UC Davis
IDAV Publications

Title
RGVis: Region Growing Based Visualization Techniques for Volume Visualization

Permalink
https://escholarship.org/uc/item/3td71868

Authors
Huang, Runzhen
Ma, Kwan-Liu

Publication Date
2003

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

RGVis: Region Growing Based Techniques for Volume Visualization

Runzhen Huang Kwan-Liu Ma
Department of Computer Science
University of California at Davis
{huangru,ma}@cs.ucdavis.edu

Abstract

Interactive data visualization is inherently an iterative trial-and-error process searching for an ideal set of parameters for classifying and rendering features of interest in the data. This paper presents 3-d region growing based techniques that can assist the users to locate and define features of interest in volume data more quickly and more accurately. One technique employs partial region growing to generate a 2-d transfer function that effectively reveals the full features of interest. The other technique uses the result of full region growing to systematically construct a boundary surface for the extracted features. The resulting polygonal representation of the boundary surface can facilitate comparison, measurement, and simulation. A visual assessment method is suggested by using the extracted volume and surface information. These techniques either shorten or completely eliminate the typical trial-and-error step in the process of interactive data exploration.

1. Introduction

Increasingly, many studies in science and medicine produce volume data. Volume rendering is a powerful technique which can effectively reveal different aspects of the volumetric data content [3]. Over the past decade, various research efforts have gone into improving the quality, performance, and usability of volume rendering. In particular, the recent commodity graphics hardware support for interactive volume rendering has brought to us a desktop solution to volume visualization. While volume rendering will not replace the traditional surface rendering in work place, it is gaining growing popularity and becoming the primary method to use for visualizing volume data.

Volume rendering, however, is only the image synthesis step of a complete volume visualization process. Typically, there are a filtering step and a classification step prior to the rendering step, which could be followed by an assessment step. In this paper, we present our study on using region growing methods to assist the classification and assessment steps of volume visualization. We have applied a similar approach to the visualization of nondestructive testing data and obtained good results [6], which motivated us to refine our designs and conduct a more comprehensive study for more general volume visualization problems. The result is a new visualization technique which we call RGVis—Region Growing based Visualization. Our new investigations have focused on how the degree and extent of region growing impact the quality of transfer functions as well as the subsequent surface modeling. We have also developed an efficient and robust modeling technique, which is supplemented with an assessment method through visualizing uncertainty. We have built an interactive visualization system based on RGVis. This paper presents the results of applying RGVis to several data sets from medical applications.

2. Interactive Volume Data Exploration

A high-resolution volume data set can contain a wealth of information. The features which the scientists are interested in are generally related to some local density variation and can be either clustered or scattered. Even though the scientist might have some knowledge about the data content, due to the inherent complexity of the embedded structures, noise resulted from data acquisition, low contrast of some features, or large dynamic range of data values, it could take a great effort to locate a feature and effectively portray it by visualizing the data directly. A visualization system should thus provide to the user as much assistance and hints as possible during the whole course of data exploration and visualization.

The advent of hardware-accelerated volume rendering enables interactive feature finding and visualization. The latest graphics cards also support sophisticated programmable features which allow for the making of high quality, expressive visualizations. Hardware accelerated volume rendering is only limited by fill rates and the texture memory size of the video card since data must first be loaded into the texture memory before rendering can be performed.

If interactive rendering is possible, volume data exploration is generally achieved by editing color and opacity transfer functions [12]. Kindlmann and Durkin [7] introduced an intuitive way to derive opacity transfer functions based on the fact that the derivatives of data values suggest material boundaries. They show that by looking at a two-dimensional scatterplot of data values and gradient magnitudes (i.e., a 2-d histogram), opacity transfer functions can be easily defined to effectively capture features composed of boundaries between materials of relatively constant data value. Kniss, et al. [8] extended their work by introducing a set of direct manipulation widgets as the interface for defining multidimensional transfer functions for volume visualization. The concept of dual-domain interaction is particularly powerful and demonstrated with several effective visualization examples. However, to derive desirable classification and visualization result a trial-and-errors process is still needed.

We have developed an interactive volume visualization system which uses region growing to assist user in the data exploration and visualization process. As shown in Figure 1, the user interacts with the volume data through an interface which displays a slice and a 2-d histogram of the volume. The user slices through the volume and select a region of interest on the slice by plotting a line segment over the region. The system uses the points on the line segment to initiate a region growing process. The statistical properties of the region growing result are then used to derive a transfer function that would capture all related regions of interest in the volume. The central idea here is that a full region growing is generally not needed. The left image of Figure 2 displays the results of applying a partial region growing to a confocal microscopy ganglion data set which has 64×300×400 voxels. The partial growing took 0.03 seconds and 5,000 voxels were extracted. The left image was rendered using the derived 2-d transfer function.

We will discuss the relation between the degree and extent of region growing and the quality of the transfer function derived. In contrast, the dual-domain interactive method [8] takes a point specified by the user and then highlights the corresponding point on the 2-d histogram of the volume. The user would use this point as a starting place to edit a transfer function while simultaneously watching the resulting image.

