Interactive Visualization of Solar Mass Ejection Imager (SMEI) Volumetric Data

Yang Yu*, P. Paul Hick**, and Bernard V. Jackson***
Center for Astrophysics and Space Sciences, University of California, San Diego

ABSTRACT

We present a volume rendering system developed for the real time visualization and manipulation of 3D heliospheric volumetric solar wind density and velocity data obtained from the Solar Mass Ejection Imager (SMEI) and interplanetary scintillation (IPS) velocities over the same time period. Our system exploits the capabilities of the VolumePro 1000 board from TeraRecon, Inc., a low-cost 64-bit PCI board capable of rendering up to a 512-cubed array of volume data in real time at up to 30 frames per second on a standard PC. Many volume-rendering operations have been implemented with this system such as stereo/perspective views, animations of time-sequences, and determination of coronal mass ejection (CME) volumes and masses. In these visualizations we highlight one time period where a halo CMEs was observed by SMEI to engulf Earth on October 29, 2003. We demonstrate how this system is used to measure the distribution of structure and provide 3D mass for individual CME features, including the ejecta associated with the large prominence viewed moving to the south of Earth following the late October CME. Comparisons with the IPS velocity volumetric data give pixel by pixel and total kinetic energies for these events.

Keywords: Volume visualization; volume rendering; heliospheric volumetric data

1. INTRODUCTION

Over the last decade, direct volume rendering has become an invaluable visualization technique for a wide variety of applications. Examples include visualization of 3D sampled medical data (CT, MRI), seismic data from oil and gas exploration, computed finite element models[1]. Mitsubishi Electronic Research Lab, Cambridge, MA developed the cost-effective PCI VP500 board capable of rendering 256×256×256 volumes at 30 frames per second in 1998 and spun off Mitsubishi Real Time Visualization (RTVIZ) Concord MA in 1999 which was sold to TeraRecon San Mateo CA in 2001. TeraRecon developed the 2nd generation VolumePro 1000 PCI board which is capable of rendering of 512×512×512 volumes up to 30 frames per second using full Phong shading and interactive classification.

Several specific applications have been proposed to render complex and large-size scientific data sets. Based on the *Volume Explorer* program written by visualization researchers at the San Diego Supercomputer Center[2], we developed our solar wind rendering system around the Volume-rendering Library Interface (VLI) distributed by Mitsubishi to support the VolumePro 1000 board. The VLI supports basic volume rendering functions and manipulation of rendering parameters, much like OpenGL supports attributed primitive rendering. The main objective of our system is integrating high-end volume rendering into a low-cost PC-based virtual environment. The possibility of directly interacting with volume and geometric objects alike makes for a more intuitive and efficient workflow than conventional interaction modes.

The tomographic techniques used to provide the heliospheric volumetric measurements are not described here, and are found elsewhere[3,4] and references therein. The density analyses are derived from Solar Mass Ejection Imager (SMEI) Thomson-scattering brightness data. The velocity volumes shown are from interplanetary scintillation (IPS) velocity measurements from the Solar Terrestrial Environment Laboratory (STELab), Toyokawa City Japan. In Section 2, following, we describe previous work and the location of an early version of the volume rendering interactive software. Section 3 describes how this system is used to determine mass and energy of specific heliospheric structures. Section 4 describes how our interactive capability has been extended to incorporate multiple users via internet access. We conclude in section 5.

Solar Physics and Space Weather Instrumentation, edited by Proceedings of SPIE Vol. OEI133 (SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.616358

2. PREVIOUS WORK

Our system is based on SDSC's Volume Explorer (*vx*) which has implemented basic interaction with volume data around the TeraRecon VolumePro 1000 board. For those interesting in trying it, *vx* can be downloaded from the Web[2] at: http://dvl.sdsc.edu/vx. The rest of the article demonstrates some preliminary work from two viewpoints: rendering volume data and interacting with volume objects.

3. DETERMINATION OF VOLUME, MASS, AND ENERGY

We currently use the Volume Explorer software application developed at UCSD and the San Diego Supercomputer Center to drive the Volume Pro board. In its current implementation the program provides several 'widgets' as controls for the display parameters. The system permits the display of 4D volume data (3 spatial, plus time) allowing for real-time manipulation of an evolving (time-dependent) scene. This feature is essential for the analysis of solar disturbances propagating from the Sun outward into the heliosphere. The TeraRecon Volume Pro board also enables use of stereographics when supported by the appropriate hardware (*i.e.*, a graphics card supported by a stereo-graphics driver, and stereo goggles with an infrared pickup).

