

Volumetric Medical Imaging Environment

Bobi Gilburd¹, Michal Holtzman-Gazit², Alon Spira¹, Dorit Goldsher³, and
Ron Kimmel¹

¹ Computer Science Department
{bobi,salon,ron}@cs.technion.ac.il

² Electrical Engineering Department
miki@cs.technion.ac.il

³ Faculty of Medicine
Technion - Israel Institute of Technology,
Technion City, Haifa 32000, Israel

Abstract. We describe a graphical tool for analysis and visualization of three dimensional medical data. The application is intended to help physicians interpret the geometric structures of volumetric medical data and navigate through it easily. It combines novel approaches for segmenting the data and advanced techniques for visualizing the models which allow fast processing on a standard personal computer. The software's size and its minimal system demands make it suitable for remote operation. We describe the various components of the application, and provide sample results.

1 Introduction

As volumetric medical data analysis algorithms improve, there is a growing interest of physicians for cheap and simple systems implementing them. Most of today's commercial systems ([30, 29] for example) require powerful dedicated workstations for visualizing the models, thus increasing the overall cost of the system. Other systems, such as [5], require minimal hardware yet provide only part of the features of our system.

In this paper we describe a *Volumetric Medical Image Manipulation* environment - VMIM, a graphical tool for analysis and visualization of models created from slices of volumetric data. It achieves high graphical performance on a standard personal computer while allowing easy interactive browsing through anatomical models. It integrates all aspects required for analyzing volumetric data: segmentation, visualization, and navigation through the models. In addition, the size of the software and its minimal system demands make it suitable for remote operation.

The input to our system is the volumetric data. It is given typically as slices produced by a CT or MR scan. The segmentation algorithm creates an implicit model from the volumetric data, in which the object boundaries are embedded as the zero set of a three dimensional function. This algorithm appears in Section 2 and its initialization is described in Section 4. The visualization stage consists

of transforming the implicit object representation into a triangulated manifold and rendering it to the screen, see Section 3. Navigating through the object is described in Section 5.

2 Segmentation

The aim of the segmentation process is to extract a three dimensional implicit surface for each anatomic objects, e.g. vessels and bones. The algorithm we use, fully described in [16], is based on geometric active surfaces. These are surfaces that evolve according to geometric partial differential equations until they stop at the boundaries of the objects. More information on active contours and surfaces can be found in [19, 8, 26, 25, 10, 9, 17, 18, 24, 15, 27, 12, 11].

The algorithm receives the slices produced by a CT or MR scan as an input, and pre-processes them to enhance their contrast. Then, we use active surfaces which minimize the functional

$$E_{Total} = E_C + \beta E_{MV} + \gamma E_{GAC}, \quad (1)$$

where β, γ are constants and the other measures are described below.

$$E_C(S) = \iint_S \langle \nabla I, \mathbf{n} \rangle da - \iiint_{\Omega_S} H_I |\nabla I| dx dy dz, \quad (2)$$

is the alignment term, where $S(u, v)$ is the evolving surface, $I(x, y, z)$ is the volumetric data, \mathbf{n} is the unit normal to the surface, da is a surface area element, Ω_S is the volume inside the surface and H_I is the mean curvature of the level sets of the volumetric data. This functional was presented in [21] and minimizing it yields the Euler-Lagrange (EL) equation $I_{\xi\xi} \mathbf{n} = 0$, where $\boldsymbol{\xi}(x, y, z) = \frac{\nabla I}{|\nabla I|}$ is the gradient direction vector field, and $I_{\xi\xi}$ is the second order derivative along the gradient direction.

$$E_{MV}(S, c_1, c_2) = \iint_{\Omega_S} (I - c_1)^2 dx dy dz + \iint_{\Omega \setminus \Omega_S} (I - c_2)^2 dx dy dz, \quad (3)$$

is the minimal variation term, where c_1, c_2 are the averages of I in the interior and exterior of S respectively. This criterion was presented in [11].

$$E_{GAC}(S) = \iint g(S) da, \quad (4)$$

where $g(x, y, z)$ is an inverse edge indicator function given, for example, by $g(x, y, z) = \frac{1}{1 + |\frac{\nabla I}{\alpha}|^2}$, with α constant. This is the geodesic active surface term [9, 10]. Minimizing this functional results in a surface along which g obtains the smallest possible values. The EL equation of this functional is $(gH - \langle \nabla g, \mathbf{n} \rangle) \mathbf{n} = 0$, where H is the mean curvature of the surface S .

