Interactive tools for image segmentation

M. Jackowski*, A. Goshtasby* and M. Satter†

*CSE Dept., Wright State University, Dayton, OH 45324

†Wallace-Kettering Neurosci. Inst., Kettering Med. Ctr., Dayton OH 45429

ABSTRACT

Interactive tools for segmenting 2-D and 3-D images are presented. These tools allow a user to quickly revise a segmentation result obtained from an automatic method. A thresholding technique is described that finds a unique threshold value for each homogeneous region in an image. The threshold value is found such that variance in the region is minimized under change in the threshold value. Curve- and surface-fitting methods are described that can accurately represent a region boundary in 2-D or 3-D with a parametric curve or a surface, respectively. A curve or a surface is optimized to minimize the number of control points representing a region with a prescribed accuracy. The optimized curve or surface is then revised by moving its control points interactively. Once a curve or a surface is found to accurately enclose a region of interest, it is quantized to produce the final 2-D region contour or 3-D region surface. These interactive tools can be used to revise unsatisfactory results obtained from any automatic segmentation method.

Keywords: Image segmentation, interactive segmentation, iterative thresholding, RaG curve, RaG surface

1. INTRODUCTION

Image segmentation is the process of partitioning an image into meaningful parts. Often, however, image segmentation techniques partition an image into regions of homogeneous properties, or regions bounded by sharp intensity changes. For a region to be meaningful, a user may need to interactively revise it to accurately delineate an object of interest in an image. In this paper, we will introduce interactive tools that will enable a user to effectively revise segmentation results in 2-D and 3-D.

Interactive tools are needed because automatic segmentation methods often produce errors, and in many applications it is essential that accurate object boundaries are obtained before data analysis can begin. Interactive tools should enable a user to revise the result of an automatic segmentation quickly and easily.

Monitoring changes in tumor volumes represents a sensitive method for determining recurrence from post-surgical or radiation changes and can give an early indication as to the efficacy of treatment. These volumes can be based on anatomical imaging (MRI or CT) or biochemical imaging (PET). However, unless a physician can generate such volumes quickly and easily, quantitative tumor volume monitoring will not become a standard of care. The tools described in this work will be employed in a research protocol for measuring both biochemical and anatomical brain tumor volumes during the course of immunotherapy with genetically modified brain tumor cells (glioblastoma multiforme). What is learned in this research regarding the rapid generation of tumor volumes will then be implemented into standard clinical care.

Interactive tools for image segmentation have been proposed before. Kundu[7] developed an interactive thresholding method with which a user could interactively change the threshold value until it was visually confirmed that a region of interest was properly isolated. Cabral *et al.*[2] developed a region-growing method for segmenting volumetric images that was coupled with editing tools to revise the segmentation results. Hinshaw and Brinkley[5] developed a 3-D shape model that used prior knowledge of an object's structure to guide the search for detecting the object. Object structure was defined interactively by a graphical user interface.

Höhne and Hanson[6] developed low-level segmentation functions to interactively extract regions of interest. The functions developed were based on morphological operators. Pizer *et al.*[8] segmented a volumetric image into a hierarchy of visually sensible regions with different resolutions. The intention, by pointing to an object on a cross-sectional image, was to select the corresponding region and revise it. Welte *et al.*[10] developed an interactive method

Send correspondence to A. Goshtasby, E-mail: agoshtas@cs.wright.edu

for separating vessels from each other and from the background in MR angiography volume data. In order to reduce the complexity of the displayed structures during interactive segmentation, a capability to interactively select the subtrees was provided.

Barrett and Mortensen[1] developed a semiautomatic segmentation method known as "live-wire." With this method a user can click on a point on the boundary of a region and drag the cursor roughly around the outline. An automatic process will then find the best path according to a predefined criterion from the start point to the current cursor position. Segmentation is achieved in a fraction of the time needed to trace the boundary by hand. This method was used to segment ultrasound as well as CT and MR images. Schiemann *et al.*[9] developed an approach that allows a user to segment image volumes interactively using thresholding and 3-D morphological operations.[6] The segmentation is performed concurrently with 3-D visualization, providing direct visual feedback to assist the user segment an image.

The tools introduced in this paper are based on parametric curves and surfaces, representing segmentation results in 2-D and 3-D. By moving the control points of a curve or a surface interactively, the segmentation results are revised.

Interactive tools are known to be very effective in design and editing of industrial parts. We have developed similar tools for interactively revising or editing segmentation results. Regions in an image can be considered free-form shapes that need to be revised to desired shapes. We will introduce tools that are based on rational Gaussian (RaG) curves and surfaces.[3] RaG curves and surfaces can represent very complex shapes at desired levels of detail.

In the following, first a method to segment 2-D images is presented and use of RaG curves to revise the segmentation results is outlined. Then a method to segment 3-D images is described and use of RaG surfaces to refine the segmentation results is presented. An optimization process to minimize the number of control points in RaG curves and surfaces is also given.