Therefore, with our system the user only need to specify the region of interest, and a transfer function is suggested to the user. According to our test results, most of the time the suggested transfer function is close to the desired one, which if defined manually could take several iterations of editing over the 2-d histogram.

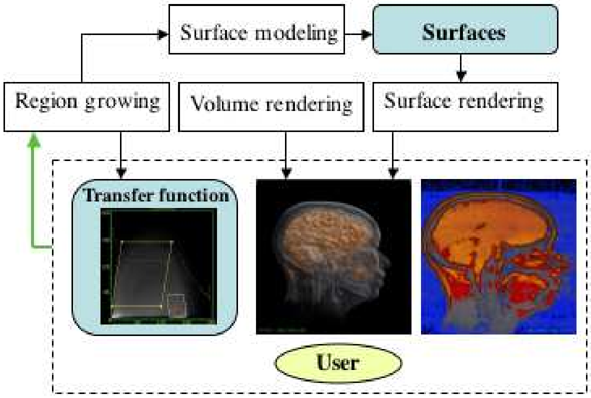


Figure 1. An interactive volume visualization system using region growing to assist the user to extract and visualize features of interest.

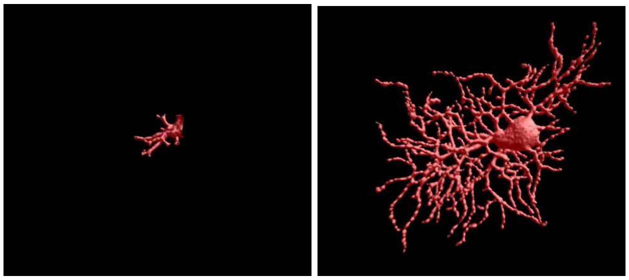


Figure 2. Left: Partial growing which takes 0.03 seconds. Right: Visualization of the full feature using the derived transfer function.

A full region growing collects the set of voxels completely representing the feature of interest. It is often useful to derive a boundary surface for the extracted feature. Such a surface representation can facilitate comparative studies, measurement, and quality assessment. Constructing an isosurface directly can miss many fine parts in the features. Our system can generate a boundary surface from the collection of data points out of a full region growing. Figure 3 compares an isosurface and a boundary surface of ganglia extracted from a confocal microscopic data set. The boundary surface represents the vessels more completely, including many many very fine ones. In addition, a hybrid rendering of the surface representation and the points allows us to assess the quality of the surface, as demonstrated in Section 5.

The rest of the paper describes how region growing works to extract features of interest, how a transfer function is defined with partial region growing, the surface model algorithm, and finally a quality assessment method.

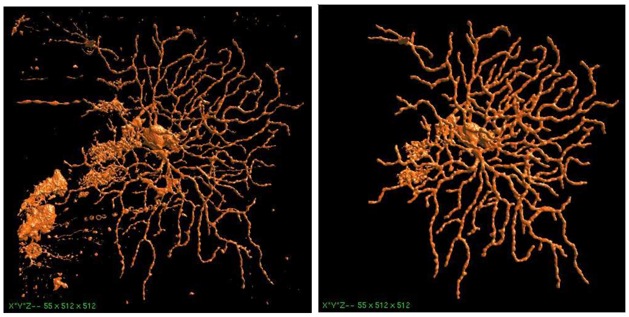


Figure 3. Surface visualization of a confocal microscopic ganglion data. Left: isosurface. Right: region-growing based surface extraction results in less ambiguous surfaces and leaves out all the undesirable parts.

3. A Region Growing Method

Region growing is a fundamental segmentation technique in image processing [13]. In the context of volume data segmentation, the goal of region growing is to map the input volume data into sets of connected voxels, called regions, according to a prescribed criterion which generally examines the properties of local groups of voxels [9]. The growing starts from a voxel in the proximity of the seed point selected by the user. The voxel can be chosen based on either its distance from the seed point or the statistical properties of the neighborhood. Then each of the twenty-six immediate neighbors of that voxel are visited to determine if they belong to the region. This growing expands further by visiting the neighbors of each of these twenty-six voxels. This recursive process continues until either some termination criterion is met or all voxels in the volume are examined. The result is a set of connected voxels determined to be located within the region of interest.

Feature finding thus becomes semiautomatic starting with an interactive seed point selection step, followed by the region growing process. As a result, the user only needs to find a few representative features and lets the region growing locate all features of similar properties in the same volume data. This approach is similar to the volume seedling method introduced by Cohen et al. [2]. In their work, region growing is used to help identify fine blood vessels in MRA volume data. More recently, Hahn et al. [4] develop a pipeline of 3D image processing steps to derive accurate models for visualization and exploration of vascular structures from radiological data. The resulting vessel models are used to study the branching patterns, for measurement, etc. Hu et al. [5] apply region growing to the segmentation of lung data to derive airway tree for surgical planning.