The analysis of heliospheric modeling results of the solar wind requires effective visualization tools specifically designed to handle 3D display of volumetric heliospheric plasma density and velocity phenomena. We have developed an interactive visualization application called the Volume Explorer based on Open GL technology[5] and the TeraRecon Volume Pro 1000 board[6]. The system allows the user to "navigate" through the inner heliospheric space covered by the tomographic reconstruction[4].

The user relies on "widgets" based on the open source GLUI[7] user interface library to navigate through the heliospheric volume data. Through the widgets the user may change the camera settings, toggle on/off geometries such as the sun, the earth and its orbit, show the time of the events as well as calculate the mass and/or energy of solar wind plasma in a specific region in heliospheric space. The rendering engine of the application then interprets the commands sent by the widgets for proper display.

Once the display is available and manipulated to an appropriate viewing location, and highlighted using an appropriate contour interval, a cursor can be manipulated into the highlighted volume and the integration begun by using pixel or combined-pixel summing within the interval. The pixels used in the summing are also highlighted. This technique and the summed analysis is demonstrated in Figure 1 for the October 28, 2003 (Halloween) CME that began outward motion from the Sun at about 10:30 UT. In this way specific volumes can be defined and their extents, 3D masses, and other parameters determined.

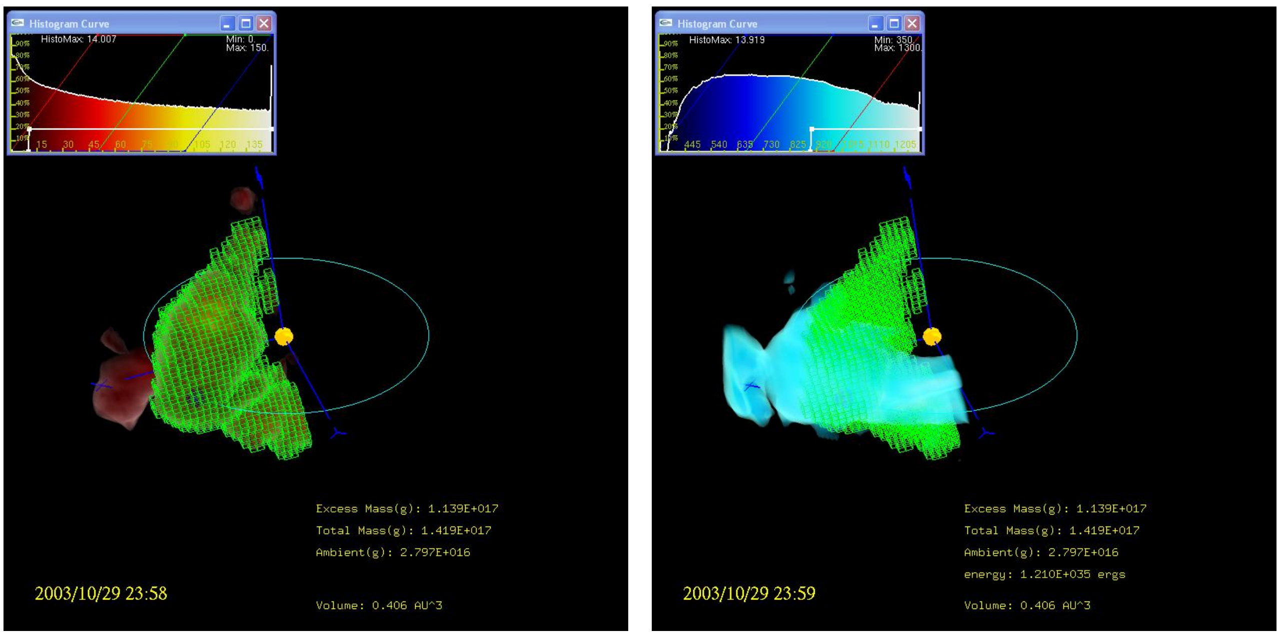
Since the event volume is determined, both the total mass within the volume and an ambient mass defined by the volume can be calculated. The volumetric density display has an r^{-2} scaling applied so that there is no large change in the volume extent as the volume outward motion proceeds. The ambient value at 1 AU is determined in the original volumetric analysis, and it is listed in the volume header. This value is read from the volume header by the volumetric rendering program. Both total and excess mass is scaled by r^{-2} to provide a mass within the defined volume that is much larger per volume element closer to the Sun.