2. 2-D ITERATIVE THRESHOLDING

To demonstrate how our 2-D interactive tools work, we first describe an automatic thresholding method. Then we show how to revise the segmentation results interactively.

Intensity thresholding is a quick method for image segmentation, but thresholding is known to produce errors if the regions to be extracted are not completely homogeneous. Consider an image containing a brain tumor (Fig. 1a) or an image containing a skin lesion (Fig. 1b). In such images, thresholding can approximately isolate the tumor or the lesion from the background. If the threshold value is gradually changed, the size of the region changes and at a certain threshold value the region best representing the tumor or the lesion will be obtained. If we plot the region size against the threshold value, we will observe that as the threshold value is increased (decreased) with uniform steps, change in region size will decrease up to a point and then increase sharply when two neighboring regions merge. The threshold value at which the smallest change in region size is obtained by changing the threshold value is the threshold value at which the region is most stable and doesn't change with variations of the threshold value. We will take such a threshold value as the optimal threshold value for our automatic segmentation method.

To locate bright (dark) regions in an image, the image is smoothed, and locally maximum (minimum) intensities are used as seed points. A region containing one or more seed points is then tracked as the threshold value is changed to determine variations in its area size.

The algorithm to find the threshold value for isolating a bright region in a dark background is summarized below:

1. Smooth the image with a rather wide Gaussian and find local intensity peaks in the smoothed image. Take the local peaks as the seed points.

2. Starting from threshold value 1.0, decrement the threshold value with steps of 0.1 until 0.0 is reached and at each threshold value find the regions.

3. For each seed point, track the region containing the seed point in the thresholded images and determine the difference of region sizes at the threshold values.

4. Find two consecutive threshold values t and $t + 0.1$ such that change in region size is minimum. Then set optimal threshold value to $t + 0.05$. This threshold value determines the region corresponding to the provided seed point.

5. Repeat steps 3 and 4 to extract the remaining regions in the image using the other seed point. If a region contains more than one seed point, once the region is obtained, all seed points inside it are removed.

The 0.1 increment in threshold value is arbitrary and can be decreased to improve the segmentation accuracy. Experiments on images of skin lesions and brain MR images have shown that a thresholding step smaller than 0.1 does not change the segmentation result noticeably since the threshold value selected is the most stable one, and a small variation in it will not change the region size drastically. Segmenting images of Figs. 1a and 1b with this iterative thresholding method and using a seed point inside the tumor and the lesion, we obtain the results shown in Figs. 2a and 2c, respectively.

Although the segmentation results appear reasonably well, a human expert would segment the images as shown in Figs. 2b and 2d. We would like to provide the capability that can revise a result obtained by an automatic method. One may completely discard results of an automatic method and manually produce the results. However, in many situations only a small portion of a result from an automated segmentation is inaccurate, and we do not want to discard the good results. Automatic methods are valuable because they eliminate user bias. However, they may involve errors that would not be acceptable for data analysis. In situations where an automatic method fails, or when portions of a result are inaccurate, we would like to have the ability to locally revise the results. Next we will show that if region boundaries in an image are represented by RaG curves, the boundaries can be revised quickly and easily.

3. RAG CURVES

Given a set of points $\{\mathbf{V}_i : i = 1, \ldots, n\}$ along a contour, the rational Gaussian curve that approximates the points is given by

$$\mathbf{P}(u) = \sum_{i=1}^{n} \mathbf{V}_i g_i(u), \qquad u \in [0, 1], \tag{1}$$

where $g_i(u)$ is the ith basis function of the curve defined by

$$g_i(u) = \frac{G_i(u)}{\Sigma_{j=1}^{n} G_j(u)}, \tag{2}$$

and $G_i(u)$ is a 1-D Gaussian:

$$G_i(u) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(u-u_i)^2/2\sigma^2} \tag{3}$$

for an open curve and

$$G_i(u) = \sum_{k} \frac{1}{\sqrt{2\pi}\sigma} e^{-(u-u_i+k)^2/2\sigma^2} \tag{4}$$

for a closed curve. k is a small range, such as -1 to 1. Since a Gaussian extends from $-\infty$ to $+\infty$, when a curve is closed, the Gaussian makes an infinite cycle around it. In practice, however, because a Gaussian approaches zero exponentially, after a few cycles the effect of the Gaussian will not be measurable. Therefore, it is only sufficient to change k in a small range such as -1 to 1. Then, at the point where the curve closes, points near the start of the curve will affect the shape of the curve near its end, and points near the end of the curve will affect the shape of the curve near its beginning. This will enable the curve to close smoothly.

u_i is the ith node of the curve and is estimated from

$$u_i = \frac{|\mathbf{V}_{i+1} - \mathbf{V}_i|}{\sum_{j=1}^{n} |\mathbf{V}_{j+1} - \mathbf{V}_j|}, \qquad i \in [1, n], \tag{5}$$

where $\mathbf{V}_{n+1} = \mathbf{V}_1$ and $|\mathbf{V}_{i+1} - \mathbf{V}_i|$ is the Euclidean distance between points $\mathbf{V}_{i+1}$ and $\mathbf{V}_i$. σ is the standard deviation of Gaussians in a curve and shows the smoothness of the curve. The larger the standard deviation, the smoother the curve.

