implement interactive user interface but forces the user to remember the sequences of hand gestures. In the process of down-sampling, we can apply some criteria such that the selected samples can still represent the main features of the original data set. We will also add more functions into Cosmic Explorer so that it can visualize a wide range of astrophysical phenomena.

Acknowledgement

We would like to thank Steve Bryson for his generous support and valuable comments during the development of the Cosmic Explorer. In fact, our BOOM and DataGlove library was taken from his virtual windtunnel program [3]. We would also like to thank Michael McNeill for his technical support for the NCSA Virtual Reality Lab. Finally, we want to thank Dr. Jerry Ostriker and Renyue Cen at the Princeton University for providing us with their galaxy formation data set.

This work was partially supported by the NSF under grant number AST9012353 and utilized Silicon Graphics 4D/440 at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign.

References

[1] Brooks, Frederick P. Jr., "Grasping Reality Through Illusion — Interactive Graphics Serving Science," *CHI'88 Conference Proceedings: Human Factors in Computing Systems*, May 15-19, 1988, pp. 1–11.

[2] Bryson, Steve and Creon Levit, "A Virtual Environment for the Exploration of Three Dimensional Steady Flows," In *Stereoscopic Displays and Applications II*, John O. Merritt, Scott S. Fisher, Editors, Proc. SPIE 1457, 1991, pp. 161–168.

[3] Bryson, Steve and Creon Levit, "The Virtual Windtunnel: An Environment for the Exploration of Three-Dimensional Unsteady Flows," *Proceedings of Visualization '91*, October 22-25, 1991, San Diego, California, pp. 17–24.

[4] Carolina, Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, "The CAVE: Audio Visual Experience Automatic Virtual Environment," *Communications of the ACM*, Vol. 35, No. 6, June 1992, pp. 64–72.

[5] Cen, Renyue and Jeremiah P. Ostriker, "CDM Cosmogony with Hydrodynamics and Galaxy Formation: Galaxy Properties at Redshift Zero," Princeton University Observatory, submitted to *The Astrophysical Journal*, December 1992.

[6] Conner, D. Brookshire, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and Andries van Dam, "Three-Dimensional Widgets," *Proceedings of 1992 Symposium on Interactive 3D Graphics*, 23 March – 1 April, 1992, pp. 183–188.

[7] Geller, Margaret J. and John P. Huchra, "Mapping the Universe," *Science*, Vol. 246, November 17, 1989, pp. 897–903.

[8] Jacob, Robert J. K., "Using Formal Specifications in the Design of a Human-Computer Interface," *Communications of the ACM*, Vol. 26, No. 4, April, 1983, pp. 259–264.

[9] Latta, Sara, "Computational Astrophysics Comes of Age," *access* (NCSA High-Performance Computing Magazine), Vol. 6, No. 2, May-June 1992, pp. 6–7.

[10] McDowall, I.E., M. Bolas, S. Pieper, S. S. Fisher, and J. Humphries, "Implementation and Integration of a Counterbalanced CRT-based Stereoscopic Display for Interactive Viewpoint Control in Virtual Environment Applications," In *Stereoscopic Displays and Applications*, John O. Merritt, Scott S. Fisher, Editors, Proc. SPIE 1256, 1990, pp. 136–146.

[11] McKenna, Michael, "Interactive Viewpoint Control and Three-Dimensional Operations," *Proceedings of 1992 Symposium on Interactive 3D Graphics*, 23 March – 1 April, 1992, pp. 53–56.

[12] Neeman, Henry, "Interactive Steering of Adaptive Mesh Refinement for Multiscale Simulations," Ph.D. dissertation in preparation, Dept. of Computer Science, Univ. of Illinois at Urbana-Champaign, 1993.

[13] Rheingold, Howard, *Virtual Reality*, Summit Books, New York, 1991.

[14] Shneiderman, Ben, "Direct Manipulation: A Step Beyong Programming Languages," *IEEE Computer*, Vol. 16, No. 8, August 1983, pp. 57–69.

[15] Smarr, Larry and Charles E. Catlett, "Metacomputing," *Communications of the ACM*, Vol. 35, No. 6, June 1992, pp. 44–52.

[16] Song, Deyang and Michael L. Norman, "Nonlinear Interactive Motion Control Techniques for Virtual Space Navigation", National Center for Supercomputing Applications, Technical Report 003, June, 1992.

[17] Sturman, David J., David Zeltzer, and Steve Pieper, "Hands-on Interaction with Virtual Environments," *Proceedings of the Symposium on User Interface Software and Technology (UIST)*, November 13-15, 1989, pp. 19–24.

[18] Ware, Colin and Steven Osborne, "Exploration and Virtual Camera Control in Virtual Three Dimensional Environments," *Computer Graphics*, Vol. 24, No. 2, March 1990, pp. 175–183.

[19] Zimmerman, Thomas G., Jaron Lanier, Chuck Blanchard, Steve Bryson and Young Harvill, "A Hand Gesture Interface Device," *CHI'87 Conference Proceedings: Human Factors in Computing Systems and Graphics Interface*, 1987, pp. 189–192.

Figure 2: A perspective view of the last time step in the galaxy formation simulation.

Figure 3: A thin slice, made by hand guestures, shows filamentary structures in the last time step.

Figure 4: A sub-volume, made by hand gestures and highlighted against surrounding environment, shows a super-cluster in that region.