3.1. Seed point selection

Region growing begins from a seed point. In our system the user picks a seed by interactively slicing through the volume data. As soon as a slice is picked the user can move the cursor into the desired region and click to complete the selection. To further assist interactive seed point selection, some information about the seed voxel and its surrounding voxels is presented to the user. As shown in Figure 4, when a point is selected, it is highlighted on the scatterplot together with its 26 neighbors. This resembles to the dual-domain interface but more information is provided to the viewer. The coordinates of each highlighted point are the corresponding voxel's scalar and gradient magnitude values. A solid line connects the current point to each of its 26 neighbors. Another very important piece of information is the blue rectangle which shows the standard deviation of the data values (x-axis) and the standard deviation of the gradient magnitudes (y-axis) of the neighbors. Essentially, a tighter rectangular box suggests a region of high homogeneity. All these pieces of information provide the user some hints about the goodness of the selection.

Alternatively, the user can specify a line segment instead of a point. A line segment consists of a sequence 3-d points. Figure 5 shows the interface displaying some properties of those selected 3-d points divided into preclassified zones (i.e., the bottom color rectangles). The user can then choose a a seed point for subsequent region growing by selecting one of the preclassified zones. Again, such an interface offers the user more information assisting in the data exploration process.

3.2. Growing criteria selection

To do region growing, a set of criteria must be appropriately selected to effectively extract the regions. Possible criteria include region homogeneity and contrast with the background, strength of the region boundary, size, conformity to a desired texture or shape, and so on [13]. We have derived three criteria mainly based on region homogeneity and region aggregation using either data values or gradient magnitudes of the voxels. To make our region growing method more robust, at each voxel we also take into account some first-order statistical information about its 26 neighboring voxels.

Each criterion is defined by a *cost function* which is designed to extract a particular type of feature. If the value returned by the cost function is less than 1, then the voxel under consideration is within the region. Three functions are defined:

A. $f_{ca} = \frac{|v - v_s|}{k\sigma_v}$ where v is the data value of the current voxel, v_s is the data value of the seed voxel, k is a constant specified by the user, and σ_v is the standard de-

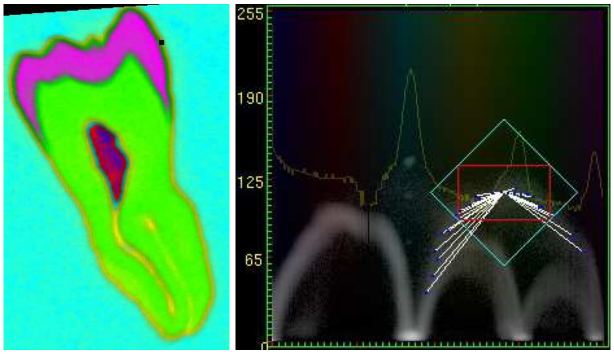


Figure 4. Left: A selected point. Right: Additional information displayed to assist seed selection.

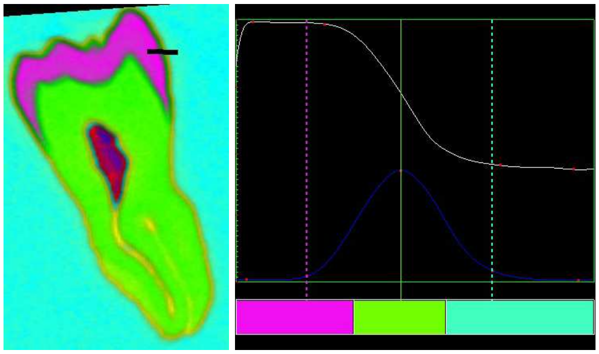


Figure 5. Left: A selected line segment. Right: Additional information displayed to assist seed selection. The white curve plots the data values of the points while the blue curve plots their gradient magnitude values.

viation of the values of the 26 neighboring voxels of the seed voxel. Note that $0 < k$. It is used to control the strictness of the criterion, and its default value is 1. This cost function exploits the data values. It works well for capturing homogeneous regions in which gradient magnitudes are nearly zero.

B. f_{cb} = $\frac{|g-g_s|}{k\sigma_g}$ where g is the gradient magnitude value of the current voxel, g_s is the gradient value of the seed voxel, and σ_g is the standard deviation of the gradient values of the 26 neighboring voxels of the seed voxel. This cost function exploits gradient magnitudes and is used to capture only the boundaries of a region.

C. $f_{cc}(p)$ = $f_{ca}p + f_{cb}(1-p)$ where p is a weight specified by the user or by the system. By default, p = $\frac{\sigma_g}{(\sigma_v+\sigma_g)}$. As a result, this cost function exploits both the data values and gradient magnitudes. It is more flexible than f_{ca} and f_{cb}.

We can make fcc adaptive that is by adjusting the criterion during the course of growing. Essentially, σ_v, σ_g, and p are continuously recomputed using the voxels collected so far. The user can choose to stick to a more conservative growing criterion by limiting the k value. While this adaptive approach would require a little more calculations, it frequently prevents from growing into erroneous regions and thus lowers the cost of growing.

