

INTERACTIVITY IS THE KEY

William Hibbard and David Santek

Space Science and Engineering Center
University of Wisconsin
1225 West Dayton Street
Madison, WI 53706

ABSTRACT

The interactivity provided by rendering at the animation rate is the key to visualizing multivariate time varying volumetric data. We are developing a system for visualizing earth science data using a graphics supercomputer. We propose simple algorithms for real-time texture mapping and volume imaging using a graphics supercomputer.

KEYWORDS: Interactive, texture mapping, volume image, earth science.

INTRODUCTION

At the Space Science and Engineering Center we are concerned with the problem of helping earth scientists to visualize their huge data sets. A large weather model output data set may contain one billion points in a five-dimensional array, composed of a 100 by 100 horizontal grid, by 30 vertical levels, by 100 time steps, by 30 different physical variables. Remote sensing instruments such as satellites, radars and lidars produce similarly large data sets. During the last six years we have been developing software tools for managing and visualizing such data sets, as part of the Space Science and Engineering Center's Man-computer Interactive Data Access System (McIDAS). These tools run on an IBM 4381 and produce animation sequences of multivariate three-dimensional images which are viewed on our large multiframe workstations. However, each image takes 10 to 30 seconds of CPU time to analyze and render, and another 30 seconds to

load into the workstation frame store, so turn-around time for producing an animation sequence can be several hours. Figure 1 is a typical image produced by this system. We have applied this system to generate animations from at least twenty different model simulation and remote sensed data sources. Although our earth scientist collaborators are usually pleased with the results, they all express a desire to change the animations with quicker response. This is by far their (and our) primary request. Therefore we have begun developing a highly interactive workstation based on the Stellar GS-1000 graphics supercomputer. This system can produce three-dimensional images from model output data sets in real-time, giving the scientist control over the image generation with immediate feedback.

This paper describes earth science data sets, the work we have done so far with the Stellar GS-1000, and some thoughts on further development.

LARGE DATA SETS

Earth scientists are concerned with the three-dimensional domains of atmosphere, ocean, and earth. They are also concerned with the two-dimensional domains which define the boundaries between atmosphere, ocean and earth, particularly with the new drive to understand our planet as a single coupled system. Data sets over these domains are generally sampled at regular grids of points in some two or three-dimensional coordinate system. The vertical dimension is often sampled at higher resolution, and visualized on an expanded scale, because of the extreme thinness of the atmosphere, oceans, and crust when viewed on a global scale. Earth systems are dynamic, so the data sets include a time dimension, usually sampled at regular intervals. The time intervals depend on the nature of the earth system being studied, and are matched to the spatial resolution to give reasonable motions. The data sets are usually multivariate reflecting the interacting physical quantities being studied. At the extreme, atmospheric chemistry data sets may include



densities for hundreds of different chemical constituents.

Much remote sensed data takes the form of images. These may lie on some two-dimensional surface through three-dimensional space, or each pixel may integrate along a ray through space. Each pixel may have multiple components, for different spectra, polarizations or other physical quantities. Large satellite images are up to 16,000 by 16,000 pixels. Planned instruments will return hundreds of spectral components at each pixel.

INTERACTIVE 4-D WORKSTATION

We are actively developing software to apply the graphics power of the Stellar GS-1000 to visualize large earth science data sets. Our approach is to store a data set and derived data structures (such as polygon and vector lists) in main memory and to produce real-time animations of multivariate three-dimensional images from these data, giving the user as much interactive control over the image generation process as possible. For a five-dimensional model output data set, main memory contains:

- ** a five-dimensional (three space, time and physical variable) array containing the model output data set. For a very large data set, the memory resident array has reduced time and space resolution and coverage, and a subset of the physical variables.
- ** polygon lists representing contour surfaces through three-dimensional scalar fields.
- ** vector lists with time coordinates representing particle trajectories generated from four-dimensional vector fields.
- ** polygon lists representing a topographical map, along with vector lists for boundary lines.
- ** vector lists representing contour lines of two or three-dimensional scalar fields on a two-dimensional surface.
- ** pixel maps as the destination of rendering operations.