The surface evolution towards an extremum derived from (1) is given by:

$$S_t = (-I_{\xi\xi} - \beta((I - c_1)^2 - (I - c_2)^2) + \gamma(gH - \langle \nabla g, \mathbf{n} \rangle)) \mathbf{n}. \quad (5)$$

Using the Osher-Sethian level set formulation [28] to represent the surface as the zero set of a three dimensional function $S = \{x, y, z \mid \Phi(x, y, z) = 0\}$ yields the equation

$$\Phi_t = \left(-I_{\xi\xi} - \beta((I - c_1)^2 - (I - c_2)^2) + \gamma \left(\operatorname{div} \left(g \frac{\nabla \Phi}{|\nabla \Phi|} \right) \right) \right) |\nabla \Phi|. \quad (6)$$

In order to extract more than one object from the volumetric data, we use a hierarchical method. If there is a need for further segmentation, after the first use of the algorithm, we apply it again to one of the regions generated from the previous step. This way, we segment only the interesting parts of the volume.

In order to implement the PDE on a computer we need a suitable numerical scheme. $\Phi(x, y, z)$ is restricted to be a distance map, computed by the fast marching algorithm [31]. We use the narrow band approach [4, 13] and the implicit locally one dimensional (LOD) scheme suggested in [20]:

$$\Phi^{n+1} = \prod_{i=1}^3 (\mathcal{I} - \tau \gamma A_i)^{-1} (\Phi^n + \tau f), \quad f = -(\beta((I - c_1)^2 - (I - c_2)^2) + I_{\xi\xi}),$$

where \mathcal{I} is the identity matrix, τ is the time step, and A_i are tri-diagonal matrix operators given by $A_i = \frac{\partial}{\partial i} g \frac{\partial}{\partial i}$, where $i = x, y, z$. This efficient numerical scheme enables a relatively fast segmentation process. For a volumetric data of size 100^3 voxels the program runs for a few minutes on a Pentium III PC for complicated segmentation tasks. See Figure 1 for a demonstration of the algorithm on a three dimensional MRI of the brain.

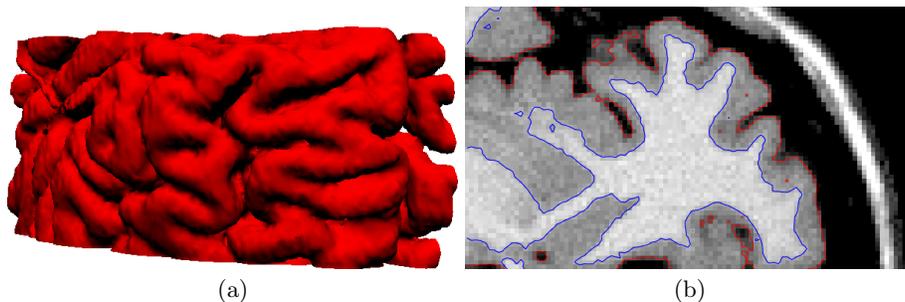


Fig. 1. The result of our segmentation algorithm on a 3D MRI of the brain (a) as a three dimensional model (b) as a two dimensional slice.

3 Visualization

The segmentation process described above results in a surface implicitly defined in 3D for each of the segmented objects. Our goal is to construct a medical

analysis tool that would enable a medical expert to easily and efficiently examine the objects. A common approach to this task is to represent the surface as a collection of voxels, and render them to the display by using various ray-tracing techniques; see [6, 7] for examples and overview. In our test cases, a 100^3 volume of voxels from a CT scan of the brain resulted in a surface represented by 150,000-300,000 voxels. Rendering that many voxels requires high processing power, which would not allow smooth rendering of the the volume when rotating it or zooming into it, while using a standard processing-power machine. To avoid these performance problems, several heuristic techniques were implemented, such as rendering optimization in the case of tubular shapes [6]. Here we introduce and explore a more general solution.

