Volume Seedlings

Michael F. Cohen, James Painter, Mihir Mehta, Kwan-Liu Ma
Department of Computer Science
University of Utah
Salt Lake City, Utah 84112

Abstract

Recent advances in software and hardware technology have made direct ray-traced volume rendering of 3-d scalar data a feasible and effective method for imaging of the data's contents. The time costs of these rendering techniques still do not permit full interaction with the data, and all of the parameters effecting the resulting images. This paper presents a set of real-time interaction techniques which have been developed to permit exploration of a volume data set. Within the limitation of a static viewpoint, the user is able to interactively alter the position and shape of an area of interest, and modify local viewing parameters. A run length encoded cache of volume rendering samples provides the means to rerender the volume at interactive rates. The user locates and plants "seeds" in areas of interest through the use of data slicing and isosurface techniques. Image processing techniques applied to volumes (i.e. volume processing), can then automatically form regions of interest which in turn modify the rendering parameters. This "region growing" of "seedlings" incrementally alters the image in real-time providing further visual cues concerning the contents of the data. These tools allow interactive exploration of internal structures in the data which may be obscured by other imaging algorithms. Magnetic Resonance Angiography (MRA) provides a driving application for this technology. Results from preliminary studies of MRA data are included.

1 Introduction

Three dimensional scalar fields (or volumes) of data arise in a number of applications from computer simulation of physical phenomena to data gathered for medical diagnostic use via CAT scans and Magnetic Resonance. Rendering images directly from the volume has been demonstrated to be an effective method for visualizing such data [7, 9, 11, 12, 14, 19, 27, 29]. Volume rendering avoids many artifacts which may arise when intermediate graphics primitives are required [7, 13].

Volume images are constructed either in image order by sampling the volume along a ray from an eye point through the data or by projection of the data directly onto the pixel array. Differences in algorithms also deal with the order in which the samples are processed, either front to back as in ray tracing [12], or back to front analogous to a painter's algorithm. Ray tracing algorithms process a pixel at a time, while in projection techniques, a singe sample or data point may effect an area of pixels around the sample in some way via "splatting" [28], or projecting of a representative area onto the screen [22, 29].

As in earlier image synthesis techniques, acceleration methods focus on exploiting coherence in the image and in the data, and/or by progressively refining the image to provide rough results early in the rendering process [11, 15]. However, high quality images still require many seconds or minutes. Thus interactive exploration of volume data sets via these techniques is still not feasible. Changes in viewing parameters, mappings of data values to opacity or color, or enhancing regions of interest require complete new renderings.

The research presented here exploits the coherence across *all possible images from a given viewpoint* to provide interactive rendering rates for high quality images. The starting point of this algorithm is the volume ray casting technique as presented by Levoy [12, 15]. Earlier work in raytracing [21] has shown that a view dependent cache can be exploited to good effect when surface properties and light source intensities need to be adjusted while view position and geometry remain unchanged. In this paper we apply a similar idea to ray casting based volume rendering. The method described here caches rendering information at each sample point along each ray. The cache allows new images based on changes in rendering parameters to be generated *as the changes are made*, providing an interactive loop for volume exploration.

Local areas of interest within the volume can be indicated by the user planting a "seed" in the volume. Local rendering parameters can then be modified based on location relative to the seed. The basics of interactive use of local rendering modification through the use of "volume seeds" has been discussed in an earlier paper [17], and will be summarized in the next section. Problems in the earlier system included excessive storage requirements, and difficulty in placing and forming regions of interest.

The paper continues with a discussion of the use of coherence in the sample caching process followed by a description of new interactive positioning tools utilizing an integration of slicing and isosurface techniques. We then describe the use of image processing techniques generalized to volumes (volume processing) to automatically generate matte volumes modifying local rendering parameters. In this way, the seed sprouts into a "seedling" to enhance the rendering in connected regions of particular interest. Rendering parameters such as opacity are then based on minimum dis-

---

tance from the seedling. Image processing methods have been applied to volumes to *segment* the volume into discrete regions [18, 25, 26]. However, it should be noted that, in the application described here, the seedling itself is never rendered directly, but rather the volume of data continues to be rendered with directly, but rather the volume of data continues to be rendered with volumetric techniques with modified parameters based on position relative to the grown region.

Much of the motivation for the development summarized above has come from a medical application. In particular, imaging of data arising from Magnetic Resonance Angiography (MRA), in which the focus of attention is on exploring the vascular structure within the brain or other regions of the body. MRA techniques are used to diagnose malformations and aneurisms within the brain's blood supply, and to plan surgical and catheterization procedures. The intricate nature of the vascular structure as well as the somewhat noisy data capture require the ability to focus attention on specific vessels as potential anomalies are discovered. Results of the use of the above algorithms on this application will be presented and discussed.

