Volume Graphics (2007)
H. - C. Hege, R. Machiraju (Editors)

Automatic Feature Modeling Techniques for Volume Segmentation Applications

Runzhen Huang, Hongfeng Yu, Kwan-Liu Ma, and Oliver Staadt†

Department of Computer Science
University of California, Davis

Abstract

In many volume segmentation and visualization tasks, the ability to correctly identify the boundary surface of each volumetric feature of interest in the data is desirable. This surface can be used in subsequent quantitative studies of the segmented features. In this paper, we present an automatic approach to generate accurate representations of a feature of interest from volume segmentation. Our method first locates a set of points, which tightly define the boundary of the volumetric feature. This set of points can then be used to construct a boundary surface mesh. We also describe how to construct an anti-aliased volume representation of the segmented feature from this point set to enable high-quality volume rendering of the feature. These three representations – point set, boundary surface mesh, and anti-aliased volume segment – have a wide variety of applications.

1. Introduction

Volume segmentation is an important task in applications that relies on 3D imaging methods for nondestructive testing and evaluation. However, due to the limitations of a typical imaging process such as sampling resolution [YK01], the boundary between two materials is a partial volume which consists of voxels that receive contributions from multiple materials. This situation presents great challenges to the volume segmentation task, and is more severe if the two materials have low-contrast intensity values. A feasible solution is manual segmentation based on domain knowledge, which is commonly done for 2D cases, but for 3D the task is too tedious. What is desired is a robust segmentation method that can succinctly identify and construct the boundary of volumetric features with minimum user intervention.

One approach to this goal is to obtain segmentation results with voxel-level accuracy and automatically generate an accurate boundary from them. In this paper, we present such a new boundary construction method for volume segmentation. Figure 1 illustrates a volume segmentation and visualization pipeline. The step of feature extraction may be accomplished with any existing segmentation technique by voxel labeling. The boundary-finding step generates a set of new points defining the boundary surface that intersects or are close to the boundary voxels. The step of voxel identification may be done with region growing [HMMW03], clustering [SBC00], or graph cuts [BJ01]. In this work, region growing is used. The surface identification process is done based on the boundary model suggested in [KD98]. The resulting boundary points are then used in an anti-aliasing step to create a more precise volumetric definition of the feature. The boundary points can also be used to construct a tight boundary surface using a method similar to [HDD*92] but performs better in the presence of noise. To visualize the boundary surface, we can render either the boundary points directly [PZvBG00] or a constructed surface.

Our method generates three accurate boundary representations including a point set, an anti-aliased volume, and a geometric surface. These representations are derived from one mathematical boundary model and therefore share the same accuracy. Based on application demand, the user can select one of them for further analysis and processing. Thus, our method serves many applications.

2. Related Work

Accuracy is a key issue in volume segmentation because it determines the reliability of subsequent tasks such as vol-

† huangru, yuho, ma, staadt@cs.ucdavis.edu

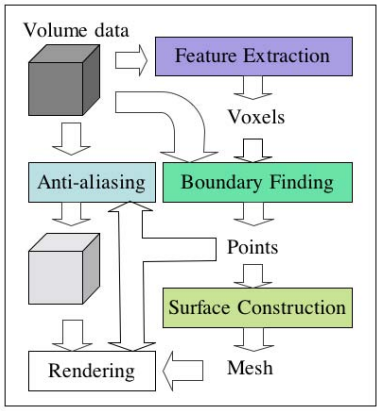

Figure 1: *The volume segmentation and visualization pipeline.*

ume measurements, medical diagnosis, and non-destructive testing [PXP98]. Current segmentation techniques can be roughly divided into two categories based on their accuracy. First, binary segmentation approaches like region growing and thresholding output a collection of voxels to represent the features of interest [GP00]. Second, more sophisticated methods such as deformable models [KWT88] and fuzzy segmentation [Dun74] give results with subvoxel accuracy.

Material boundaries are generally detected by thresholding gradient magnitudes or finding the zero-crossing of the second-order derivatives of the data [TMHG03]. These methods are fast and do not need priori information about the image. However, the former often forms unclosed boundaries while the latter is sensitive to noise. Edge linking algorithms [Loh98] that make closed boundaries and a technique called Laplacian of Gaussian (LoG) [MH79] have been proposed to handle these cases, respectively. In particular, Goshtasby and Turner [GT95] combine these two methods followed by a curve-fitting algorithm to extract the closed boundary of ventricular chambers.