Density and velocity volumes can be simultaneously displayed, and at present the volume defined for density can be translated pixel by pixel to the velocity volume. The result of these two rendered volumes can also be combined and displayed giving, in this instance, the total solar wind outward motion kinetic energy ($1/2 \sum_N m_N v_N^2$) of the material within the defined volumes as shown in Figures 1a and 1b.

In the case of the extensive and fast-moving CME on October 28, 2003 CME, this analysis gives a very large amount of mass and energy for this event. The specifications of this event pertaining to the 3D volume rendering are presented as calculated on the display, and repeated in Table 1. Table 1 gives the volume and other specifications highlighted above the 10 $e^{-}cm^{-3}$ contour level.

Table 1. Specifications of the highlighted volumes of Figure 1

Volume (AU^3)	0.4
Excess mass (g)	1.1×10^{17}
Total mass (g)	1.4×10^{17}
Total energy (ergs)	1.2×10^{35}



(a) **(b)**

Figure 1. Example of volume rendering and volume pixel summing for the October 28, 2003 CME as shown at 0 UT October 30, 2003. **a)** Highlighted summed density. **b)** Velocity above 900 kms^{-1} is highlighted, and this is shown with the transposed associated superposed density volume. The pixel-by-pixel energy sum is shown on the display, and this is repeated in Table 1.

A more complete summary of this October 28, 2003 CME can be found in many different papers including that by *Jackson et al.* [4] in these proceedings. The reconstructed velocity is determined from interplanetary scintillation (IPS) measurements that at this time are very sparsely represented to the south of the Sun. However, for this event the high speed solar wind shows up primarily as a high-speed cap above the dense structure. The shock response for this event takes an abnormally short time[8] of slightly over 20 hours to reach Earth, and thus arrives at Earth on October 29 at about 7 UT. Since the dense material arrives somewhat later and takes longer to arrive (~36 hours, see figure 1), we suppose that the shock stands off in front of the bulk of the CME mass beyond the outer extent of the high velocity, and that this shock response is not measured well as density in the volumetric analysis. We note that we have observed this same type of solar wind response (the high-speed cap of solar wind) for the other events we have studied[3], including the Bastille-Day CME of July 14, 2000.

4. DESIGN AND IMPLEMENTATION OF CLIENT/ SERVER NETWORK COMMUNICATION

We have explored ways to communicate over the internet between two computers. In recent innovations with this system, UCSD has provided a server/client program which enables users at remote locations use these systems for interactively manipulating the other's volume and all of the widgets of the volume-rendering system, using a dedicated

IP address and the internet. Only widget commands are sent from one location to another, and thus volume rendering can proceed between server and client at remote locations at the full resolution of the computer video card, and with almost no delay caused by the internet connection. We can also send and receive volume-state information so volumes at each location are at the same state without interactive manipulation.

To facilitate this we have designed and implemented a socket-based client/server communication system. One of the two machines acts as a server and the other acts as a client. The two communicate through a socket connection. A group of functions are used for the communication. Some are specific to the server, some to the client, and both server and client share a few.

For the server and client to communicate, a connection must first be established. The server prepares the connection with the following steps:

1. Call *socket* to create the communication socket.
2. Call *bind* to establish a specific network port to accept connection requests. Port numbers are 16 bit unsigned integers. The lower numbers are reserved for standard services. For example, the port number for the FTP server is 21. It is important that the port number we use does not conflict with those used by standard services. We use port 4286 in the communication.
3. Listen to client connections by calling *listen*.
4. Accept the connection from the client by calling *accept*. This call blocks communication until the client connects the server.

On the client side, the following steps are taken:

1. Call *socket* to create the communication socket.

2. Connect the socket to the address of the server using the *connect* call. If the server is not available, the client will act as a stand-alone application.

For the communication to function properly, the server must be initiated first, and the client must know the server's IP address in advance.

Once the connection is established, both parties send and receive data packets by calling *send* and *recv*.

Next of concern is what to transmit between the client and the server. We define the current display parameters of the volume explorer as its display state. One option is to transmit the entire current state to the other party each frame. Another approach is to compare the current state to the previous one and only send an update when there is any difference. We choose the latter approach to reduce network traffic. A typical packet has only 50 to 100 bytes and is transmitted through the Internet without noticeable delay.

The current state of the volume explorer display can be also saved to a file. The file when reloaded later recovers the state at the time of file creation.