2 Volume Seeds

Ray traced volume rendering involves sampling the data volume at evenly spaced points along a ray, computing a local illumination and opacity value and composing the result with earlier samples along the ray. Individual sample contributions are computed from trilinearly interpolated values from surrounding voxels, where each voxel contains a data value and an estimated gradient determined through finite differencing from its neighbors. The interpolated value and gradient provide arguments to mapping functions to determine color and opacity at the sample point. These color and opacity values may be a derived from a simple mapping from value to RGB (or opacity) or be determined from more complex statistical procedures intended to classify the likelihood of a particular material (e.g., bone, muscle) being present at a particular location [7].

The final illumination contribution at each sample point is computed from the color, opacity, local normal (estimated from the gradient of the data), and direction vectors to the eye and lights. These parameters to a Phong lighting model produce the final illumination at each sample point. Finally each sample illumination value is composited with earlier samples along the ray based on the accumulated opacity along the ray. Sampling can stop when the accumulated opacity approaches unity.

2.1 Image Coherence from a Static Viewpoint

By examining the volume rendering process described above, the required calculations can be broken into the two categories, those which are independent and dependent on mappings from *position and value* to *color and opacity*. Map-independent computations include:

- gradient calculation at voxels,
- determination of rays, and sample points along each ray,
- trilinear interpolation of data values and gradients to the sample points,
- the determination of local shading parameters, e.g. angles between view vector, light vector(s), and normal,
- and evaluation of a monochrome local lighting model, (i.e. independent of color and opacity).

Map-dependent calculations include only:

- mapping of data and position values to opacity and color,
- final evaluation of the local illumination,
- and compositing sample value illumination for final pixel color.

The above lists illustrate the fact that most of the computation is map-independent. However, the remaining map-dependent calculations leave a wide discretion for modification of the final image. This includes changes in:

- the mapping from value to color,
- the mapping from value (and/or the length of the gradient) to opacity,
- and position based variation in color and/or opacity.

By providing interactive tools to modify the local mapping to color and opacity, a user can create new renderings in interactive times (less than one to five seconds at 480x480 resolution on an SGI 240GXT). The locality of the mappings is controlled by the interactive specification of *matte volumes* [7]. By planting a *seed* at a point in the volume, opacity values can be modified as a function of the distance from the seed location. This allows the user to focus attention on particular regions of interest. By adding a binary decision indicating if the sample is in front or behind an imaginary plane through the seed, virtual cut-aways can also be produced in the same way. A final acceleration to the rendering process can be made by recognizing that only a local region of screen space will be effected by a new seed location when the matte volume is limited in size. Details of the matte volume functions and cut-away techniques can be found in Ma et al [17].

2.2 Sample Data Caching

The ability to quickly modify the image based on new matte volumes and the related mappings depends on caching the map-independent information at each sample point. This includes the partially computed illumination value and trilinearly interpolated data value for lookup into the interactively modified mappings. Unlike standard volume rendering, the storage of samples along a ray cannot stop when opacity reaches unity since opacity values can be changes interactively. The current implementation stores a two byte illumination value, and one byte data value per sample. Although the three bytes per sample is compact, this may require substantial memory, on the order of 150 Mb for a 500x500 image with 200 samples per ray. This problem can be largely ameliorated for most data sets by run-length encoding the sample values along each ray. In particular, if some range of values, e.g. zeros indicating empty space, can be a priori ruled as transparent, then both the storage and subsequent rerenderings can often be reduced by one to two orders of magnitude. The run length encoding is accomplished by stealing a bit from the 16 bit illumination value.

3 Seed Positioning

The need for the user to locate and position seeds to indicate areas of interest requires the ability to easily move and position a cursor in the three dimensions of the volume. Visual feedback for this process should provide clues both about the cursor's position and some indication of the volume's content to allow a seed to be placed near a region with a suspected anomaly. A multi-modal approach has been taken to serve these needs. Operations which can be performed smoothly in real time include manipulation of a rough 3D isosurface model and display of data on a slice through the volume.

Isosurface and slice display provide the basis for the user interface which has been developed. A low resolution isosurface is computed from a downsized data set by a polygonization algorithm [3, 16]. This provides enough detail to give the user a correspondence between the data set and what can be seen in the volume rendered image. A slice through the data volume orthogonal to the view direction indicates the depth position. A "screen door" transparency rendering of the slice permits continued view of the portion of the isosurface behind the slice. Finally, the coloration of the isosurface is based on distance from the slice plane, white away from the plane, and red where the plane slices the isosurface. Color plate 1 shows the full screen presented to the user in the Volume Seedlings system. The slicer/isosurface interface is in the upper left. Seeds can be deposited on the slicing plane which will then effect the subsequent rendering of the volume in the upper right.