Deformable boundary finding in volumetric data uses a deformable surface model to fit the boundary by balancing the influence of internal forces. These forces are determined by the model and the external forces obtained from the image data [XP00]. A priori knowledge in features can be incorporated into the models to achieve better results [MT00]. One popular deformable model is known as snakes [KWT88], which are planar deformable contours. Other deformable models use a geometric representation [MBL*91], finite element methods [MT95], and level sets [SZSD00]. Deformable boundary finding inherits the constraints from the model representations [HXP03]. In addition, deformable models that adapt their geometric changes to topology have a number of parameters, which need a lot of user invention [LT04].

Surface reconstruction from volumetric data sets or point clouds have been extensively researched [NBM05] [WOK05] [ACK01]. The existing approaches can be roughly divided into two categories: model-based and nonmodel-based methods. A more detailed survey can be found in [MM97].

3. Automatic Boundary Finding

As shown in Figure 1, volume segmentation begins with a feature extraction step which finds those voxels composing the feature of interest. In our system, this step is done with region growing through an interactive user interface [HMMW03]. The following step of boundary finding derives a set of points defining the boundary of the feature. We have developed a robust, automatic method for this task. The resulting points tightly mark the feature boundary with sufficient density. This boundary-finding method consists of the following steps:

1. Dilate the pre-segmented volume to also include layers of voxels external to the boundary region, which gives some information about the materials adjacent to the extracted feature.
2. Cast sampling lines from voxels inside the boundary through the boundary region into the adjacent material. Sample along those lines to construct a boundary function f for each of the materials adjacent to the feature.
3. Locate the boundary point on each sampling line using the corresponding f.

In the rest of this section, we use two synthetic data sets to illustrate the boundary-finding method and results. The first synthetic volume contains one sphere surrounded by one material, and the other has a cylinder adjacent to four different materials as shown in Figure 2. Gaussian noise was added to both data sets.

3.1. The Boundary Model

Kindlmann and Durkin [KD98] proposed a boundary model, and subsequently evaluated the model parameters from a histogram volume. Our method adopts the boundary model to represent the properties of the boundaries between a feature and its adjacent materials. Figure 3 shows the intensity function and its first and second derivatives defined on a sampling line passing through the boundary region perpendicularly. The equation of the function is:

$$v = f(x) = v_{min} + (v_{max} - v_{min}) \frac{1 + erf(\frac{x}{\sigma\sqrt{2}})}{2} \quad (1)$$

where x is a position on the line and v is the intensity value at x; v_{min} and v_{max} are the data values of two materials beside the boundary; erf is an error function, and σ is related to the boundary thickness. The first derivative of the function is:

$$f'(x) = \frac{v_{max} - v_{min}}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (2)$$

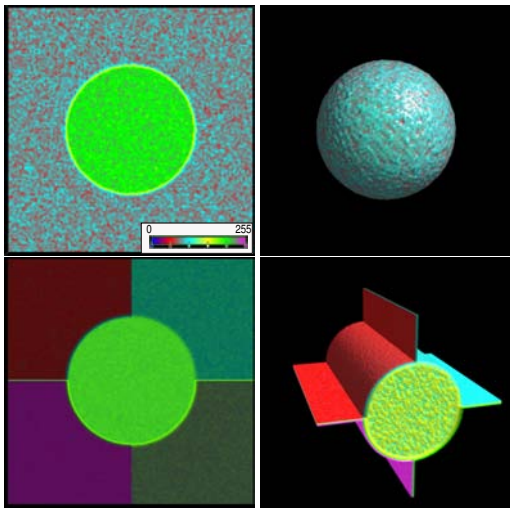

Figure 2: *Two synthetic noisy data sets with the size of* 128^3. *The top row shows the slice and shape of a sphere surrounded by one material; the bottom row gives the slice and shape of a cylinder adjacent to four materials.*

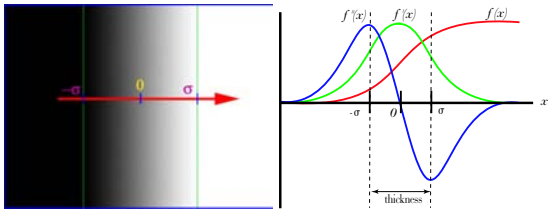

Figure 3: *The intensity function* f *and its 1st derivative* f' *and 2nd derivative* f'' *along a line perpendicularly going through the boundary region. Left: a line samples the boundary region; Right: the functions* f, f', *and* f'' *along the line.*

and the second derivative is:

$$f''(x) = -\frac{x(v_{max} - v_{min})}{\sigma^3\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{3}$$

From the extrema of values of f' and f'', σ used to determine the boundary thickness can be calculated as follows:

$$\sigma = \frac{f'(0)}{\sqrt{e} f''(-\sigma)} = -\frac{f'(0)}{\sqrt{e} f''(\sigma)} \tag{4}$$

The extrema of f' and f'' can be computed from the histogram volume indexed by the data values and their first and second derivatives.