Current communication is through a public network. A particular port on the computer has to be opened to allow communication. A malicious user could attack the computer through this port or tamper with the data transmitted between two legitimate users. A lesser concern is that the transmitted data are subject to eavesdropping. To address this problem, in the future we intend to base the networking communication on Secure Socket Layer (SSL) protocols. The SSL protocols allow the authentication of both the client and the server using standard key cryptographic techniques. The integrity and privacy of the data transmitted are also protected by the SSL protocols.

It is desirable to allow better visual and audio contact between users of the Volume Explorer using the same computers that analyze the volumetric data. Currently, this is done through NetMeeting from Microsoft. Currently, NetMeeting posses many security concerns[9]. Also, many features of NetMeeting that weakens its securities such as remote desktop sharing are not required for users of the Volume Explorer. In the future we expect to provide better privacy in visual and audio communication using cryptographic techniques between designated IP addresses within our server/client user group.

UCSD has distributed this VP 1000 PC system and software to our colleagues at Sacramento Peak Observatory, Hanscom Field Massachusetts, and the IPS group at STELab, Japan. A similar system is also available at the Space Environment Center (SEC), Boulder, Colorado.

5. CONCLUSION

We have presented some preliminary work of our visualization system which aims to integrate high-end volume rendering with virtual reality interaction into low-cost PC-based virtual environment. Our future interactive software will be distributed with the visualization rendering and interaction tasks that use a client-server rendering approach.

ACKNOWLEDGEMENTS

This work was supported in part by NSF grant ATM-0331513 and MURI subcontracts F49620-01-1-0360 (University of California, Berkeley) and F49620-01-1-0335 (U. Michigan) from the U.S. Air Force. This work would not have been possible without the support of T. Kuchar, D. Mizuno, S. Price and D. Sinclair of the Air Force Research Laboratory, Hanscom AFB, who organized and have distributed the SMEI data to team members, and to M. Kojima and M. Tokumaru and the staff at STELab, Japan who have made IPS data available to UCSD. SMEI was designed and constructed by a team of scientists and engineers from the U.S. Air Force Research Laboratory, the University of California at San Diego, Boston College, Boston University, and the University of Birmingham in the U.K. Financial support was provided by the Air Force, the University of Birmingham, and NASA

REFERENCES

1. A.L. Fuhrmann, B. Ozer, L. Mroz, H. Hauser. VR ^{2} - Interactive Volume Rendering using PC-based Virtual Reality, *IEEE Visualization*, 2002, (submitted).
2. M. Bailey, Visualization Viewpoints: Interacting with Direct Volume Rendering, *IEEE Computer Graphics and Applications*, **21, No. 1**, 10, 2001.
3. B.V. Jackson, and P.P. Hick, “Three-dimensional tomography of interplanetary disturbances”, in: Solar and Space Weather Radiophysics Current Status and Future Developments, D.G. Gary and C.U. Keller, eds., *ASSL* **314**, Kluwer, The Netherlands, 355, 2004.
4. B. V. Jackson, A. Buffington, P. P. Hick, X. Wang, Low Resolution Three Dimensional Reconstruction of CMEs Using Solar Mass Ejection Imager (SMEI) Data, *Proc of the S.P.I.E.*, this issue, 2005, (submitted).
5. A. Kaufman, F. Dachille IX, B. Chen, I. Bitter, K. Kreeger, N. Zhang, and Q. Tang, Real-Time Volume Rendering, *Int. Journal of Imaging Systems and Technology*, **11**,44,2000.
6. Volume Pro 1000 Operation guide http://www.terarecon.com/support/vp1000/vp1000_sup_document.html
7. P. Rademacher, GLUI User Interface Library, http://glui.sourceforge.net
8. E.W. Cliver, and L. Svalgaard, The 1859 solar-terrestrial disturbance and the current limits of extreme space weather activity, *Solar Phys.*, **224**, 407, 2004.
9. NetMeeting security concerns, http://www.sans.org/rr/whitepapers/windows/276.php.

* yyu@cs.ucsd.edu; phone 1 858 534-3358; fax 1 858 534-0177

** pphick@ucsd.edu; phone 1 858 534-8965; fax 1 858 534-0177

*** bvjackson@ucsd.edu; phone 1 858 534-3358; fax 1 858 534-0177; http://casswww.ucsd.edu/solar/smei/index.html; mail code 0424, UCSD, 9500 Gilman Drive, La Jolla, CA 92093-0424