Thus, the user can, in real-time, manipulate both the rotation of the isosurface and position of the slice plane. By pointing to some point on the resulting image, a seed is placed at the depth of the slice plane and in the location of the cursor. The volume rendering can then be modified based on the new seed location.

4 Volume Seedlings

A single seed highlights a spherical region around the seed point. In many applications, however, the shape of the region of interest within the volume is not strictly spherical, but rather is *data dependent*. The idea of Volume Seedlings is to use the seed point as a base from which to sprout a seedling along paths of "maximum interest", thus highlighting the region of interest.

Identifying regions of interest within the volume is closely related to the computer vision problem of identifying regions of interest within an image. Hence the seedling growth algorithm is similar to region growing algorithms described in the computer vision literature [1] and 2d seed fill algorithms described in the computer graphics literature [8, 24]. One important difference is a primary interest in the intermediate states of the growth process. Computer vision region growing algorithms are primarily concerned with a final segmentation of the image. A similar problem of extracting closed regions in a volume of data has been addressed by Miller et al [18],

The seedling growth algorithm used in the work presented here is voxel based. A *priority queue* [20] of voxels is maintained determining voxels within the volume which need to be explored. Initially, the priority queue contains only the user specified seed. At each growth step, the highest priority voxel is extracted from the priority queue and its 26 neighboring voxels are examined. The priority assigned to each voxel within the queue is based on the "degree of interest" of that voxel. We have currently experimented with linear combinations of three priority functions:

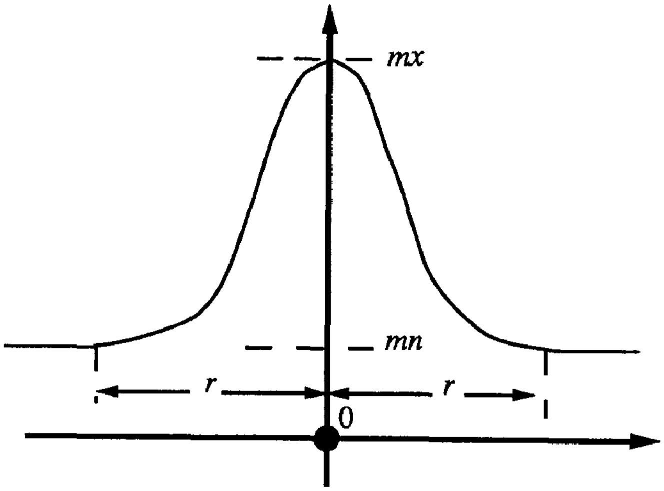

Figure 1: A graphical illustration of the opacity matte as a function of the distance from the seedling.

- classification based

  The priority of a voxel is based on its material classification. A voxel is given a higher priority the greater its percentage of some user specified desired material. This priority function encourages growth within regions of this material.

- gradient based

  The priority of a voxel is based on the magnitude of the gradient at the voxel. High gradient values indicated a surface boundary between materials so this priority function encourages growth along the surfaces boundaries.

- position based

  The priority of a voxel is based on the distance from the original seed point, thus encouraging growth of the seedling near the position indicated by the user.

Many other priority functions are possible.

In addition to the continuous priority function, a discrete test is used to eliminate many voxels. Thus, the neighbors are inserted in the queue only if:

- they haven't been visited before
- they pass an "eligibility" test

The eligibility test is not required but can significantly reduce the size of the priority queue by eliminating obviously uninteresting voxels. Currently, our eligibility test is a simple threshold on the priority, thus voxels are included in the queue only if their priority indicates at least a modicum of interest.

The seedling growth process yields a set of voxels, in priority order, defining a region of interest within the volume. The region of interest is highlighted through the use of an opacity matte as before for a single seed. The opacity matte volume is based on a function of the distance to the *closest* voxel of the seedling as illustrated in Figure 1.