3.2. Boundary Sampling

It is important to generate a sufficient number of sampling lines passing through the boundary region in order to obtain enough boundary points for representing the real boundary. Figure 4 illustrates the sampling scheme in 2D. P and Q

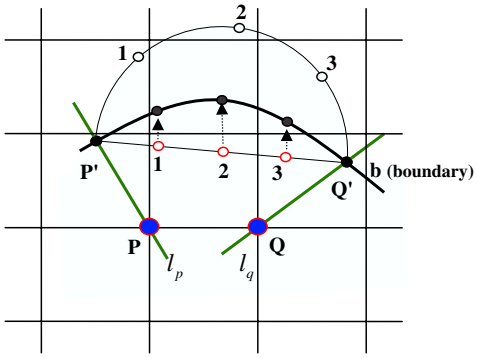

Figure 4: *The sampling scheme: two sampling lines* l_p *and* l_q *defined by two adjacent pre-segmented voxel P and Q have found* P' *and* Q' *on the boundary b; the number of new sampling lines through points on* $P'Q'$ *is determined by evenly partitioning the half circumference of* $P'Q'$ *into segments shorter than one voxel size.*

are two adjacent pre-segmented boundary voxels. Two sampling lines l_p and l_q, defined by P and Q and their gradient directions, are generated to find the points P' and Q' on the boundary b. It is possible that P' and Q' are far enough to miss some characteristics of b. However, since P and Q are adjacent and close to b, we can assume that the boundary segment between P' and Q' is simple. This assumption means that the boundary segment is convex and falls in the area enclosed by the line $P'Q'$ and the half circle connecting P' and Q'.

The scheme first divides the half circumference to a number of segments which have an equal length less than one cell size. This step determines the number of partitions in the next step – then the line segment $P'Q'$ is also evenly partitioned to the same number of parts. Finally the partitioning points on $P'Q'$ and their gradient directions are used to generate new sampling lines in the same way to find more boundary points.

In the 3D case, a quadrilateral is formed with four boundary points found with four adjacent boundary voxels. Each edge of the quadrilateral is partitioned using the above method and a sampling mesh is generated by connecting the partitioning points of opposite edges. Accurate boundary points are found by traversing all boundary voxels.

Based on assumption, we can conclude that the maximum distance from each obtained boundary point to its nearest neighbor is no more than one voxel size. For example, there are 116057 boundary points found on the sphere boundary, and the maximum distance from each point to the nearest neighbor is 0.7563.

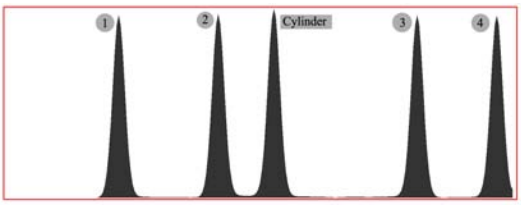

Figure 5: *The histogram constructed to detect the number of adjacent materials for the cylinder, as shown in Figure 2.*

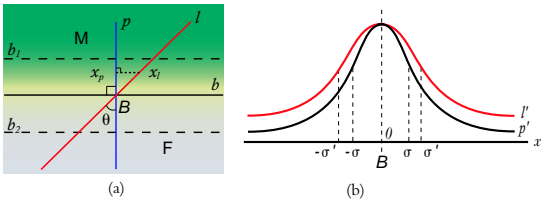

Figure 6: *Finding the boundary point B with a sampling line l. (a) l goes through the boundary region between M and F and intersects with the real boundary b at point B; the data values, first derivatives and second derivatives at x_p and x_l are the same when b is flat; (b) the curves of the first derivatives for p and l with the origin at B. p is perpendicular to b and therefore is represented with the boundary model. We simplify the finding process by fitting l' with p' derived from the histogram volume.*

3.3. Boundary Function Construction

The pre-segmented volume contains voxels close to the actual boundary. Using morphology, a dilated version of the volume is generated to include additional layers of voxels such that information about the materials adjacent to the corresponding feature is included. This dilated volume is used to find the number of adjacent materials and thus construct the boundary functions needed. The sampling approach described in the previous section is used to construct the boundary functions. The density of the sampling lines is more than that of the original volumetric mesh, which helps capture small adjacent materials.