We give the user interactive control over:

- ** time animation, which may be enabled or disabled, single stepped forward or backward, and the animation rate changed.
- ** the perspective mapping for rotation and zoom.
- ** the combination of physical variables to be rendered, including contour surfaces and lines, trajectories and topographical map.

- ** the selection of new contour levels for contour surfaces of scalar variables.

The user gets immediate (a tenth to a fifth of a second) visual feedback to all of these controls. Although the system cannot calculate the polygons for a new contour level at the animation rate, the new surfaces replace the old ones in the animation as they are calculated, so the user immediately begins to see the effect of this control. For three-dimensional grids containing 20,000 points, contour surfaces are calculated at a rate of about two per second, concurrent with real-time animation rendering. It is important that a scientific visualization workstation have the floating point performance to support this type of interaction with the analyses underlying the display.

The reaction of scientists to this interactive visualization of their data sets has been very enthusiastic. A typical pattern of use is to look at the most important variables at various contour levels, along with trajectories, noting all the expected phenomena, until something unexpected is seen. Then a cause is sought, by viewing the unexpected phenomenon in combination with other variables which offer candidate explanations. Often adjustments to contour levels are necessary to see the clearest correlation between variables. When a cause is found, it may also need an explanation, and a backward chain of cause and effect links is followed. The key element of this system is that it reacts to the scientists' controls immediately, so that their thoughts are not interrupted by waiting for the system.

We have sometimes heard that vector line segments and polygonal surfaces are not appropriate techniques for volume visualization. Our experience shows that interactivity is the most important tool for attacking multivariate, dynamic, volume visualization problems. The appropriate rendering techniques for such problems are those which can be performed in real-time, giving the user interactive control with immediate response. Thus, we have been concentrating on the vector line segments and polygonal surfaces which can be rendered at real-time animation rates. In the next two sections we will discuss the prospects for real-time animation with other rendering techniques.

VOLUME IMAGING

As part of our software development on the IBM mainframe we have experimented with fast approximate algorithms for producing volume

images, in which a three-dimensional scalar variable is depicted by a transparent fog, with the opacity at every point proportional to the value of the scalar. We have applied these algorithms to cloud water density in a three-dimensional array of about 40 by 40 by 20, as shown in figure 2. The simplest algorithms offer the hope of real-time rendering. The images they produce do have some artifacts, but they are certainly adequate for interactive visualization.

For our approximate algorithm we use the view direction to the center of the image, and choose the principal axis of the scalar grid most nearly parallel to this view direction. The grid is then treated as a series of layers perpendicular to the chosen axis, which are rendered as transparent flat surfaces, interpolating transparency over each little grid rectangle. An alpha value, which is 0.0 for totally transparent and 1.0 for opaque, is calculated at each grid point as proportional to (or some other function of) the gridded scalar value. The rectangular polygons are then rendered with variable alpha, the layers rendered from back to front. In our implementation we use an alpha buffer containing $\log_{10}(1.0 - \text{opacity})$, which composites by addition rather than multiplication. Table look-ups are used to take logs and antilogs. We assume that color is constant throughout the grid, so that it is not necessary to maintain color buffers. Red, green and blue are simply proportional to the final opacity values. The calculation of opacity from grid point value includes a correction for the path length of the view ray between two layers of the grid. This correction is implemented by applying a table look-up to the opacity which raises it to the power, the secant of the angle between the view direction and the chosen principle axis. This look-up table is merged with the logarithm look-up.

This algorithm generates three noticeable artifacts. First, thin high density layers running obliquely through the grid may be rendered with oscillating bands of intensity, a form of aliasing. Second, the individual layers can be seen at the edge of the array domain. Third, parts of the image change discontinuously when the chosen principal axis changes during rotation. There is a simple fix for the third problem, which helps with the other two. Render three versions of the image, one for each principal axis, and take an average of these weighted by the cosines of the view direction with the principal axes.