Our approach is to represent the surfaces as triangulated manifolds - a collection of triangulated “faces”, as seen in Figure 2. This representation is the natural choice, considering that this is the native language for most of today’s state-of-the-art video-processing hardware and software. In our test cases, the same 100^3 volume resulted in only 30,000 triangles describing the vessels’ manifolds and 15,000 triangles describing the bones’ manifolds. This is an order of magnitude less objects to render to the screen and keep in memory, compared with the voxels approach.

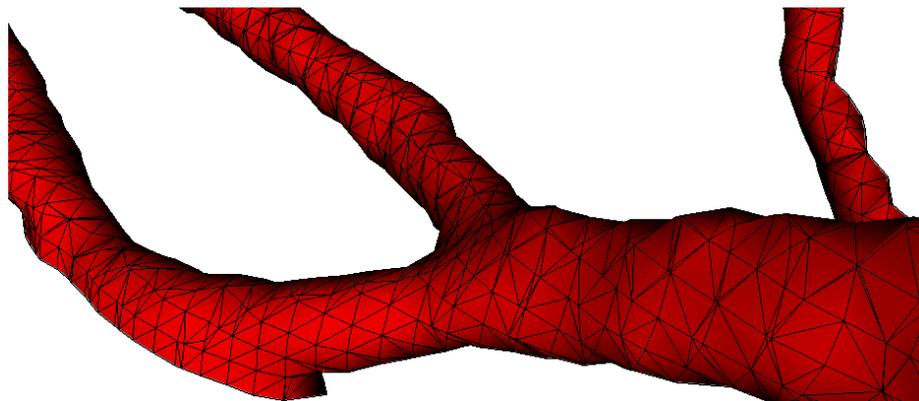


Fig. 2. The triangulated manifold representing a blood vessel.

In addition to the faster rendering and the lesser memory requirements, representing the volume as a manifold has two main advantages. The first is that it allows a faster calculation of light shadows for the objects, which provide a more realistic three dimensional visualization. This is due to the simpler calculation of reflection from a flat triangle as compared to shadowing a discrete set of voxels. The same goes for fast calculation of transparency and blending for the objects. Transparency defines the way in which an object hides other objects, while blending defines how a texture is combined with the colors of the surface

to which the texture is applied. See [22] for a good introduction regarding these terms. The second advantage is that the triangles approach results in a smooth view of the volume for any zoom level of the object. This is in contrast to the discrete voxels approach, in which the voxels are visible at close range.

In order to convert the implicit surface into a triangulated manifold, we use the “pyramids extrapolation” technique from [18]. Using this technique, which is motivated by the “Marching Cubes” algorithm [23, 14], we achieve sub-voxel accuracy for the manifold by splitting each grid cube into five pyramids and within each pyramid approximating the distance by a four dimensional hyperplane.

Our experimental system is based on Microsoft Windows’ DirectX graphics library [1] which implements today’s state-of-the-art rendering technologies, see [32]. Of course, other environments can be used to achieve similar results, e.g. OpenGL [2] and VTK [3]. For creating the shadow we use a model of directional light, supported by the package. In most cases more than one anatomic object exists in the volume and each of them can be shown together with its own color and transparency level. This way, one can easily visualize the structure of vessels and bones, as can be seen in Figure 3. To achieve fast rendering we use a back buffer: a nonvisible surface to which the models’ components are rendered while the front buffer displays the currently visible image. Only when the back buffer is ready we replace it with the front buffer.