The next step is to construct a 1D histogram of the sampled values along these lines. This histogram is used to determine the number of neighboring materials and their values. Figure 5 shows such a histogram for the cylinder data set in which we see four different materials surrounding the cylinder.

The data values sampled on a line are used to determine which material boundary the line has sampled. This is accomplished by comparing the sampled data values near the boundary region to the average values of the adjacent material. The boundary function for each adjacent material is constructed with the sampling lines passing through the boundary regions to sample the data values, as well as compute the first and second derivatives of the samples. These three values (f, f', f'') are then used to create a histogram volume, which is used to calculate σ using Equation 4. The σ of the sphere boundary computed from its histogram volume is 1.24099.

3.4. Boundary Points Finding

The boundary functions derived are used to find accurate boundary points within the dilated volume from sampling lines generated with the same scheme described in Section 3.2. Note that due to noise in the data, a sampling line is mostly not perpendicular to the actual boundary surface at the point it goes through. Fortunately, our method can handle this issue effectively based on our experiments and results. Figure 6(a) illustrates a 2D case where l is the sampling line passing through the real boundary b of M and F at point B, and p is a line perpendicular to b at B. Suppose the adjacent material M and the feature F have the average voxel value v_M and v_F, respectively, and $v_M < v_F$, then according to Equation 2 the first derivative of p is as follows:

$$p'(x) = \frac{v_F - v_M}{\sigma\sqrt{2\pi}} e^{-\frac{(x)^2}{2\sigma^2}} \tag{5}$$

where x is zero at B.

Assume the vicinity of B is a straight segment of b and the angle between p and l is $\theta \in [0, \frac{\pi}{2})$. The first derivative value of a point x_l at l is equal to the first derivative of the point x_p on p under the condition that x_p is a perpendicular mapping of x_l on p. Then we can obtain l' using Equation 5 as follows:

$$l'(x) = p'(x cos\theta) = \frac{v_F - v_M}{\sigma\sqrt{2\pi}} e^{-\frac{(xcos\theta)^2}{2\sigma^2}}, \theta \in [0, \frac{\pi}{2})$$

Let $\sigma' = \frac{\sigma}{cos\theta}$,

$$l'(x) = \frac{v_F - v_M}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma'^2}} \tag{6}$$

Since $cos\theta \leq 1$, we have $\sigma' \geq \sigma$. Figure 6(b) plots p' and l' where we can see that the maximum value of p' and l' is the same but l' is wider than p', or the same if $\theta = 0$. Because we need to find B at l whose origin is the sampling point used to generate l, we modify Equation 6 to:

$$l'(x) = \frac{v_F - v_M}{\sigma\sqrt{2\pi}} e^{-\frac{(x-x_B)^2}{2\sigma'^2}} \tag{7}$$

where x_B is the relative position of B at l.

Given a sampling line, we can use Equation 7 to fit the first derivative values on the line to find the boundary point with nonlinear least square fitting. Let $K = \frac{v_F - v_M}{\sigma\sqrt{2\pi}}$ and K is also different for each line. So strictly speaking, there are three variables σ', x_B, and K to be solved, which is very expensive. Fortunately, x_B can be found by only using the

boundary model without calculating σ' and K, and the error is small enough to be neglected. That is, instead of using Equation 7, we can modify Equation 5 to 8 and use it in the fitting operations:

$$p'(x) = \frac{v_F - v_M}{\sigma\sqrt{2\pi}} e^{-\frac{(x-x_B)^2}{2\sigma^2}} \tag{8}$$

There are 116057 boundary points found on the sphere boundary using our method. We calculate the radius (the distance between the boundary points and the center) using Equation 8. The radiuses are between 32.936 and 33.4351, the average is 33.19, and the standard deviation is 0.056. The numbers indicate that the points are accurate compared to the truth numbers which show that the average is 33.186 and the standard deviation is 0.059. Our simplified approach only uses 25.2 seconds, comparing to the method using Equation 7 that takes 278 seconds. In addition, the maximum distance from each point to the nearest neighbor is 0.7563.