The one axis method requires rendering a number of rectangles roughly equal to the number of grid points of the scalar variable, while the weighted average of axes method renders three times that number of rectangles. And, of course, each pixel must be accessed a number of times roughly equal to the number of grid layers rendered. For a 40 by 40 by 20 grid, this is 32,000 or 96,000 rectangles, and accessing each pixel 20, 40 or 100 times. These numbers are a bit high for current graphics engines to do in real-time, but they can be reduced significantly by culling out all rectangles with all four corners below some opacity threshold.

We would certainly like to see support for such rendering algorithms on commercial graphics engines.

IMAGE MAPPING

Much remote sensed environmental data is generated in the form of images on two-dimensional surfaces embedded in three dimensions. For example, a radar scan at a positive elevation sweeps out an image on a cone. Figure 3 shows two perpendicular vertical lidar scans. It is desirable to visualize such data in their natural geometry, possibly combined with other data, and rotating under user control. We believe that current graphics engines can support real-time texture mapping for such images, using a simple algorithm.

The source image to be mapped is stored in memory in a simple two-dimensional array, with X and Y pixel addresses. The surface is divided into polygons. At each vertex of the polygon mesh record the X and Y addresses of the corresponding pixel from the source image. Then render the polygon mesh in three-dimensions with a Z-buffer algorithm. However, where Gouraud shading would interpolate color intensities, interpolate instead the X and Y addresses of pixels. The rendering engine should also combine the X and Y address into a single array address, requiring a multiply. Where this is difficult, it may be possible to use shift and add, and assume that the image is stored in an array with one dimension equal to an even power of two. The result will be an image filled with addresses from the source image. These addresses would then be replaced by the values from the source image, probably using a fast scalar processor or a special vector instruction.

This algorithm requires a flexible system. The rendering processor must be able to combine pixel addresses, and the rendering output should be available to some processing element which can interpret its pixels as addresses and replace them with the addressed values. We would like to see support for this type of texture mapping algorithm on commercial systems.

CONCLUSION

The interactivity implied by real-time animation is the key to giving scientists visual access to their large, multivariate, time dynamic, volume data sets. At SSEC we have begun developing an interactive system for managing, analyzing and visualizing earth science data using the Stellar GS-1000. Our initial work with the GS-1000 demonstrates a breakthrough in providing earth scientists with access to their large data sets.

Because interactivity is crucial, the appropriate rendering techniques for volume visualization are those which can be rendered in real-time. Currently, this means rendering using vector line segments and polygonal surfaces.

However, it should be possible to adapt current graphics engines to real-time rendering of volume images, at least by approximate algorithms, and to texture mapping. The types of simple algorithms presented here, and more complex variations, argue for flexibility in commercial rendering systems. At SSEC we selected the Stellar GS-1000 because its rendering processor is micro-coded and because its pixel maps are rendered into main memory, where they are available as input for multi-stage rendering algorithms. It is even possible in such an architecture that the power of the rendering processor could be applied to data analysis. Environmental data are generated in a great variety of forms, with unpredictable processing requirements; this argues for a flexible and programmable system. Besides flexibility in the rendering processor, we also appreciate systems with powerful scalar and vector processing, and without communications bottlenecks between partitioned memories or processors.

Many meteorologists still draw contour analysis from weather observations by hand. In fact, pads of printed maps showing weather observing station locations are manufactured for this purpose. These pads give the meteorologist coordinated hand-eye interaction with their data sets, which they miss with computer-generated graphics. The

graphics supercomputer workstation gives this hand-eye feel back to meteorologists, applied to data sets containing millions rather than hundreds of points.