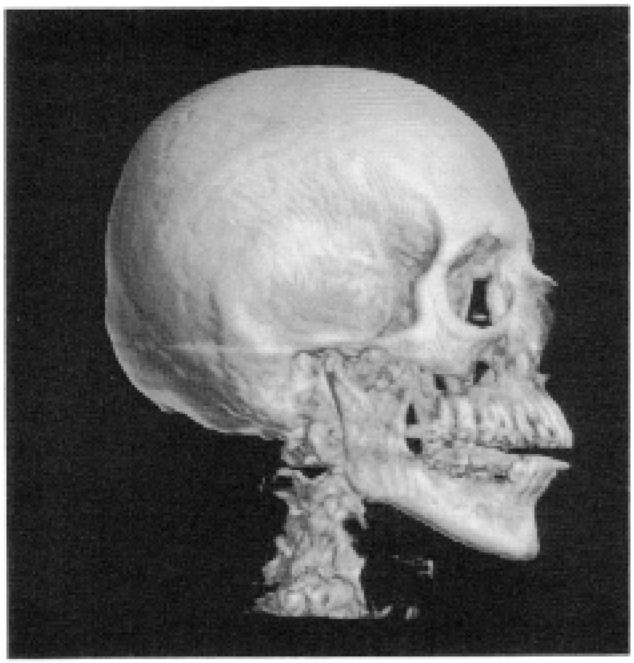

Figure 2: Traditional volume rendering of the UNC Chapel Hill CT head data.

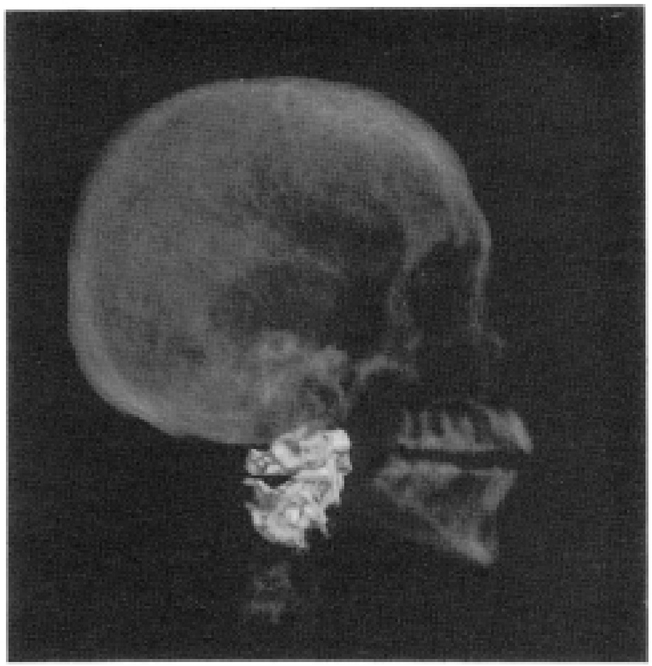

Figure 3: A seed point is used to highlight a spherical area in the data set.

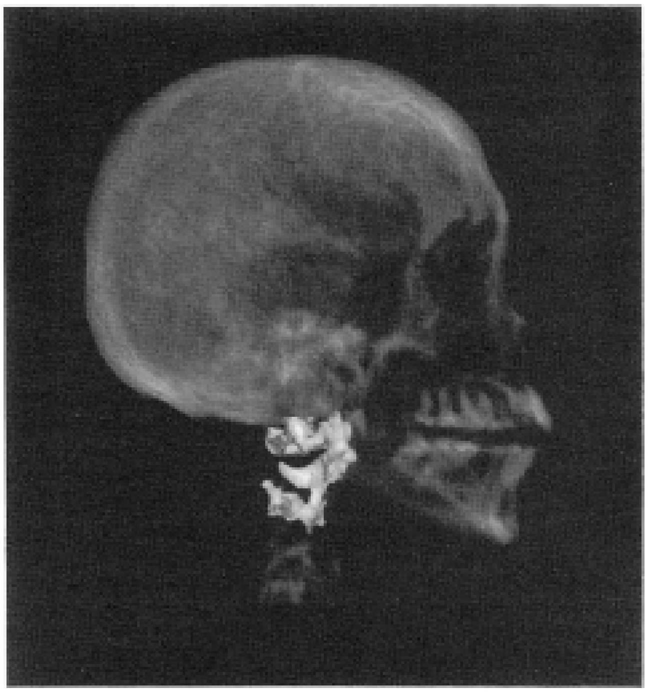

Figure 4: A gradient based seedling is used to highlight a structure in the data set.

The seedling opacity matte is computed according to the following formula:

$$\begin{aligned} \alpha(p) &= mn + (mx - mn) \cdot \beta(mindist(p,s), r) \\ \beta(d,r) &= \begin{cases} cos^2(\frac{\pi}{2} \cdot \frac{dist}{r}) & \text{if } dist < r, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

where *mindist(p,s)* is the minimum distance between any voxel of the seedling s and the sample point p in three-space. The mn, mx and r parameters are specified by the user.