Modayur et al. [11] design an adaptive cost function which takes into account the statistical information of the region extracted so far to handle the situation where small connections between two regions should be discarded. Basically, an adjacency criterion is employed in a postprocessing step but their cost function does not consider the gradients of voxels which are useful for growing boundaries.

3.3. Postprocessing

After region growing, a postprocessing step might be desirable to extend the results or to improve the results such as removing erroneous voxels or to fill gaps introduced by noisy data [6]. Depending on which criterion was used, the results of feature extraction may vary. For example, due to noise it is possible for a feature to grow into another through a very thin connection that is one- or two-voxels wide. A postprocessing filtering step can be performed to remove this type of erroneous connection. The lung structure displayed in both Figures 6 and 12 was postprocessed.

Since it is easier for the user to extract one feature at a time, our system allows the user to compose multiple extracted features into a single visualization, as depicted in Figure 6. The composition is done interactively through the user interface providing the user with multiple views. In this way, the user can extract and enhance each of the features completely independent of the other features in the same data set but present them in a single visualization for other specific purpose. The multiple views also allow the user to perform comparative visualization of similar features in the same volume or two different volumes.

4. Partial Region Growing and Transfer Function Generation

When the cost of region growing is high or there are multiple regions, it becomes desirable to perform partial region growing instead, especially during the exploration stage of volume visualization. That is, feature extraction is only done for a subset of the volume, or only a small subset of all the interested regions in the data is explicitly extracted. The system takes the partial region growing result to construct a classifier, and then a 2-d transfer function, for the visualization of the overall regions/features of the

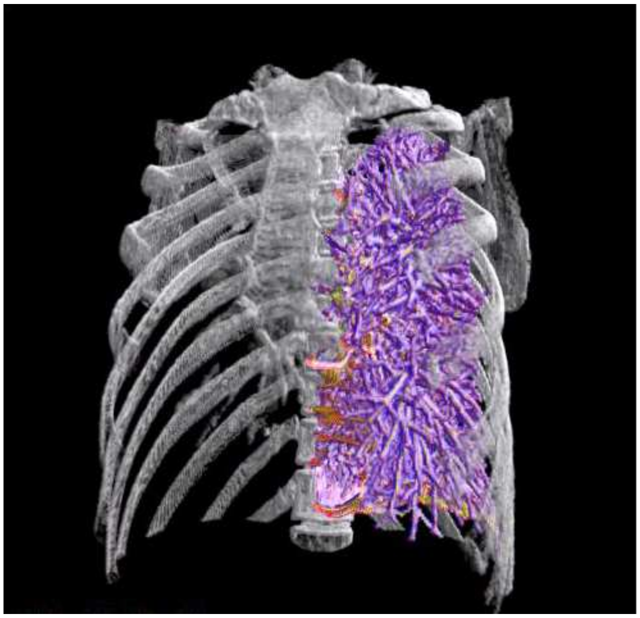


Figure 6. Two features, rib and lung, are extracted independently, and composed to form a single visualization.

same nature in the data. In a highly interactive visualization setting, such exploration procedure allows the user to very quickly browse through the features of interest in the data as completely as possible.

Figure 7 displays the image results of using partial region growing, derived transfer functions, and full region growing. The corresponding costs for each of the four data sets used are summarized in the table. All timing results were obtained on a PC with a 1.8GHz Pentium 4 CPU. In most cases the cost of a full region growing is too high for interactive visualization. Partial growing, on the other hand, would take only a small fraction of the time, and the wait time is generally only tens of milliseconds rather than several seconds. However, when using RGVis, the quality of the raw data has a significant impact on the resulting visualizations. For example, in Figure 7 we can observe some differences in the quality of the visualizations generated with the derived transfer function (classification) and the full region growing (segmentation). A cleaner structure is depicted in Figure 2 than those in Figure 7 because the corresponding data set is less noisy.