If the given points represent pixels on a region boundary, obtained as a result of intensity thresholding, the curve will approximate the boundary. By reducing the standard deviation of Gaussians, the curve will pass closer to the points and will recover details in the contour. Sometimes the details are due to noise, and to remove them, a larger standard deviation of Gaussian is needed.

A RaG curve is revised by moving its control points. Moving the positions of some of the control points will change the shape of the curve locally near the moved control points. If the control points of a curve represent pixels in a region boundary, revising the boundary requires moving many boundary pixels, which is not very convenient. We would like to approximate a region boundary by a curve that has far fewer control points than the number of pixels in the contour so that by moving a small number of control points, the shape of the curve can be revised quickly. In the following, we will show how an optimization process can substantially reduce the number of control points in a RaG curve without significantly changing its shape.

4. MINIMIZING THE NUMBER OF CONTROL POINTS IN A RAG CURVE

Suppose pixels in a contour are represented by $\{\mathbf{P}_j : j = 1, \ldots, N\}$. We would like to determine a RaG curve with control points $\{\mathbf{V}_i : i = 1, \ldots, n\}$ that will approximate the contour with a prescribed accuracy. Let's assume $\mathbf{P}(u) = [x(u), y(u)]$, $\mathbf{P}_j = (X_j, Y_j)$, and $\mathbf{V}_i = (x_i, y_i)$. Then, the sum of squared distances between the curve and the given pixels horizontally and vertically can be written as:

$$E_{2x} = \sum_{j=1}^{N} [x(u_j) - X_j]^2 \tag{6}$$

$$E_{2y} = \sum_{j=1}^{N} [y(u_j) - Y_j]^2, \tag{7}$$

where $u_j = \frac{j-1}{N-1}$ for an open curve and $u_j = \frac{j-1}{N}$ for a closed curve.

Since E_{2x} and E_{2y} are independent, we will determine the x and y components of the control points independently. To do so, we will rewrite (6) in terms of the x-component of the control points:

$$E_{2x} = \sum_{j=1}^{N} \Big[\sum_{i=1}^{n} x_i g_i(u_j) - X_j \Big]^2 \tag{8}$$

and find $\{x_i : 1, \ldots, n\}$ such that E_{2x} is minimized. To achieve this, we find the partial derivatives of E_{2x} with respect to the x_i's, set them equal to zero, and solve the obtained system of linear equations. That is, we find:

$$\frac{\partial E_{2x}}{\partial x_i} = \sum_{j=1}^{N} g_i(u_j) \Big[\sum_{i=1}^{n} x_i g_i(u_j) - X_j \Big] = 0, \qquad i = 1, \ldots, n. \tag{9}$$

Note that (9) is a linear equation of x_i's, which always has a solution because the obtained matrix of coefficients has diagonally dominant elements due to the nature of the RaG basis functions. By solving the system of equations defined by (9), we can determine the x-component control points. The y-component control points can be determined similarly.

In the above procedure, the standard deviation of Gaussians and the number of control points n were preassigned. The process determined n control points in such a way that the sum of squared errors between the obtained curve and the given contour was minimum. This minimum error, however, may be larger than the required accuracy. Suppose required accuracy is given in terms of the root-mean-squared error. That is, suppose

$$E = \sqrt{\frac{1}{N}(E_{2x} + E_{2y})}. \tag{10}$$

We will determine E after the optimization process, and if the obtained error is below the required value, we will stop. Otherwise, we will increase n and repeat the process until obtained error becomes less than required error. For an open curve, we can start with $n = 2$, and for a closed curve we can start with $n = 3$. Since the number of pixels in a contour could be in the order of hundreds and even thousands, for the process to converge faster, we can start with a rather large n and increase or decrease it if the obtained error is respectively larger or smaller than the required accuracy. Experimental results show that $n = N/(7E)$ is an appropriate starting point, where E is the required root-mean-squared error. For instance, if the number of pixels on a given contour is 500 and required root-mean-squared error between the contour and the approximating curve is 2 pixels, we may start with n =50 control points and increase or decrease n depending on whether the obtained error is greater than or less than the required value.

Figure 2c shows a contour consisting of 711 pixels, and Figs. 3a, 3b, and 3c show RaG curves approximating the contour with root-mean-squared errors equal to 2, 3, and 4 pixels, respectively. The number of control points representing these curves are 44, 30, and 22, respectively. Standard deviation of Gaussians were set to 0.02 pixels in all three cases. Increasing the standard deviation to 0.03 pixels we obtain the results depicted in Figs. 3d-f. The original contour is shown in white, while the approximating curves are shown in black. As the curve-fitting error is allowed to increase, we should increase the smoothness of the curve. As the curve-fitting error is decreased, we should decrease the standard deviation of Gaussians to let the curve bend sharply where the contour has sharp corners. The standard deviation of Gaussians should be proportional to the required accuracy. If required accuracy is 2 pixels, the standard deviation of the RaG curve approximating the curve should be about $1/n$, where n is the number of control points in the curve. Division by n is required because σ and u should be in the same scale, and we know that u varies from 0 to 1.