ACKNOWLEDGMENTS

We would like to thank P. Pauley, G. Tripoli, R. Schlesinger, L. Uccellini, E. Eloranta and all of our other earth science collaborators. This work was supported by NASA (NAS8-33799 and NAS8-36292).

REFERENCES

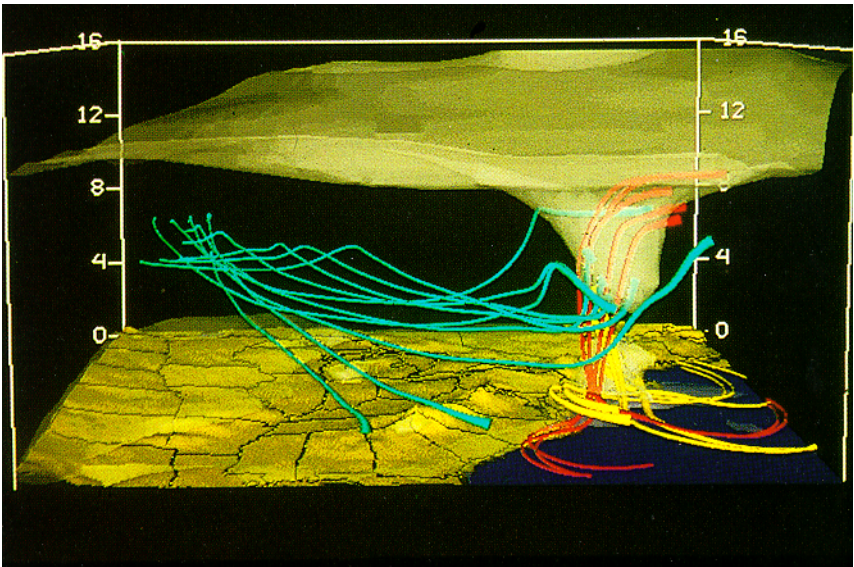
- Dreben, R., Carpenter, L., and Hanrahan, P. [1988] Volume Rendering. *Computer Graphics* 22, 4, SIGGRAPH, pp. 65-74.
- Foley, J. and Van Dam, A. [1982] *Fundamentals of Interactive Computer Graphics*. Addison-Wesley Publishing Company, 664 pp.
- Hibbard, W., Krauss, R.J. and Young, J.T. [1985] 3-D weather displays using McIDAS. Preprints, *Conf. Interactive Information and Processing Systems for Meteor., Ocean., and Hydro.* (Los Angeles, AMS, 7-11 Jan. 1985), pp. 153-156.
- Hibbard, W. [1986a] 4-D display of meteorological data. In *Proceedings of 1986 Workshop on Interactive 3D Graphics*. (Chapel Hill, SIGGRAPH 23-24 Oct. 1986) pp. 23-36.
- Hibbard, W. [1988] A next generation McIDAS workstation. Preprints, *Conf. Interactive Information and Processing Systems for Met, Ocean, and Hydro.* (Anaheim, American Meteorology Society, 1 Jan. - 5 Feb. 1988) pp. 57-61.
- Hibbard, W. and Santek, D. [1988] Presidents' Day Storm. Visualization/ State of the Art: Update. SIGGRAPH Video Review number 35. ACM SIGGRAPH.
- McCormick, B.H., DeFanti, T.A., and Brown, M.D., Editors, [1987] *Visualization in Scientific Visualization. Report to the NSF by the Panel on Graphics, Image Processing and Workstations*. ACM SIGGRAPH, 88 pp.
- Pauley, P. M., Hibbard, W., and Santek, D. [1988] The use of 4-D graphics in teaching synoptic meteorology. Preprints, *Conf. Interactive Information and Processing Systems for Meteor., Ocean., and Hydro.* (Anaheim, AMS, 1 Jan. - 5 Feb. 1988) pp. 33-36.

Rogers, D. [1985] *Procedural Elements for Computer Graphics*. McGraw-Hill Book Company, 433 pp.