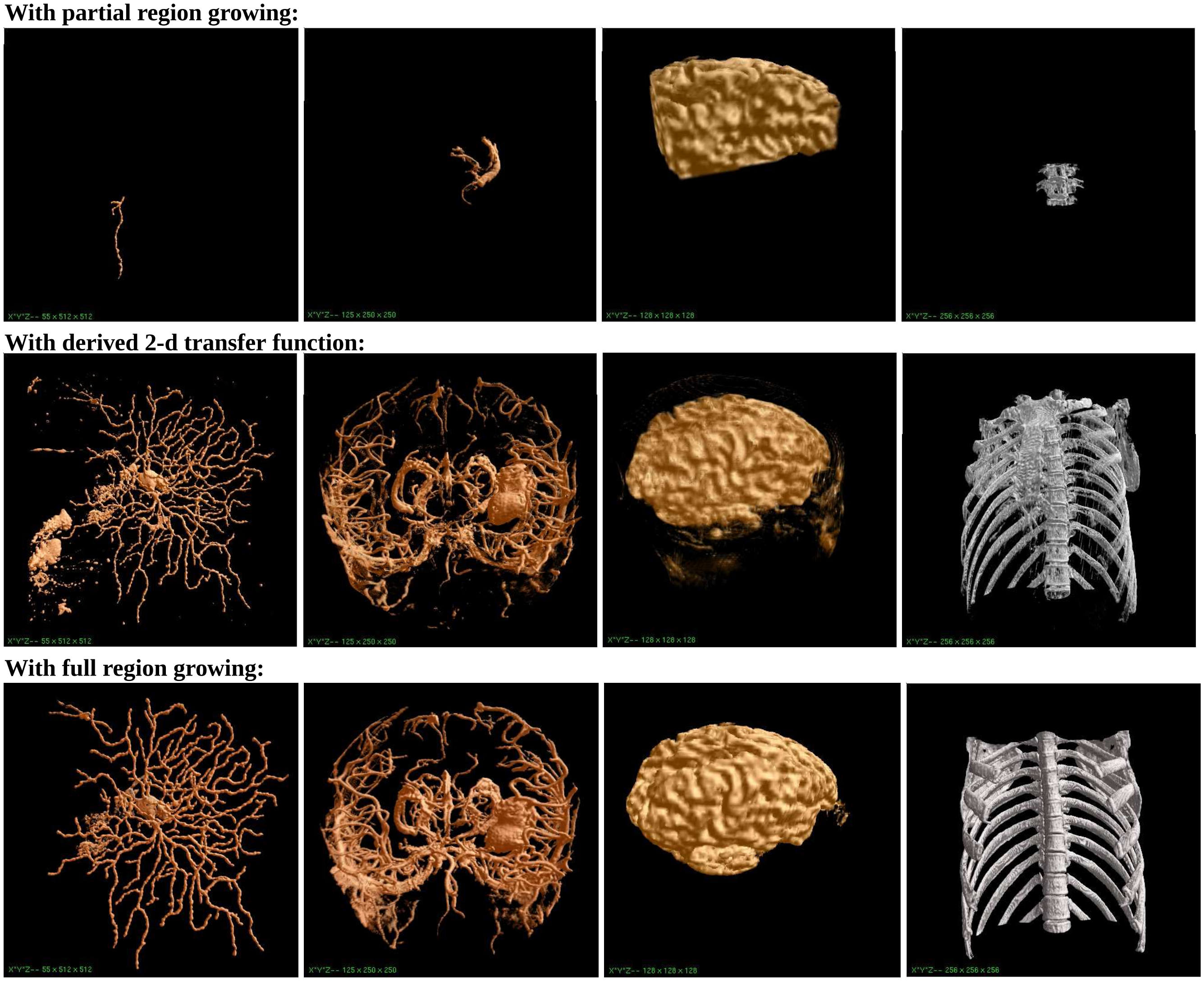
The derived 2-d transfer function can be generated by using the averages as well as standard deviations of the voxel values and gradient magnitude values. These four values define a rectangular region on the two-dimensional scatterplot of the data values and gradient values, which is the classifier we are looking for to capture the extended features. The two average values are used as the coordinates of the center of the rectangular region, and the two standard deviation values define the size of the region. We have used this method in our previous work [6].

Alternatively, a classifier can be generated directly in the feature's 2-d histogram space, and the resulting classifier is often more robust. The construction of the classifier starts with a k-d tree partitioning of the histogram region defined by partial growing. For each leaf node, the data points in the corresponding region are then sorted according to their values. Next, for each leaf node a local classifier is constructed by using the data points with values over a pre-selected threshold. Other voxels are discarded as noise. The following step recursively merges these local classifiers whenever appropriate by traversing the k-d tree. Two classifiers are merged into one if the new classifier can be constructed to contain most data points in these two classifiers while the new classifier's area is similar or less than the sum area of the original two. Figure 8 gives an example showing two classifiers automatically constructed to tightly capture a boundary surface. The corresponding k-d partitioning is shown in Figure 9. After generating the classifier, an opacity map and a color map are assigned to it for rendering the extracted features. An simple way is to use the feature's histogram data to construct the opacity map, in addition to using predefined ones like a linear or Gaussian function.

More information about the features of interest helps derive a better classification function. Figure 10 displays the difference of using a point, a small rectangular region, and a region obtained with partial region growing to construct a classification function. It is clear the transfer function derived from the region growing captures the enamel part of the tooth more completely. We have also conducted a study on how the degree and extent of a partial region growing affect the goodness of the derived transfer function. From the study results, we found that for most of the data sets we have used, a 20-40% growing is often sufficient for deriving a transfer function that is good enough. Figure 11 displays the rendered images using transfer functions derived from 20% and 80% growing. The two images are visually indistinguishable. It took under 0.2 seconds to reach 20% growing, and about 0.6 seconds to complete 80%.