5. INTERACTIVE REVISION OF A 2-D SEGMENTATION RESULT

The result of our iterative thresholding method on a skin cancer image is shown in Fig. 4a. The obtained region contour is overlaid with the original image to demonstrate the quality of the segmentation. Although for the most part the segmentation is correct, there are parts of the contour that need refinement. If we approximate the contour of Fig. 4a with a RaG curve of standard deviation 0.02 and root-mean-squared error of 2 pixels, we obtain the control points shown in Fig. 4b. The small squares show the control points of the curve. The original contour contains 711 pixels, and the approximating curve has 44 control points. We can now revise the curve and the segmentation result by moving the control points of the curve. Figure 4c shows the result of this interactive revision.

Results of the automatic thresholding technique on a few other images are shown in Fig. 5. Although for the most part the segmentation results are satisfactory, some corrections are necessary to remove the local errors. Fig. 6 shows results after interactively revising the results. The curves were obtained using root-mean-squared error of 2 pixels and RaG curves of standard deviation 0.02 when fitting the contours obtained by the automatic method.

6. 3-D ITERATIVE THRESHOLDING

If an object of interest has rather homogeneous intensities that are different from intensities in its surroundings, we can isolate the object by a thresholding process. If the object of interest is not homogeneous or it has similar intensities compared to those in its surrounding, errors will be obtained by the thresholding process. Consider a lesion in a skin image or a tumor in an MR brain image. If the object of interest is known to be brighter than its background, as shown in Fig. 7a, we can start with a very high threshold value and gradually decrease the threshold value and determine change in the region size. The threshold value where change in region size as a function of change in threshold value is minimum can be automatically determined and used as the threshold value for segmenting the image. This threshold value shows the intensity at which region size is most stable under variations of the threshold value.

With this thresholding scheme, we can detect different regions using different threshold values automatically. Since objects in an image may have different intensities, they may require different threshold values to detect them. If a region has rather homogenous intensities, and we use an intensity from its boundary as the threshold value, we will obtain a region whose size will remain stable under variations of the threshold value. Our iterative thresholding method, therefore, automatically determines the best intensity in an object boundary for image segmentation–best in the sense that the region size will be most stable under variations of the threshold value. A threshold value selected in this manner ensures that if a small error is made in selecting the threshold value, the size of the detected region will not change drastically.

An algorithm similar to that described for segmenting 2-D images will be used to segment 3-D images. The only difference between our 2-D and 3-D methods is in the use of intensities of a sequence of image slices in 3-D rather than intensities of a single slice in 2-D to determine the threshold values. Thresholding is a very fast operation where each pixel requires only one comparison. Moreover, since segmented images are binary, they require very little memory for storage. Therefore, an image can be thresholded at multitudes of thresholds and saved to enable fast tracking of a region in the thresholded images. Tracking involves simply determining which regions in one threshold value are contained in which region in the subsequent threshold value. A simple neighborhood test can determine this. The result of iterative thresholding on the image of Fig. 7a is shown in Fig. 7b. This segmented image is overlaid the original image to show the quality of the segmentation. Three orthogonal cross-sections of the tumor are shown in these figures, and the 3-D region obtained by iterative thresholding is shown by 3-D dots in Fig. 7b. Some errors are visible in this segmentation. We will correct the errors interactively later.

To effectively refine an automatically obtained 3-D region, we first represent the region with a parametric surface. Then we optimize the number of control points in the surface and finally, by moving the control points, refine the segmentation result.

7. RAG SURFACES

Given a set of 3-D points $\{\mathbf{V}_i : i = 1, \ldots, n\}$, the rational Gaussian surface that approximates the points is given by[3]

$$\mathbf{P}(u, v) = \sum_{i=1}^{n} \mathbf{V}_i g_i(u, v), \qquad u, v \,\epsilon\, [0, 1], \tag{11}$$

where $g_i(u, v)$ is the ith basis function of the surface defined by

$$g_i(u, v) = \frac{G_i(u, v)}{\sum_{j=1}^{n} G_j(u, v)}, \tag{12}$$

and $G_i(u, v)$ is a 2-D Gaussian centered at (u_i, v_i):

$$G_i(u, v) = \frac{1}{2\pi\sigma^2} \exp\{[(u - u_i)^2 + (v - v_i)^2]/2\sigma^2\} \tag{13}$$

for an open surface and

$$G_i(u, v) = \sum_k \frac{1}{2\pi\sigma^2} \exp\{[(u - u_i)^2 + (v - v_i + k)^2]/2\sigma^2\} \tag{14}$$

for a cylindrical surface.