In essence, r is used to control how wide an area the user wants to see. Surfaces outside this area should be semi-transparent or fully transparent, determined by mn. Mx is used to indicate how much enhancement is to be made to the area near the seed. Note that the opacity matte is never stored explicitly but is instead computed on the fly from the distance to the seedling.

By adding one additional byte to the sample cache to hold the distance to the nearest point in the seedling, images can be computed incrementally. As each new voxel of interest is extracted from the priority queue, only rays representing pixels which pass near the new voxel need to be processed. The minimum distance from a sample point to any point on the seedling is maintained by updating the distance only when the new point on the seedling is closer than any previously processed points (as in a Z-buffer algorithm).

Interactive changes can be made to the matte function mn, mx, and r parameters, as well as the color and opacity maps based on data value, without invalidating the cache. The rerenderings are thus very rapid due to the sample distance caching and the fact that only a small portion of the image space is affected by each new voxel added to the region of interest. The current implementation on an SGI 240GTX extracts new seedling points and rerenders the volume image approximately 10 times per second. This dynamic nature of the seedling growth also provides visual cues to the user.

Figures 2, 3, 4 illustrate the use of seeds and seedlings on the CT head set from the UNC Chapel Hill Volume Data Sets. Figure 2 is a normal volume rendering of the data set without the use of a seed. Figure 3 uses a seed point to highlight a spherical region in the neck area. Figure 4 shows a seedling grown from the same seed point using a gradient based seedling growth priority function. Figure 4 uses a smaller opacity matte radius than Figure 3 to focus on the seedling itself rather than a broader area around the seadling. Notice that Figure 4 does a much better job of isolating the region of interest.