5. Surface Modeling

Although volume rendering can generate images of the features of interest without first extracting any geometric information from the data, a geometric representation of the features allows for evaluation beyond visual inspection. The commonly used isosurface extraction method [10] based on a binary decision would not work well in general, particularly when the volume data is noisy or contains small, fine features.



With partial region growing:

With derived 2-d transfer function:

With full region growing:

Summary:

data	Confocal Microscopy ganglion	MRA head vasculature	MRI head	MRI Chest
volume size (voxels)	$55\times512\times512$	$125\times250\times250$	128^3	256^3
feature size (voxels)	224,316 (1.55%)	270,696 (3.46%)	123,571 (4.89%)	307,644 (1.83%)
time (seconds)	5.7	7.2	2.8	8.1
partial growing (voxels)	4,363 (0.03%)	5,000 (0.064%)	65,652 (3.13%)	20,001 (0.119%)
time (seconds)	0.022	0.028	0.34	0.12

Figure 7. Visualizations using partial and full region growing for four different data sets. The corresponding costs are summarized in the table. The feature size is the number of voxels resulted from full region growing. All timing results were obtained on a PC with a 1.8GHz Pentium 4 CPU.

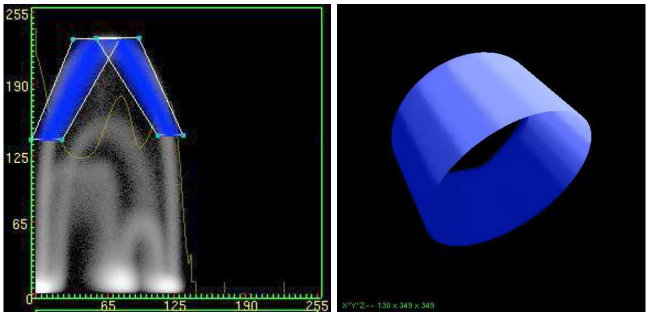


Figure 8. Left: Two classifiers on the 2-d histogram. Right: The corresponding visualization which shows a boundary surface from a CT nondestructive testing data.

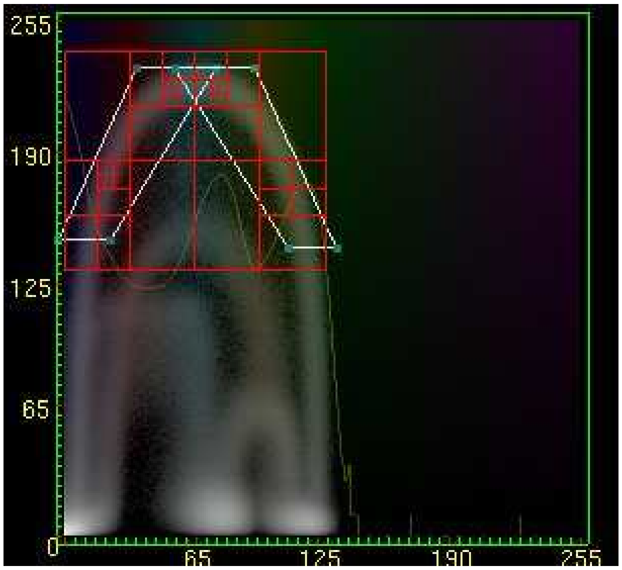


Figure 9. Kd-partitioning of the space for constructing classifiers. The starting rectangular region is the bounding box of the histogram of the extracted voxels from the partial region growing.

To construct boundary surfaces for the features, we make use of the segmented volume from a full region growing. A boundary tracking algorithm [1] is used to identify all the boundary faces which together enclose the extracted voxels. The resulting volumetric feature representation allows us to construct compact and hole-free boundary surfaces. The next step, which is optional, is to filter out high-frequency artifacts on the boundary faces. At this point, we could render the boundary faces directly by interpolating the faces' normals. However, this would only give us a better picture

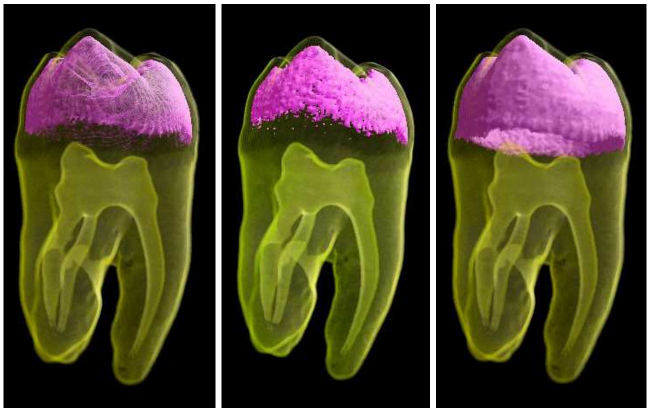


Figure 10. Left: Visualizing the enamel part of a tooth using information at a point. Middle: Using a tiny rectangular region. Right: Using the extracted region defined with partial region growing.