The openings in a cylindrical surface can be made extremely small so that after quantization, a closed surface is obtained. This surface representation can, therefore, be used to describe tumors or their parts. In a cylindrical surface, a 2-D Gaussian wraps around the surface infinitely. However, since a Gaussian approaches zero exponentially, its effect vanishes after a few cycles. When a surface is closed along parameter v, equation (14) should be used for $G_i(u, v)$ in equation (12).

A node is associated with each control point, showing the relation of that control point with the control points adjacent to it. If the control points represent the voxels on a 3-D region, and the region represents a cylindrical object, one of the parameters of the surface (u) should be selected to show positions along the axis of the cylinder, and the other parameter (v) should be selected to show positions of voxels in a cross-section of the cylinder with a plane normal to the axis of the cylinder. For instance, if the axis of the cylinder is in the axial direction and along the Z axis in an MR brain image, the angle of the line connecting the voxel in a XY slice to the center of the region with the X axis can be considered parameter v. Parameters u and v should be normalized to have values between 0 and 1.

It is possible to have an object that cannot be regarded a generalized cylinder in one direction but can be regarded a generalized cylinder in another direction. In such a situation, the proper direction should be selected to formulate the surface. An automatic process can determine the direction along which a generalized cylinder can be defined for a 3-D region. Suppose an automatic segmentation method is available. If all cross-sections of a region when viewed axially contain only one region, then the region can be modeled as a generalized cylinder axially. If the test fails axially, we may carry out the test sagittally or coronally to determine if the test succeeds and then define the surface.

It is possible for a surface to have branches and so not be describable by a generalized cylinder. In such a situation, the branches of the object can be initially cut (ignored) when parameterizing the surface. After the approximated surface is obtained, the surface can be interactively revised to have the desired branches.

The standard deviation of Gaussians determines the smoothness of a generated surface. A surface with a smaller standard deviation represents local details better than a surface with a larger standard deviation. The larger the standard deviation, the smoother the reconstructed surface.

Since a RaG surface is defined by its control points, to revise it, its control points should be moved. Since the control points of an approximating surface are originally the voxels given on a region (and there could be thousands of voxels), modifying a surface by moving the voxels is not practical. In the following, from the given voxels a surface is obtained that approximates the voxels with a prescribed accuracy. The approximating surface will have far fewer control points than the voxels, thereby enabling effective modification of the segmentation result.

8. MINIMIZING THE NUMBER OF CONTROL POINTS IN A RAG SURFACE

Suppose a region surface obtained in a volumetric image as a result of our iterative thresholding method contains N voxels: $\{\mathbf{P}_j : j = 1, \ldots, N\}$. Suppose the nodes of a generalized cylinder approximating the voxels are $\{(u_j, v_j) : j = 1, \ldots, N\}$, respectively. The nodes determine the adjacency relation between the voxels. We would like to determine a RaG surface with control points $\{\mathbf{V}_i : i = 1, \ldots, n\}$ that approximate the voxels with a prescribed root-mean-squared error. Let's assume $\mathbf{P}_j = (X_j, Y_j, Z_j)$, $\mathbf{P}(u, v) = [x(u, v), y(u, v), z(u, v)]$, and $\mathbf{V}_i = (x_i, y_i, z_i)$. Then we can determine the root-mean-squared error between the voxels and the approximating surface from

$$
\begin{aligned}
E &= \frac{1}{N}\sum_{j=1}^{N}\{[x(u_j, v_j) - X_j]^2 + [y(u_j, v_j) - Y_j]^2 + [z(u_j, v_j) - Z_j]^2\} && (15)\\
&= \frac{1}{N}\sum_{j=1}^{N}[x(u_j, v_j) - X_j]^2 + \frac{1}{N}\sum_{j=1}^{N}[y(u_j, v_j) - Y_j]^2 + \frac{1}{N}\sum_{j=1}^{N}[z(u_j, v_j) - Z_j]^2 && (16)\\
&= E_x + E_y + E_z. && (17)
\end{aligned}
$$

Since the three components of the surface are defined independently, to minimize the root-mean-squared error, we will minimize each component of the error separately. Now,

$$
E_x = \frac{1}{N}\sum_{j=1}^{N}\Big[\sum_{i=1}^{n} x_i g_i(u_j, v_j) - X_j\Big]^2. \tag{18}
$$

To determine the x-component control points $\{x_i : i = 1, \ldots, n\}$ that minimize E_x, we find the partial derivatives of E_x with respect to the x_i's, set them equal to zero, and solve the obtained system of linear equations. In the same manner, we can determine the y and the z components of the control points.

Note that the above process finds n control points in a RaG surface that approximate the N image voxels with minimum root-mean-squared error. The process assumes that n is given. n, however, is not known and has to be determined. Usually n is much smaller than N. We can start with a small n and gradually increase it until desired accuracy is reached. This, however, may take an unnecessarily long time. In our algorithm we start with $n = 2\sqrt{N}$ and gradually increase or decrease it until required accuracy is reached.