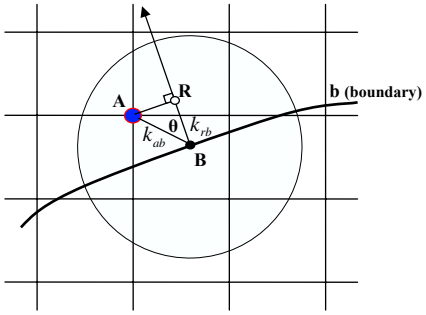

Figure 7: *Anti-aliased volume extraction: a fuzzy voxel around the boundary point B is calculated by first mapping its center A to R on k_{rb}, evaluating R's intensity value with the boundary model, and then computing the final value with intensity mapping.*

4. Anti-Aliased Volumetric Boundary Construction

Binary segmentation techniques introduce aliasing artifacts at the boundaries of extracted volume. Lakare and Kaufman [LK03] proposed an algorithm called *intensity flipping* to reconstruct the fuzzy volume around the boundary to remove the artifacts and generate better visualization. Their approach modifies the intensity of voxels on three voxel layers including the boundary layer and its two adjacent layers, so that the flipped values gradually change from the feature value to air. The equation of intensity flipping is:

$$\delta'_p = \delta_{AIR} + \frac{\delta_2 - \delta_p}{\delta_2 - \delta_1}(\delta_1 - \delta_{AIR}) \tag{9}$$

where δ_p and δ'_p are the original and new values of voxel p, respectively; δ_1 and δ_2 are the average voxel values of two regions inside and outside three layers around p; δ_{AIR} is the value of air, which is mostly zero.

The intensity-flipping algorithm works well for low-noise data sets. But it has problems when being applied to noisy data sets. In the latter case, suppose $\delta_2 > \delta_1$ and $\delta_{AIR} = 0$. When $\delta_p > \delta_2$ due to noise, δ'_p will be less than zero. On the other hand, when $\delta_p < \delta_1$, δ'_p will become bigger than δ_1. Both of these wrong flippings exacerbate the aliasing artifacts.

We propose a new method to derive an anti-aliased volume that is almost free from noise while preserving the boundary positions. First, the boundary points are grouped according to the boundary function they use. Second, for each boundary point, the values of its surrounding voxels are modified with the model function. Finally, the fuzzy region is generated with intensity flipping, which avoids sudden intensity changes from the adjacent materials to air.

Figure 7 illustrates the approach with a 2D case. In this example, B is one of the boundary points on b. We calculate the boundary region of B with a circle that is centered at B and has the radius equal to the boundary thickness defined with the σ of the model. All cells intersected with the circle are collected as the boundary region. For example, there are nine cells in B's boundary region, as shown in Figure 7. For each cell, we compute its new value by mapping its central point on the line p that is perpendicular to b at B. For instance, the new voxel value of A is evaluated using Equation 1 as:

$$v(A) = \begin{cases} v_F + (v_M - v_F)\frac{1+erf(\frac{|AB|cos\theta}{\sigma\sqrt{2}})}{2} & \theta \in [0, \frac{\pi}{2}) \\ \frac{v_F + v_M}{2} & \theta = \frac{\pi}{2} \end{cases} \tag{10}$$

where $|AB|$ is the distance between A and B.

The final fuzzy value of A is calculated from $v(A)$ by intensity flipping. Since the new value does not contain noise, wrong intensity flippings never happen. For voxels that are processed by more than one boundary point, the average value will be used.

This approach overcomes the disadvantage of intensity flipping. It is almost free from the noise effects. Figure 8 compares the pre-segmented sphere, anti-aliased sphere by intensity flipping and the sphere with our method. We can see that our approach generates a smoother result. In addition, on the slice by intensity flipping, some pink regions appear near the boundary because of wrong flippings.

5. Boundary Surface Construction

We have developed an algorithm to construct a boundary surface from the boundary point set. The approach is to create a distance field out of the point set such that the zero set is the boundary surface. The generated triangular mesh using the marching cubes is an approximation to the zero set. This construction process is similar to what is used in the anti-

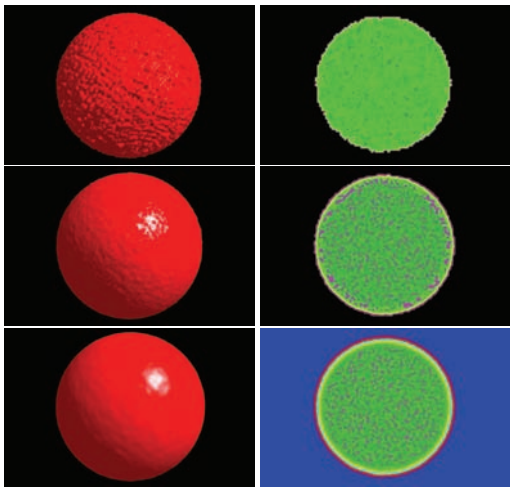

Figure 8: *Anti-aliased sphere extraction. The top row shows the pre-segmented sphere; the second row gives the anti-aliased sphere with intensity flipping; the third row illustrates the result with our approach.*