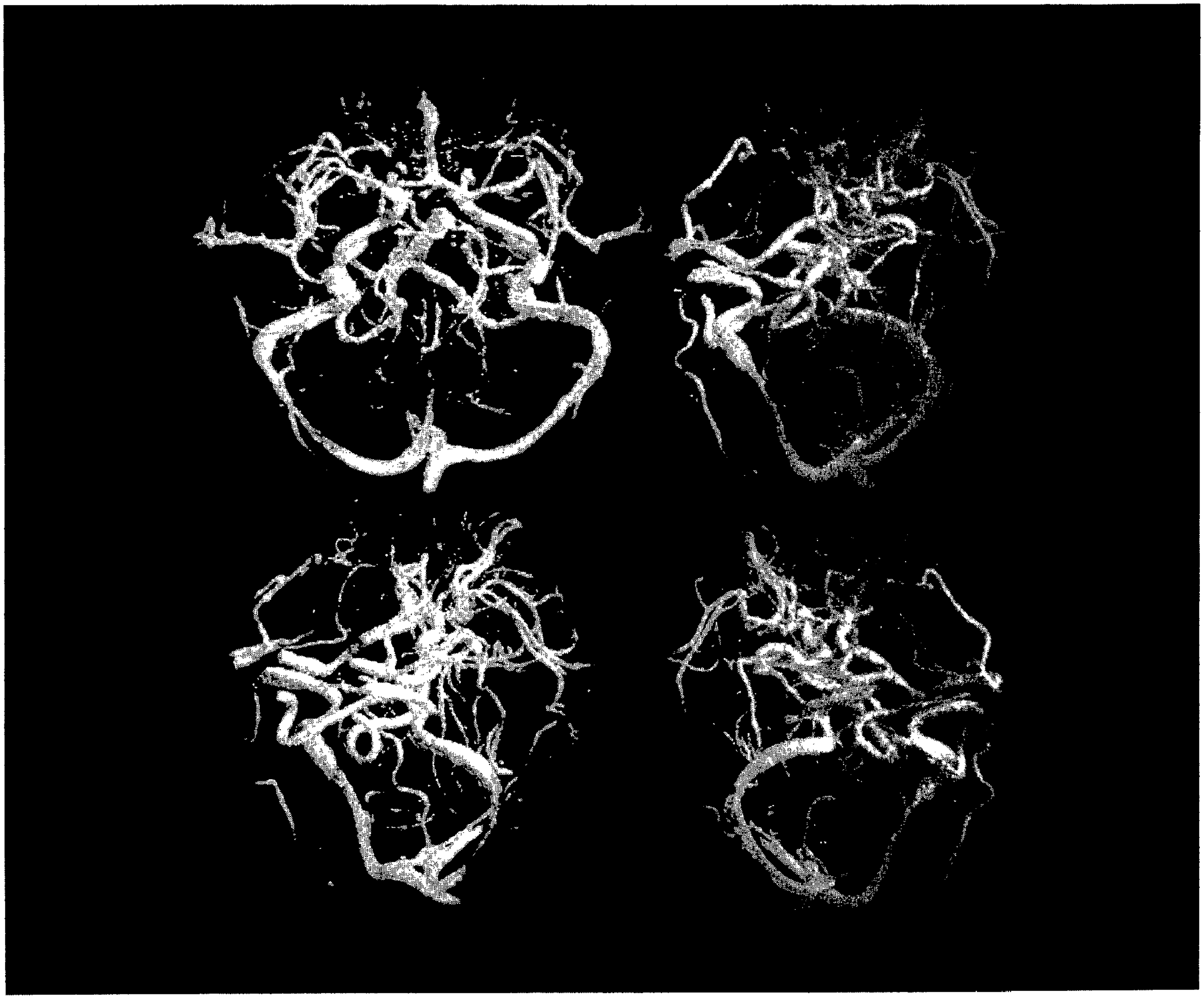

Figure 5: Four views of the MRA vascular data set.

5 Magnetic Resonance Angiography

Magnetic Resonance Angiography (MRA) is used to extract the vascular structure from within soft tissues like the brain. Visualizing the vascular structure can help in diagnosing malformations such as anuerisms and blockages, and/or help prepare surgical procedures such as catheterization through the vessels, or other invasive procedures designed to not disturb the vascular structure. The non-invasive nature of MRA over traditional angiography makes this diagnostic approach safer and thus more widely applicable. Unfortunately, MRA data capture cannot extract single vessels as can be done by selective dye release from a catheter in traditional angiography.

The vascular structure in MRA is captured by taking advantage of the fact that blood flows within the veins and arteries. The signal which is received is related to the time in which individual molecules are within the bounds of a thin slice through the body. Difficulties arise due to noisy data capture, or dropouts due to vessels which lie in the plane of excitation.

The most common visualization method used is a simple Maximum Intensity Projection (MIP) in which, as the name implies, a simple projection of the data onto a pixel grid is performed in which the pixel values are given the maximum value along a corresponding ray through the volume. A series of such images from different angles are viewed in succession to provide depth cues. However, single frames lose all or most of the depth information, and the sequence does not provide the full range of geometric information visible from more sophisticated algorithms.

The goals of the Volume Seedlings approach is to provide the three dimensional visual cues captured by volume rendering, while providing interactive tools to explore the data set and extract individual vessels for closer examination. Other work has been done in this area to extract single vessels through connectivity information. Cline et al selected a single voxel value and extracted all voxels of the same value connected to a seed point, and projected these voxels directly onto a screen [4, 5]. Other imaging techniques for MRA have been described as well [2, 6, 10, 23], however, not in the context of interactive systems with the