The standard deviation of Gaussians is another parameter that the user should provide. This parameter will depend on the number of control points used to describe the surface. As the number of control points in a RaG surface is increased to reduce the root-mean-squared error, the standard deviation of Gaussians should also be reduced to reproduce more details in the approximating surface. The standard deviation of Gaussians should be made proportional to the average distance between adjacent nodes. The denser the nodes, the smaller the distance between them, and a smaller standard deviation should be used. Experimental results show that standard deviation of Gaussians from average distance between nodes to five times that amount would be appropriate.

Figure 8b shows the voxels on a region obtained by our iterative thresholding method. The voxels are shown with dots in this figure. The generalized cylinders approximating this region with root-mean-squared error of 2, 3, and 4 are shown in Figs. 8a, 8b, and 8c, respectively. The standard deviation of Gaussians used in these figures is

0.02. Increasing the standard deviation to 0.03, we obtain the surfaces shown in Figs. 8d, 8e, and 8f for root-mean-squared-errors of 2, 3, and 4, respectively. The number of control points representing figures 8a-c or figures 8d-f are 169, 100 and 81, respectively. In the following section, the interactive revision process is outlined.

9. INTERACTIVE REVISION OF A 3-D SEGMENTATION RESULT

To revise a segmentation result represented by a RaG surface, the surface is first overlaid with the original image, as depicted in Fig. 9a. The user, by going through different image slices along axial, sagittal, and coronal axes, can then observe the intersection of the surface with each slice. The user can select a control point and move it in 3-D in a desired direction and by a desired amount revise a local segmentation result.

Figure 9b shows an intermediate result during the revision of surface of Fig. 9a, and Fig. 9c shows the final segmentation. In this particular example, the user had to spend 5 minutes revising the initial segmentation to the desired segmentation. Another example of the interactive refinement process is shown in Fig. 10.

10. DISCUSSION

The proposed interactive tools can be used with any automatic segmentation method as long as the output of the method is in the form of a region contour or a surface. An automatic thresholding method was described that determines different threshold values for different regions in an image. If the region of interest is known, interactively selecting a point from inside the region as a seed point will produce the threshold value for that region. This process works when the region of interest is rather homogeneous. The less homogeneous a region, the more the obtained error in the segmentation.

When an optimal threshold value is used to extract a region, it is possible that some noisy regions (holes) appear within the region due to nonhomogeneous intensities in the region. Such noisy regions are removed, and only the boundary contour or surface of the region of interest is used in the curve- or surface-fitting process. Segmenting a 2-D image of size 256×256 and extracting a region of interest takes about 30 seconds on an SGI Octane workstation. Total time to segment a 3-D image of size $128 \times 128 \times 128$ and extract a region of interest takes about 3 minutes.

The process of curve fitting can never fail if a closed boundary contour is given to start with. The surface-fitting method may fail if the provided region has branches. In such a situation one may find the center of the region and remove all voxels invisible to the center and then parametrize the remaining voxels by spherical or cylindrical parametrization.[3] After a surface is fitted to the remained voxels, the surface can be interactively pulled in any direction to reconstruct the desired branches. An alternative method would be to represent each of the branches by a RaG surface and combine them into a composite surface.[4]

The described optimization process determines the minimum number of control points in a RaG curve, fitting a region boundary, and in a RaG surface, fitting a region surface. The effectiveness of the optimization depends on the detailedness of the contour or surface, and on the selected standard deviation of Gaussians used in the curve- or surface-fitting process. Currently the standard deviation of Gaussians is selected by the user. Future work should consider computing the optimal standard deviation in addition to the minimum number of control points in a curve or surface. The optimization process for a region contour of length 711 pixels takes about 40 seconds, and about 15 minutes for a region surface containing 3300 voxels. The optimization step is the most computationally intensive step in the described image segmentation and revision method. Future work should focus on ways to speed up the optimization process.

The proposed interactive tools can revise a 2-D or a 3-D region to any desired shape by simply moving the control points of the curve or surface that represent it. Since the result of the automatic segmentation could sometimes be very different from the desired result, some additional tools to revise the contours before curve or surface fitting will further enhance the effectiveness of the method. For instance, tools to separate one region into two or more regions, or to combine two or more regions into one by quickly drawing or erasing image pixels/voxels, will improve the effectiveness of the process.

The results obtained by the proposed method are always satisfactory because otherwise the user can revise the results to the desired ones. Further experiments should involve replacing the automatic thresholding method described in this paper with other automatic segmentation methods. The described thresholding method can detect tumors that have rather homogeneous intensities and are surrounded by tissues that have considerably different intensities. For more complex objects, more sophisticated automatic segmentation methods should be used.

11. CONCLUSIONS

Automatic image segmentation methods often involve errors. The idea behind this work has been to develop tools that will enable a user to revise errors obtained by automatic methods. An interactive segmentation method was described that first uses an automatic method to segment an image. It then allows the user to interactively revise the results. To enable effective revision of a region, the segmentation result is first represented by a curve or a surface. Then the curve or surface is revised in the same manner that free-form industrial parts are revised.