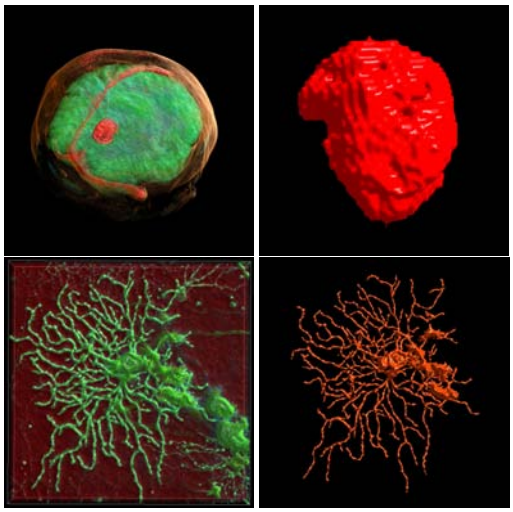

Figure 9: *Visualizations of the head and ganglion data set. Left: the whole volumes; Right: the pre-segmented features.*

aliased boundary volume construction. The difference is that it does not use a specific boundary model and does not consider surrounding materials. We evaluate the voxel values in the volume portion with a linear function instead of boundary model functions because the marching cubes use linear interpolation. Using the example in Figure 7, the voxel value of A is calculated with the following equation:

$$v(A) = v_{iso} + |AB| \cosh\theta \qquad (11)$$

where $|AB|$ is the distance between A and B, and v_{iso} is the designated value of boundary points. For each boundary point, the eight-voxel values of the cell it falls in are computed. For voxels that are processed by more than one boundary point, the average of all values is calculated. The boundary surface is obtained by applying marching cubes only to the portion of volume with isovalue v_{iso}.

We use this method to construct the sphere surface. The average radius of the triangle vertices is 33.197 and the radius standard deviation is 0.067, which is very close to the values of boundary points (33.19 and 0.059).

Our method is similar to [HDD*92]. The main difference lies on the construction of volume portion. Their approach goes through each voxel in the portion and evaluates its value with a signed distance function. This function is defined with the distance between the voxel and the tangent plane of the surface at the point closest to it. Our method traverses each boundary point and computes its contribution to the voxels of the cell it intersects. Our volume construction method thus has two advantages: it alleviates noise affects by averaging the contributions from neighboring points, and it provides a close approximation of the boundary surface by considering more neighboring points for each voxel. However, their approach can be more efficient if the number of voxels is less than the number of points, although it needs a specific data structure to organize the points to facilitate a nearest neighbor search.

6. Results

We have tested our methods with two real data sets including an MRI scan of a human head and a confocal microscopic imaging of ganglion. Figure 9 shows the visualization of whole data volumes and the pre-segmented features from the data sets.

The head data set has the size of 256 x 256 x 128 and contains a tumor which is the feature of interest. Our method finds 28982 boundary points. The top row of Figure 10 shows these points and the surface constructed from them. The bottom row shows a semi-transparent view of the anti-aliased tumor and a slice of the tumor where the constructed boundaries of dead cells inside the tumor are also visible.

The ganglion data set has the resolution of 110 x 1024 x 1024. This data set contains a tube-like feature with a lot of bifurcations. The top-left image in Figure 11 is the constructed anti-aliased volume of the ganglion. The other three images show the close-up views of the circled part in the top-left image. The top-right image is the pre-segmented volume, the bottom-left image is the boundary points, and the last one is the anti-aliased volume. We can see that the important neuraxon shape is well-caught by our approach.

Since the density of boundary points is ensured with our method, the boundary points can be used to directly depict

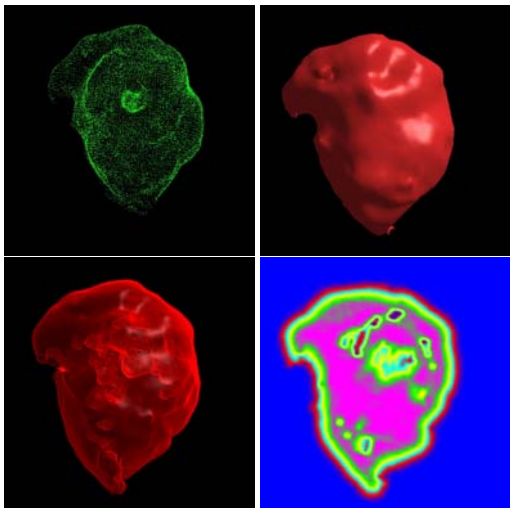

Figure 10: *The boundary of a tumor. The top row shows the boundary points and the geometric surface constructed from them; the bottom row displays the anti-aliased volumetric tumor and a slice revealing the internal dead cells' boundaries.*