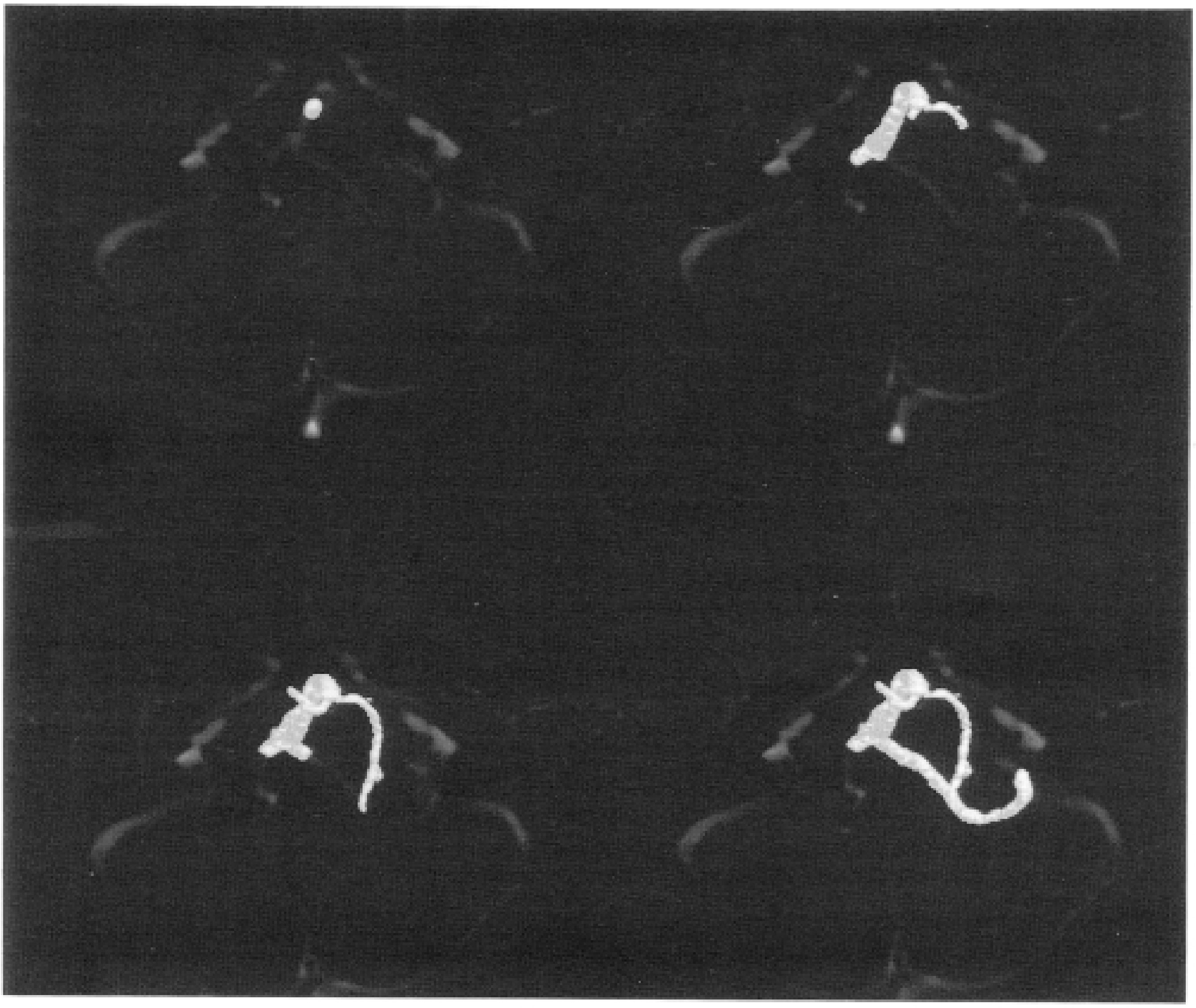

Figure 6: Four intermediate steps in the growth of a seedling.

use of volume rendering as the final imaging method.

Figure 5 shows four views of the vascular structure within the brain of a patient suffering from an anuerism. One single large seed in the center of the volume is used to capture most of the vessels while eliminating the vessels at the outer edges which complicate and obscure the interior. (These images are rendered at full 1K x 1K resolution as opposed to the 480x480 resolution in interactive mode.) After selection of a seed in the area of the anuerism a seedling is grown to extract the region of interest (Figure 6). The four images show the progress of the seedlings growth at four stages.

6 Conclusion

The ability to interactively isolate regions of interest within a volume rendering context has been discussed. By growing Volume Seedlings within the data set according to "interest" functions, features which may otherwise be hidden by the image complexity or by opaque regions can be examined. A description of an interactive volume exploration system has been described. Finally, the use of these techniques in the context of Magnetic Resonance Angiography to highlight individual vessels has been demonstrated.

The application of image processing and computer vision techniques to the problems involved in scientific visualization is an exciting area for exploration. It is expected that other more sophisticated region growing algorithms will be applicable in a wide variety of applications.

Acknowledgments

Partial support for this research was provided by a grant by International Business Machines for scientific visualization. Assistance in acquiring MRA data and other advice was given by Allen Sanderson. Dr. Wayne Davis and Dr. H. Ric Harnsberger of the department of Neuroradiology at the University of Utah supplied the MRA data and consulted on the development of the Volume Seeds system. Marc Levoy provided insight needed to interpret the UNC CT head data. Elena Driskill implemented an early version of the data slicer used in the interactive system. Video production assistance was provided by Robert McDermott and James Rose.

References

[1] Ballard, D. H., and Brown, C. M. *Computer Vision.* Prentice-Hall, Inc., 1982.

[2] Barillot, C., Gibaud, B., Scarabin, J., and Coatrieux, J. 3D Reconstruction of Cerebral Blood Vessles. *IEEE Computer Graphics and Applications 5*, 12 (Dec 1985), 13–19.

[3] Bloomenthal, J. Polygonization of Implicit Surfaces. *Computer Aided Geometric Design 5* (1988), 341–355.

[4] Cline, H., Dumoulin, C., Hart, Jr., H., Lorensen, W., and Ludke, S. 3D Reconstruction of the Brain from Magnetic Resonance Images using a Connectivity Algorithm. *Magnetic Resonance Imaging 5*, 5 (1987), 345–352.

[5] Cline, H., Lorensen, W., Herfkens, R. J., Johnson, G. A., and Glover, G. H. Vascular Morphology by Three-Dimensional Magnetic Resonance Imaging. *Magnetic Resonance Imaging 7*, 1 (1989), 45–54.