RaG curves and surfaces were used as tools to carry out the revisions. RaG curves and surfaces can represent very complex geometries, and because the user has the ability to control the smoothness of a generated curve or surface, regions at desired levels of detail can be produced. An optimization process to determine RaG curves and surfaces with a minimum number of control points approximating a region boundary or surface with a prescribed accuracy was also described. The control points of an optimized curve or surface can then be locally moved as desired to revise a segmentation result. The revision process can continue until a satisfactory result is achieved.

The described process has been used to segment 2-D and 3-D medical images. Segmentation of a 2-D image of size 256×256 takes about 90 seconds, whereas segmentation of a 3-D image of size $128 \times 128 \times 128$ takes about 22 minutes. This time includes the iterative thresholding step, the curve- or surface-fitting step, the optimization step, and the interactive revision step on an SGI Octane computer. Only 10% of the total time was spent to interactively revise a result.

ACKNOWLEDGMENTS

Partial funding for this work was provided by the Wallace-Kettering Neuroscience Institute, Kettering Medical Center.

REFERENCES

1. W. Barrett and E. Mortensen, Fast, accurate, and reproducible live-wire boundary extraction, *Proc. Visualization in Biomedical Computing,* Hamburg, 1996.
2. J. E. Cabral Jr., K. S. White, Y. Kim, and E. L. Effmann, Interactive segmentation of brain tumors in MR images using 3-D region growing, *SPIE Vol. 1898 Image Processing,* 171-179, 1993.
3. A. Goshtasby, Design and recovery of 2-D and 3-D shapes using rational Gaussian curves and surfaces, *Int. J. Computer Vision,* (10)3:233-256, 1993.
4. A. Goshtasby, Geometric modeling using rational Gaussian curves and surfaces. *Computer-Aided Design,* 27(5):363–375, 1995.
5. K. P. Hinshaw and J. F. Brinkley, Shape-based interactive three-dimensional medical image segmentation, *Proc. SPIE 3034, Medical Imaging: Image Processing,* K. H. Hanson (ed.), 236-242, 1997.
6. K. H. Höhne and W. A. Hanson, Interactive 3D segmentation of MRI and CT volumes using morphological operations, *Journal of Computer Assisted Tomography,* 16(2):258-294, 1992.
7. A. Kundu, Local segmentation of biomedical images, *Computerized Medical Imaging and Graphics,* 14:173-183, 1990.
8. S. M. Pizer, T. J. Cullip, and R. E. Fredericksen, Toward interactive object definition in 3D scalar images. In: K. H. Höhne *et al.* eds., *3D Imaging in Medicine* Berlin: Springer, 83-105, 1990.
9. T. Schiemann, M. Bomans, U. Tiede, and K. H. Höhne, Interactive 3D segmentation, *Proc. SPIE 1808, Visualization in Biomedical Computing II,* R. A. Robb (ed.), 1992.
10. D. Welte, T. Grunert, U. Klose, D. Petersen, and E. Becker, Interactive 3D segmentation and visualization of vessels, *Computer Assisted Radiology*, 329-335, June 26-29, 1996.

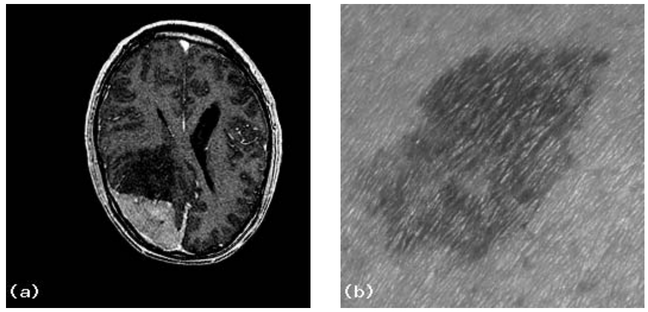

Figure 1. (a) An MR brain image and (b) a skin image.

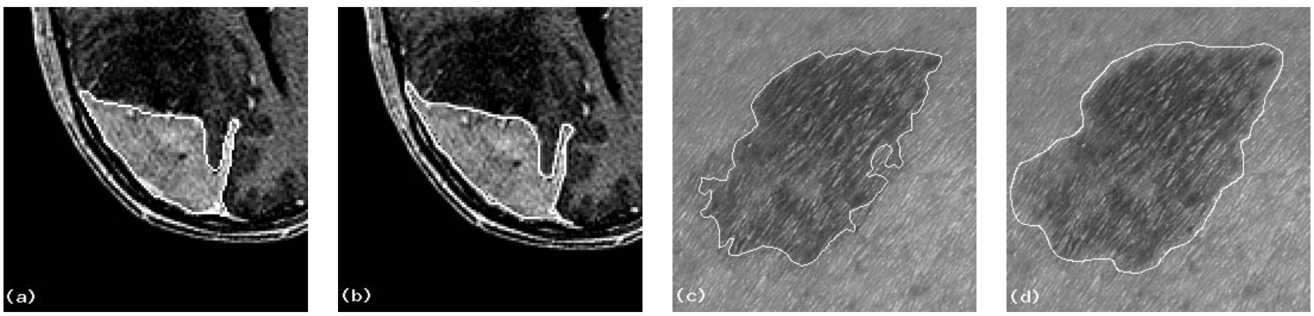

Figure 2. (a,c) Automatic and (b,d) manual segmentation results.