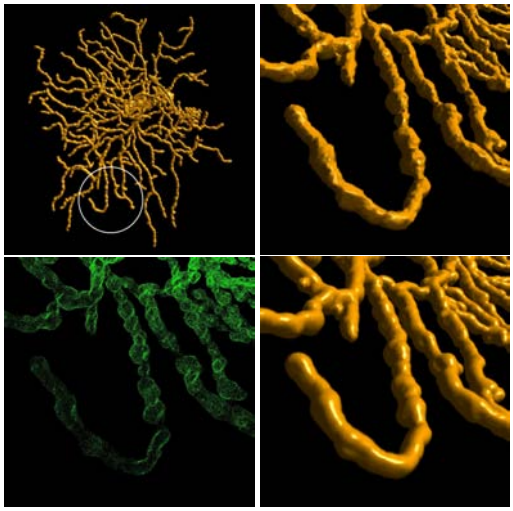

Figure 11: *The top-right shows the constructed anti-aliased volume of the ganglion. The other three images give a zoom-in part of the pre-segmented ganglion, the boundary points, and the anti-aliased volume, respectively. This part is circled in the top left image.*

the feature of interest for quick previewing using point-based rendering.

One problem we must address is that some sampling lines might not find boundary points, e.g. those for incomplete surface. In this case, we trust the pre-segmented segmentation and output the center of the boundary voxel. In addition, we use a filtering procedure to get rid of those points isolated from the boundary point set which might appear due to image noise [MT00].

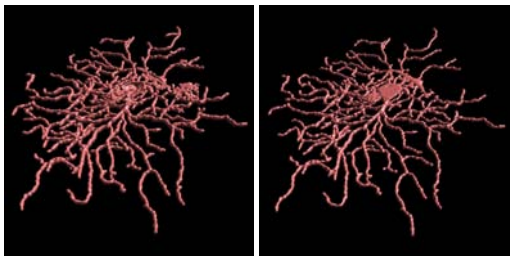

Figure 12: *Left: the surfaces of the ganglion generated with our method; Right: created with level set segmentation using itk. Our mehtod preserves better neuraxon shapes.*

Table 1: *Performance results*

Feature	Points	Time (sec.)		
		Modeling	Finding	Antialiasing
sphere	116057	5.547	25.181	3.949
tumor	28982	1.429	7.38	1.061
ganglion	1315477	43.274	204.60	45.404

Figure 12 compares the surfaces of the ganglion generated with our method and level-set segmentation approach in the itk toolkit (http://www.itk.org/). We can see that our method constructs the neuraxon shapes better than the level-set segmentation method because the latter tends to smooth surfaces, which also flattens the neuraxon shapes.

Table 6 shows the performance of our approach. We can see that boundary-finding operation takes dominant time. These numbers are obtained using a Dell PC with a 3.2GHz Pentium 4 CPU and 2GB memory.

7. Conclusion

This work has advanced the state of the art in volume segmentation in the following ways: 1) the development of an effective boundary finding algorithm by fitting sampled values based on a boundary model 2) proposing a simplification method to speed up the fitting while preserving the accuracy 3) developing a novel anti-aliased volume construction approach 4) a fully automatic approach to generate three accurate representations of volume segmentation results.

For future work, it would be helpful to extend the boundary model to address the issue of joint boundaries of more than two materials. Since the boundary point finding calculations for each sampling line are independent, programmable graphics hardware can be used to accelerate the boundary-finding procedure.

References

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust, unions of balls, and the medial axis transform. *Computational Geometry 19*, 2-3 (2001), 127–153.

[BJ01] BOYKOV Y. Y., JOLLY M.-P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proceedings of International Conference on Computer Vision* (2001), vol. 1, pp. 105–112.

[Dun74] DUN J. C.: A fuzzy relative of the isodata process and its use in detecting compact well-separated cluster. *Journal of Cybernetic 3*, 32 (1974).

[GP00] GOLDSZAL A. F., PHAM D. L.: Volume segmentation. In *Handbook of Medical Imaging: processing and analysis, edited by I. Bankman* (2000), Academic Press.