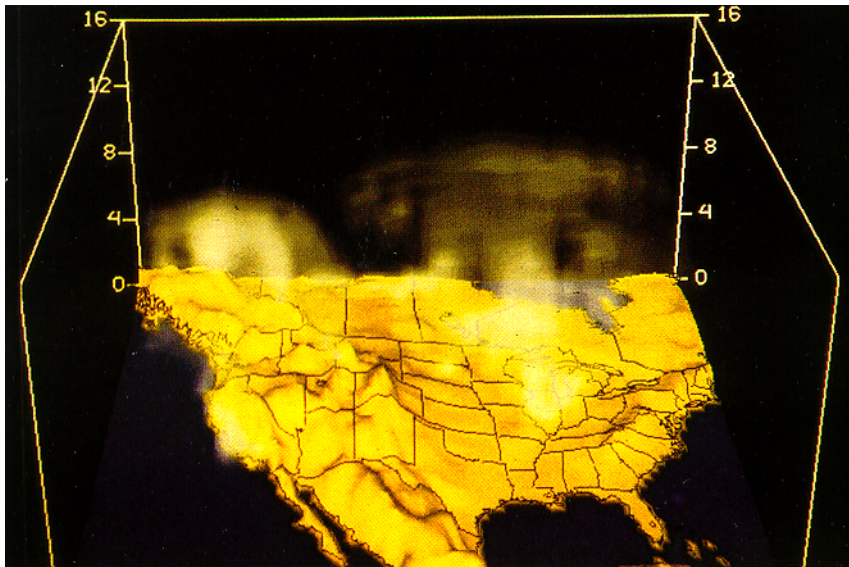
Santek, D., Leslie, L., Goodman, B., Diak, G., and Callan, G. [1987] 4-D techniques for evaluation of atmospheric model forecasts. In *Proceedings*,

Digital Image Processing and Visual Communications Technologies in Meteorology (Cambridge, SPIE, 26-29 Oct. 1987] pp. 75-77.

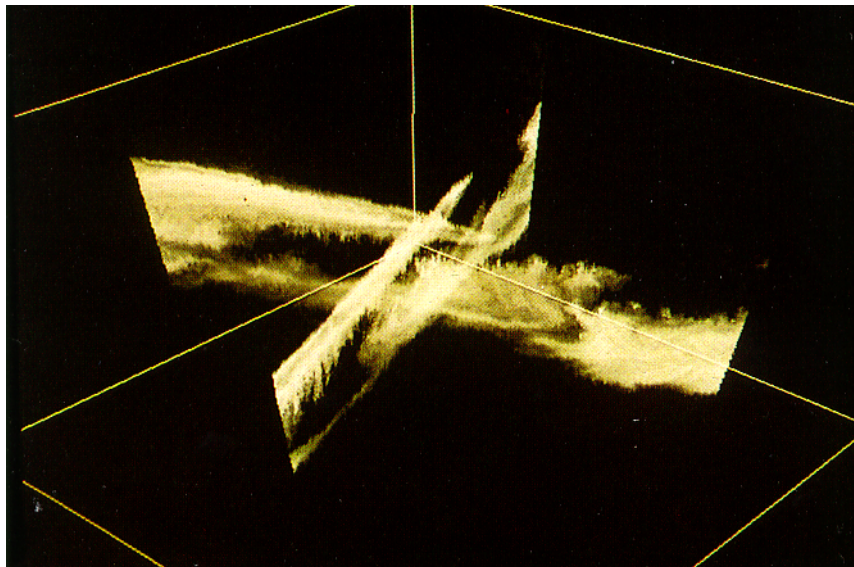
Upton, C. and Keeler, M. [1988] V-BUFFER: Visible Volume Rendering. *Computer Graphics* 22, 4, SIGGRAPH, 59-64.



Hibbard and Santek. Figure 1.



Hibbard and Santek. Figure 2.



Hibbard and Santek. Figure 3.