[6] Cline, H. E., Dumoulin, C. L., Lorensen, W. E., Souza, S. P., and Adams, W. J. Connectivity Algorithms for MR Angiography. *Book of Abstracts, Society of Magnetic Resonance in Medicine 1* (Aug. 1990), 59.

[7] Drebin, R. A., Carpenter, L., and Hanrahan, P. Volume Rendering. *Computer Graphics (Proceedings of SIGGRAPH 1988) 22*, 4 (August 1988), 65–74.

[8] Fishkin, K. P., and Barsky, B. A. A Family of New Algorithms for Soft Filling. *Computer Graphics (Proceedings of SIGGRAPH 1984) 18*, 3 (July 1984), 235–244.

[9] Kajiya, J., and Herzen, B. Ray Tracing Volume Densities. *Computer Graphics (Proceedings of SIGGRAPH 1984) 18*, 3 (July 1984), 165–174.

[10] Kraske, W. F., and Colletti, P. M. VOXAR 3-D Curvilinear Feature Description for Graphic Presentation and Analysis of MR Angiography(MRA) Aquisitions. *Book of Abstracts, Society of Magnetic Resonance in Medicine 1* (Aug. 1990), 59.

[11] Laur, D., and Hanrahan, P. Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering. *Computer Graphics (SIGGRAPH '91 Proceedings) 25*, 4 (July 1991), 285–288.

[12] Levoy, M. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications* (May 1988), 29–37.

[13] Levoy, M. A Taxonomy of Volume Visualization Algorithms, August 1990. SIGGRAPH 1990 Course Notes 11, Voulme Visualization Algorithms and Architectures.

[14] Levoy, M. Efficient Ray Tracing of Volume Data. *ACM Transactions on Graphics 9*, 3 (July 1990), 245–261.

[15] Levoy, M. Volume Rendering by Adaptive Refinement. *Visual Computer 6*, 1 (February 1990), 2–7.

[16] Lorensen, W. E., and Cline, H. E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics (Proceedings of SIGGRAPH 1987)*, 4 (July 1987), 163–169.

[17] Ma, K.-L., Cohen, M., and Painter, J. Volume Seeds: A Volume Exploration Technique. *The Journal of Visualization and Computer Animation* (1991 (To Appear)).

[18] Miller, J. V., Breen, D. E., Lorensen, W. E., O'Bara, R. M., and Wozny, M. J. Geometrically Deformed Models: A Method for Extracting Closed Geometric Models from Volume Data. *Computer Graphics (SIGGRAPH '91 Proceedings) 25*, 4 (July 1991), 217–226.

[19] Sabella, P. A Rendering Algorithm for Visualizing 3D Scalar Fields. *Computer Graphics (Proceedings of SIGGRAPH 1988) 22*, 4 (August 1988), 51–58.

[20] Sedgewick, R. *Algorithms.* Addison-Wesley, 1988.

[21] Séquin, C. H., and Smyrl, E. K. Parameterixed Ray Tracing. *Computer Graphics (Proceedings of SIGGRAPH 1989) 23*, 3 (August 1989), 307–314.

[22] Shirley, P., and Tuchman, A. A Polygonal Approximation to Direct Scalar Volume Rendering. *Computer Graphics (San Diego Workshop on Volume Rendering Proceedings) 24*, 5 (Nov. 1990), 63–70.

[23] Siebert, J. E., and Rosenbaum, T. L. Projection Algorithm Imparting a Consistent Spatial Perspective for 3D MR Angiography. *Book of Abstracts, Society of Magnetic Resonance in Medicine 1* (Aug. 1990), 59.

[24] Smith, A. R. Tint Fill. *Computer Graphics (Proceedings of SIGGRAPH 1979) 13*, 2 (August 1979), 276–283.

[25] Trivedi, S. S., Herman, G. T., and Udupa, J. K. Segmentation Into Three Classes Using Gradients. *IEEE Transactions on Medical Imaging 5*, 2 (June 1986), 116–119.

[26] Udupa, J. K. Interactive Segmentation and Boundary Surface Formation for 3-D Digital Images. *Computer Graphics and Image Processing 18* (1982), 213–235.

[27] Upson, C., and Keeler, M. V-buffer: Visible Volume Rendering. *Computer Graphics (Proceedings of SIGGRAPH 1988) 22*, 4 (August 1988), 59–64.

[28] Westover, L. Footprint Evaluation for Volume Rendering. *Computer Graphics (SIGGRAPH '90 Proceedings) 24*, 4 (July 1990), 367–376.

[29] Wilhelms, J., and Gelder, A. V. A Coherent Projection Approach for Direct Volume Rendering. *Computer Graphics (SIGGRAPH '91 Proceedings) 25*, 4 (July 1991), 275–284.