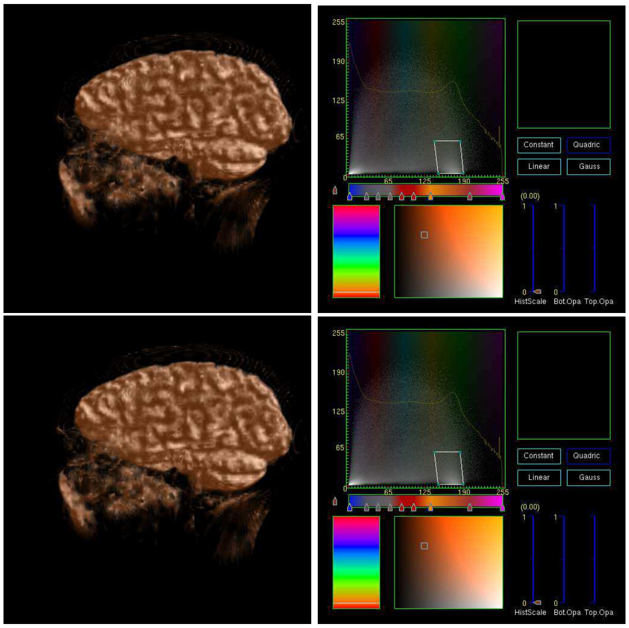


Figure 11. Top: The result of 80% growing. Bottom: The result of 20% growing. The two images look indistinguishable. The percentage was computed by dividing the number of voxels extracted with the partial region growing by the number of voxels extracted with a full region growing.

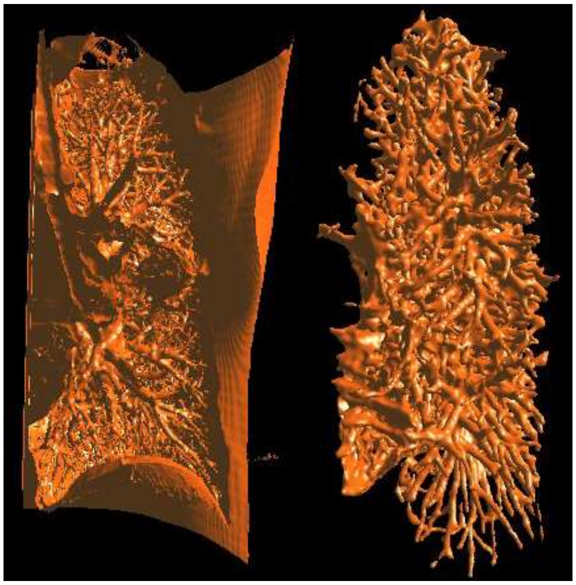


Figure 12. Left: Isosurface generated directly from a CT chest data set using Marching Cube. Right: Boundary suface of the lung from a CT chest data set derived using RGVis. The boundary surface is more useful than the isosurface.

of the boundary surfaces but not the actual more continuous boundary surfaces. The more desirable boundary surfaces can be extracted by first computing an isovalue, which can be the average of the values of those voxels on the boundary faces, and then constructing the corresponding isosurface using the marching cubes method [10]. The resulting surface is smooth while preserving the main topological information of the features, as shown in the right image of Figure 3. Figure 12 displays an isosurface and the extracted boundary surface of the lung from the MRI chest data set. Note that the isosurface also includes some surrounding unwanted parts while the boundary surface more closely captures the lung structure.

The quality of the boundary surface of an extracted feature can be assessed visually by hardware-accelerated rendering both the boundary surface and extracted voxels in a superimposed manner, as shown in Figure 13. Such an interactive visualization allows the user to examine what parts are left out in the surface representation. This technique can also be considered as a way to visualize uncertainty information. In Figure 13, those voxels outside the boundary surface are colored blue. The user can interactively change the color and transparency of either the surface or voxels to evaluate the surface.

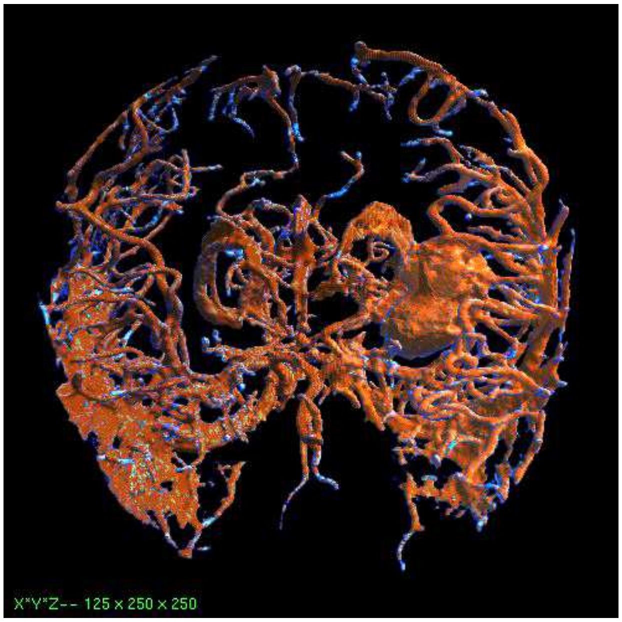


Figure 13. Superimposed rendering of the boundary surface and the extracted voxels to assess the quality of the surface. In this case, those voxels outside the boundary surface are colored blue.