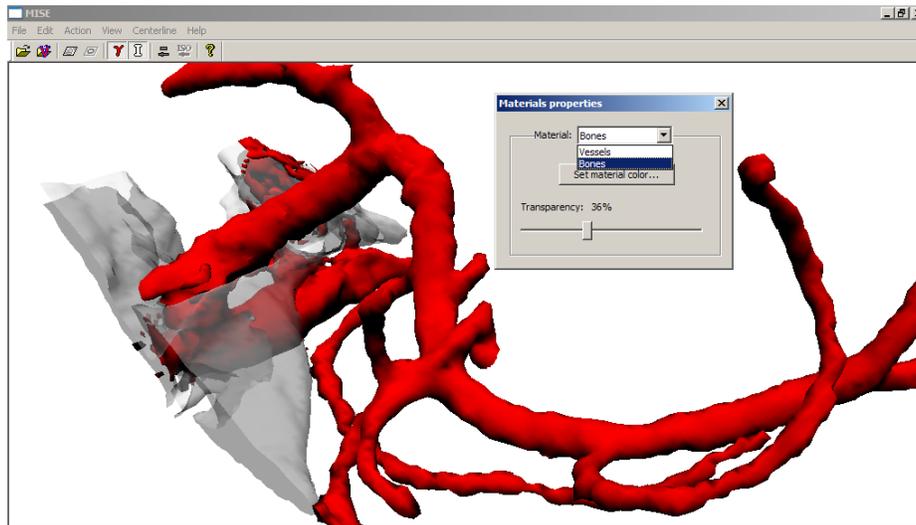


Fig. 3. Blood vessels inside a half-transparent bone in the brain.

The medical expert, who is used to working with the raw volumetric data which is usually represented as a set of MR or CT slices, might find it useful to see the slices together with the three dimensional model. The visualization

system allows the user to add a flat plane, perpendicular to the length-axis of the model, on which the original slices are drawn. This plane can be easily moved up and down, while updating it with the image of the appropriate slice, see Figure 4. This plane requires only the drawing of a single slice for each rendering of the model, so it does not affect the overall rendering speed. It is possible to rotate the model smoothly while updating the plane as well.

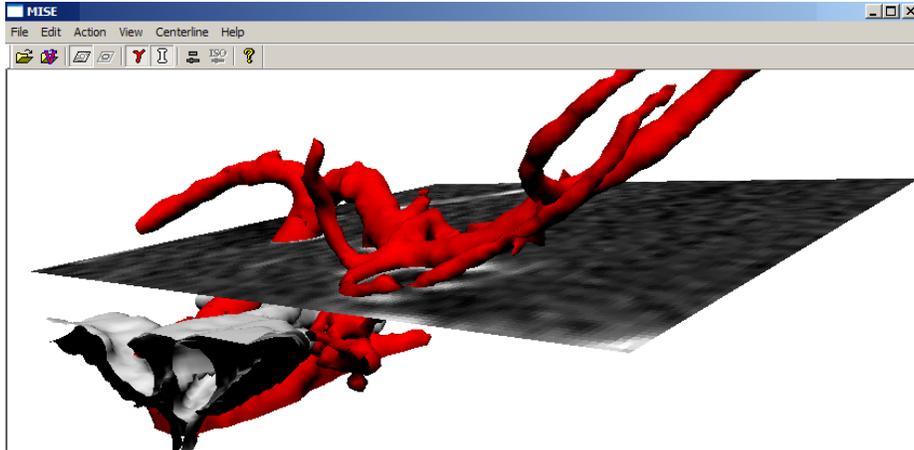


Fig. 4. Viewing the original slices together with the model.

4 Initializing the segmentation

The aim of the segmentation process described in Section 2 is to extract the model from the original volumetric data. The segmentation algorithm works by evolving an initial manifold towards the objects' boundaries. Consequently, before starting the segmentation, one should define the initial manifold, which should fulfill the following requirements: it must be closed and oriented so that we can evolve it outward toward the boundaries. In addition, it should be inside the boundaries of the objects we want to segment, as close as possible to them, so that the segmentation process will be as fast as possible.

The manifold is usually initialized from a single point inside the boundaries by creating a small balloon shaped surface around it. This step requires manual intervention: the physician should select this point. Note that the initialization is a critical stage in the segmentation; choosing an initial point outside the boundaries will result in meaningless results. Our system supports this initialization, but includes also a semi-automatic one, in which the physician controls the creation of an initial manifold, based on the grey levels of the volumetric data.