[GT95] GOSHTASBY A., TURNER D. A.: Segmentation of cardiac cine MR images for extraction of right and left ventricular chambers. *IEEE Transactions on Medical Imaging 14*, 1 (1995), 56–64.

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. *Comp. Graphics 26*, 2 (1992), 71–78.

[HMMW03] HUANG R., MA K.-L., MCCORMICK P., WARD W.: Visualizing industrial CT volume data for nondestructive testing applications. In *Proceedings of Visualization Conference 2003* (2003), pp. 547–554.

[HXP03] HAN X., XU C., PRINCE J. L.: A topology preserving level set method for geometric deformable models. *IEEE Trans. PAMI 25*, 6 (2003), 755–768.

[KD98] KINDLMANN G., DURKIN J.: Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of 1998 Symposium on Volume Visualization* (1998), pp. 79–86.

[KWT88] KASS M., WITKIN A., TERZOPOULOS D.: Snakes: active contour models. *The International Journal of Computer Vision 1*, 4 (1988), 321–331.

[LK03] LAKARE S., KAUFMANY A.: Anti-aliased volume extraction. In *Proceedings of the symposium on Data visualisation 2003* (2003), Eurographics Association, pp. 113–122.

[Loh98] LOHMANN G.: *Volumetric Image Analysis*. Wiley & Teubner Press, 1998.

[LT04] LACHAUD J.-O., TATON B.: Resolution independent deformable model. In *ICPR'2004* (2004), IEEE Computer Society Press, pp. 237–240.

[MBL*91] MILLER J. V., BREEN D. E., LORENSEN W. E., O'BARA R. M., WOZNY M. J.: Geometrically deformed models: A method of extracting closed geometric models from volume data. *Computer Graphics (Proc. of SIGGRAPH'91) 25*, 4 (1991), 217–226.

[MH79] MARR D., HILDRETH E.: *Theory of edge detection*. Tech. Rep. AIM 518, AI Lab MIT, 1979.

[MM97] MENCL R., MULLER H.: Interpolation and approximation of surfaces from three-dimensional scattered data points. In *Proceedings of Visualization Conference 1997* (1997), IEEE Computer Society, p. 223.

[MT95] MCINERNEY T., TERZOPOULOS D.: A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis. *Computerized Medical Imaging and Graphics 19*, 1 (1995), 69–83.

[MT00] MCINERNEY T., TERZOPOULOS D.: Deformable models. In *Handbook of Medical Imaging: processing and analysis, edited by I. Bankman* (2000), Academic Press, Inc.

[NBM05] NILSSON O., BREEN D., MUSETH K.: Surface reconstruction via contour metamorphosis: An eulerian approach with lagrangian particle tracking. In *Proceedings of Visualization Conference 2005* (2005), IEEE Computer Society, p. 52.

[PXP98] PHAM D. L., XU C., PRINCE J. L.: *A Survey of Current Methods in Medical Image Segmentation*. Tech. rep., Johns Hopkins University, Baltimore, 1998.

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: Surface elements as rendering primitives. In *Siggraph 2000, Computer Graphics Proceedings* (2000), Akeley K., (Ed.), ACM Press / ACM SIGGRAPH / Addison Wesley Longman, pp. 335–342.

[SBC00] SUTTON M. A., BEZDEK J. C., CAHOON T. C.: Image segmentation by fuzzy clustering: methods and issues. In *Handbook of medical imaging* (Orlando, FL, USA, 2000), Academic Press, Inc., pp. 87–106.

[SZSD00] STAIB L., ZENG X., SCHULTZ R., DUNCAN J.: Shape constraints in deformable models. In *Handbook of Medical Imaging: processing and analysis, edited by I. Bankman* (2000), Academic Press, Inc.

[TMHG03] THEUSSL T., MÖLLER J., HLADŮVKA J., GRÖLLER M. E.: Reconstruction issues in volume visualization. In *Data Visualization: The state of the art* (2003), Post F., Nielson G., Bonneau G., (Eds.), pp. 109–124.

[WOK05] WANG J., OLIVEIRA M. M., KAUFMAN A. E.: Reconstructing manifold and non-manifold surfaces from point clouds. In *Proceedings of Visualization Conference 2005* (2005), IEEE Computer Society, p. 53.

[XP00] XU C., PRINCE J.: Gradient vector flow deformable models. In *Handbook of Medical Imaging: processing and analysis, edited by I. Bankman* (2000), Academic Press, Inc.

[YK01] YOO T. S., KAKADIARIS I.: Volume segmentation. In *SigGraph 2001 course note for Image processing for Volume Graphics* (2001), pp. 176–191.