6. Conclusions

The process of interactive volume data exploration typically involves an iterative searching for the desirable visualization parameters. This process is made even more difficult if the size of the data is large hampering interactive rendering, the volume data is noisy, or a manual search of several features is required. The objective of our work is to assist the user in the process of volume visualization. We want the user to focus on the data as much as possible, rather than the user interface artifacts. The user should not always need to go through a trial-and-error process to derive useful visualizations. A visualization system should do as much work as possible for the user, provide a lot of hints to the user for making some key decisions, and then generate the visualization results the user is after.

We have developed a new volume visualization approach—RGVis which is based on region growing. In addition to the traditional way of using region growing, a new concept introduced here is the automatic generation of a transfer function for the user by making use of the result of partial region growing. Essentially, the user only needs to point to the region of interest and the system does everything else for the user. The transfer function derived is generally good enough so the previously required searching effort is eliminated.

The surface modeling feature allows the user to operate on the extracted features such as making comparison, measurement, evaluation, and simulation. We plan to add a functionality to compute topological information from the surface representation and investigate how this information can be used in some application-specific tasks, such as diagnosis or surgical planning.

We have built a prototype system based on this region-growing approach and tested it on a variety of data. The results we have obtained so far are promising as demonstrated in this paper. We feel this new technique can be made more robust by further improving the growing criteria as well as the formulation of the transfer functions. We also intend to improve the efficiency of the system so it will become even more attractive to users. We plan to exploit the programmability of the latest commodity graphics cards to accelerate not only the rendering but also the visualization calculations.

Acknowledgments

This work has been sponsored in part by the U.S. National Science Foundation under contracts ACI 9983641 (PECASE award), ACI 9982251 (the LSSDSV program), and ACI 0222991; the U.S. Department of Energy under Memorandum Agreements No. DE-FC02-01ER41202 (SciDAC program) and No. B523578; and the National Institute of Health through the Human Brain Project. The MRI chest data set was made available through Dr. H. Miyachi at Kubota Co., Japan. The two confocal microscopic data sets were provided by Dr. Leo Chalupa at the Division of Biological Sciences of the University of California at Davis. The aneurysm data set was provided by the SCI group at the University of Utah. Los Alamos National Laboratory provided the CT nondestructive testing data set. The MRI head data was provided by Siemens Medical Systems, Inc. through the University of North Carolina at Chapel Hill. The authors also like to thank members of the UCD visualization and graphics group for the valuable discussion and providing some of the test data sets.

References

[1] E. Artzy, G. Frieder, and G. T. Herman. The theory design, implementation and evaluation of a three-dimensional surface detection algorithm. *Computer Graphics and Image Processing*, 15:1–24, 1981.

[2] M. Cohen, J. Painter, M. Mehta, and K.-L. Ma. Volume seedlings. In *Proceedings of the 1992 ACM Symposium on Interactive 3D Graphics*, pages 139–145, 1992.

[3] K. Frenkel. Volume rendering. *Communications of the ACM*, 32(4):426–435, April 1989.

[4] H. K. Hahn, B. Preim, D. Selle, and H.-O. Peitgen. Visualization and interaction techniques for the exploration of vascular structures. In *Proceedings of Visualization 2001 Conference*, pages 395–402, 2001.

[5] S. Hu, E. A. Hoffman, and J. A. Reinhardt. Automatic lung segmentation for accurate quantitation of volumetric x-ray ct images. *IEEE Transactions on Medical Imaging*, 20(6):490–498, June 2001.

[6] R. Huang, K.-L. Ma, P. McCormick, and W. Ward. Visualizing industrial ct volume data for nondestructive testing applications. In *Proceedings of IEEE Visualization 2003 Conference (to appear)*, October 2003.

[7] G. Kindlmann and J. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of 1998 Symposium on Volume Visualization*, pages 79–86, 1998.

[8] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.

[9] G. Lohmann. *Volumetric Image Analysis*. Wiley & Teubner Press, 1998.

[10] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of SIGGRAPH '87*, pages 163–169, July 1987.

[11] B. Modayur, J. Prothero, G. Ojemann, K. Maravilla, and J. Brinkley. Visualization-based mapping of language function in the brain. *Neuroimage*, 6(4):245–258, 1997.

[12] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. Sobierajski Avila, K. Martin, R. Machiraju, and J. Lee. The transfer function bake-off. *IEEE Computer Graphics and Applications*, 21(3):16–22, May/June 2001.

[13] A. Rosenfeld and A. Kak. *Digital Picture Processing, Volume 2*. Academic Press, second edition, 1982.