For the manual option, our system allows the physician to select the points from the slices' images, while viewing them together with an estimated three dimensional model. The estimated model is created from the volumetric data by extracting the level set that correspond to a selected threshold. The physician selects the initial points by clicking on the slice. The initial point is located at the intersection of the slice with a directed ray from the user's view point. This selection process is demonstrated in Figure 5(a).

In the semi-automatic mode, the physician defines a manifold by selecting thresholds of the scan's grey-levels. We have experimented with two techniques for this selection, both are simple and require minimal interaction from the physician. In the first method, the physician specifies the threshold as a single grey level value. The manifold matching the selected grey level value is extracted, using the "pyramids extrapolation" technique from [18]. This is the initial manifold for the segmentation. The problem with this initialization is that selecting a single grey level value is not enough in case there is more than one object in the volume, e.g. vessels and bone. In our tests we found that $V_{low} < B_{low} < V_{high} < B_{high}$, where (V_{low}, V_{high}) and (B_{low}, B_{high}) define the optimal grey level ranges for vessels and bones respectively. Therefore, there is no single grey level threshold that results in a model that includes only vessels or only bones. To avoid this problem, the user specifies two thresholds. Then a map is created by setting a value of 1 for the cells associated with voxels whose grey level is below the lower or above the higher thresholds and -1 for the rest of the cells. The initial manifold is the zero set of the volume. This results in a manifold that represents all the voxels in the selected range. Note that this leads to an "edgy" surface (see Figure 5(b)), because we have discarded the sub-voxel information. This is not a problem, because it is only the initial manifold.

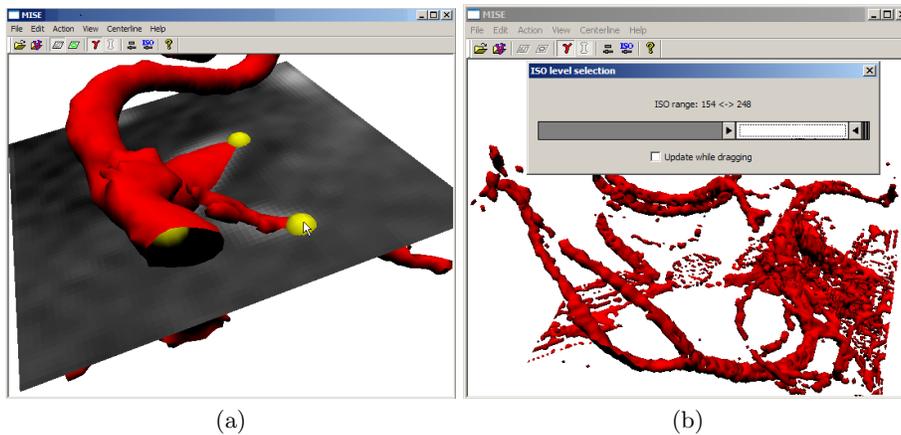


Fig. 5. Initializing the segmentation (a) from initial points by viewing the volumetric data slices together with an estimated model (b) by selecting a range of grey-levels.

5 User interface

One of the important advantages of our visualization tool is the ability to easily navigate through the three dimensional model using a standard mouse. To achieve this task we use the three mouse buttons in the following way: one button is used to move the object in the X-Y plane. The second button is used to move the object in the Z direction. The third allows the rotation of the object in the following way: clicking on the model, the current position is set as the rotation axis; then, dragging the mouse rotates the model around that axis. This gives a natural feeling of actually rotating the object by hand.

Other supported interactions by the user include moving the slice image up and down, setting the position of the light source, by simply moving it with the mouse and setting colors and transparencies for the various model objects.

The size of the software and its minimal system demands make it suitable for remote operation. We are developing an online version of our tool.

6 Summary

We described our volumetric medical imaging environment, a graphical tool for analysis and visualization of 3D medical data. By combining advanced approaches for segmentation and visualization, it requires no special hardware and can run on a standard personal computer, or even on a remote machine. Having passed the experimental stage, the system is now being prepared to be tested at the radiology department of a major hospital. Currently, we are developing a set of tools that will allow the system to extract measurements of the model, such as the diameter of blood vessels which would enable automatic detection of anomalies such as aneurysm and stenosis.

References

1. Microsoft DirectX. <http://www.microsoft.com/directx>.
2. OpenGL. <http://www.opengl.org>.
3. VTK. <http://www.vtk.org>.
4. D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. of Comp. Phys.*, 118:269–277, 1995.
5. E. Bullitt, A. Aylward, E. Bernard, and G. Gerig. Computer-assisted visualization of arteriovenous malformations on the home PC. *Neurosurgery* 48, page 576, 2001.
6. E. Bullitt and S. R. Aylward. Volume rendering of segmented image objects. *IEEE Trans. on Medical Imaging*, 21(8):998–1002, 2002.
7. E. Bullitt and S. R. Aylward. Volume rendering of segmented tubular objects. *Lecture Notes On Computer Science*, 22(8), 2002.
8. V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematic*, 66:1–31, 1993.
9. V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces based object segmentation. *IEEE Trans. on Pattern Analysis Machine Intel.*, 19(4):394, 1997.

10. V. Casseles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
11. T. Chan and L. Vese. Active contour and segmentation models using geometric PDE's for medical imaging. *UCLA comp. and applied math. rep.*, December 2000.
12. T. Chan and L. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, 2001.
13. D. L. Chopp. Computing minimal surfaces via level set curvature flow. *J. of Computational Physics*, 106(1):7791, May 1993.
14. H. E. Cline, W. E. Lorensen, R. Kikinis, and F. Jolesz. Three-dimensional segmentation of MR images of the head using probability and connectivity. *Journal of Computer Assisted Tomography*, 14(6), 1990.
15. T. Deschamps and L. D. Cohen. Fast extraction of tubular and tree 3D surfaces with front propagation methods. *ICPR'02, Canada*, 1:731–734, August 2002.
16. M. Holtzman Gazit, R. Kimmel, and D. Goldsher. Hierarchical segmentation of volumetric medical images. In *Proceedings of MICCAI 2003, Montreal, Canada*, November 2003.
17. R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudsky. Fast geodesic active contours. *IEEE Trans. on Image Processing*, 10(10):1467–1475, 2001.
18. R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudsky. Cortex segmentation: A fast variational geometric approach. *IEEE Tran. Medical Imaging*, 21(2), Dec. 2002.
19. M. Kass, A. Witkin, and D. Terzopolous. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
20. R. Kimmel. Fast edge integration. In S. Osher and N. Paragios, editors, *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer Verlag, 2003.
21. R. Kimmel and A. M. Bruckstein. Geometric edge detection. *IEEE Trans. on Pattern Analysis Machine Intel.* in press.
22. S. Lobbart and S. Markle. Dependent rendering: Visualizing multiple properties of an object. *Lecture Notes On Computer Science*, 25(26), 2002.
23. W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Computer Graphics*, 21(24), 1987.
24. L. M. Lorigo, O. Faugeras, W. Grimson, R. Keriven, R. Kikinis, C. F. Westin, and A. Nabavi. Codimension-two geodesic active contours for the segmentation of tubular structures. *Proc. of CVPR*, 2000.
25. R. Malladi and J. A. Sethian. An $O(N \log(N))$ algorithm for shape modeling. *Proc. of National Academy of Sciences*, 93:9389–9392, Sep. 1996.
26. R. Malladi, J. A. Sethian, and B. C. Vermuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on PAMI*, 17(2):158–175, Feb. 1995.
27. T. McInerney and D. Terzopoulos. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE Tran. Medical Imaging*, 18(10), 1999.
28. S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton jacobi formulations. *Journal Com. Phys.*, 79, 1988.
29. Philips. 4D Angio. <http://www.medical.philips.com/main/products/ct/applications/software/4dangio/>.
30. Philips. MxView Workstation. <http://www.medical.philips.com/main/products/ct/applications/mxview>.
31. J. A. Sethian. *Level Set Methods: evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 1996.
32. U. Tiede, T. Schiemann, and K. H. Hohne. High quality rendering of attributed volume data. *IEEE Visualization '98*, October 1998. Research Triangle Park, NC.