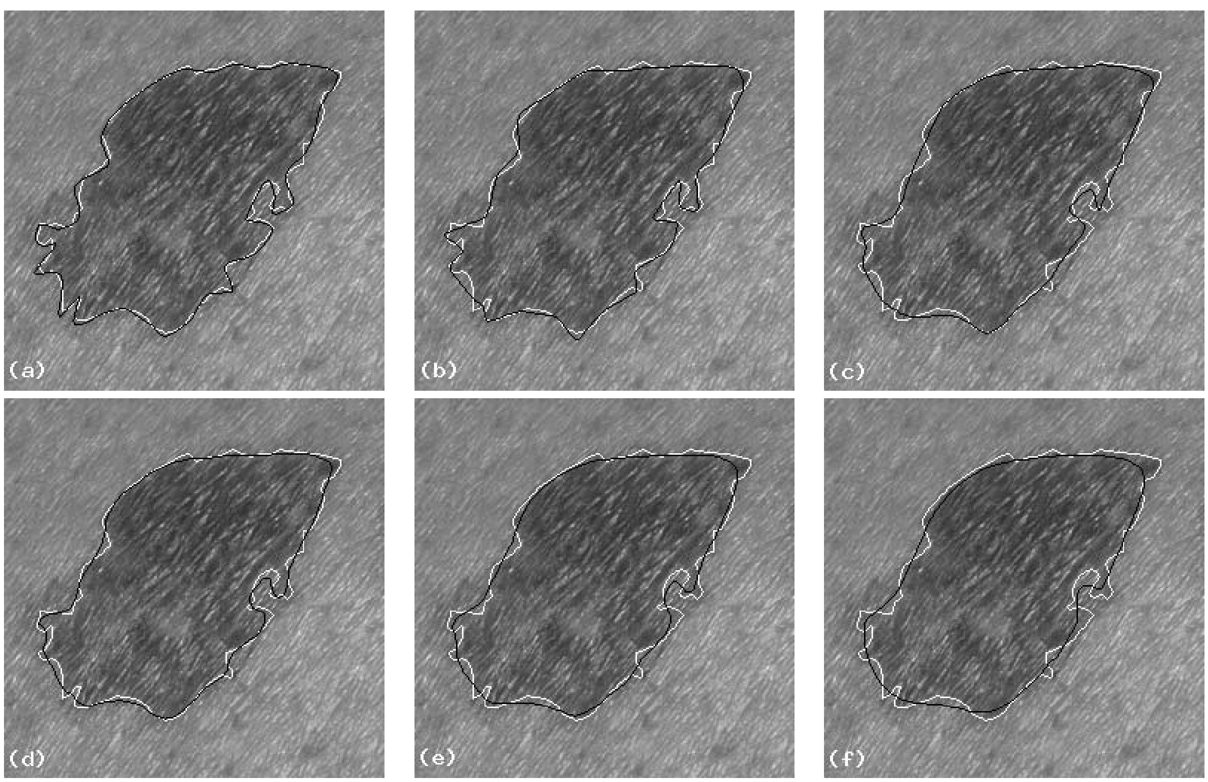

Figure 3. (a-c) Approximating curves with E=2,3,4, σ=0.02 and (d-f) E=2,3,4, σ=0.03.

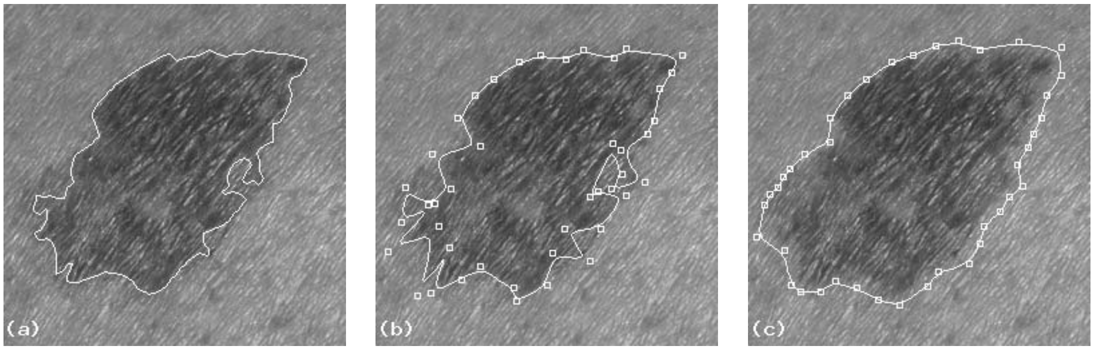

Figure 4. (a-c) Steps in revising a segmentation result.

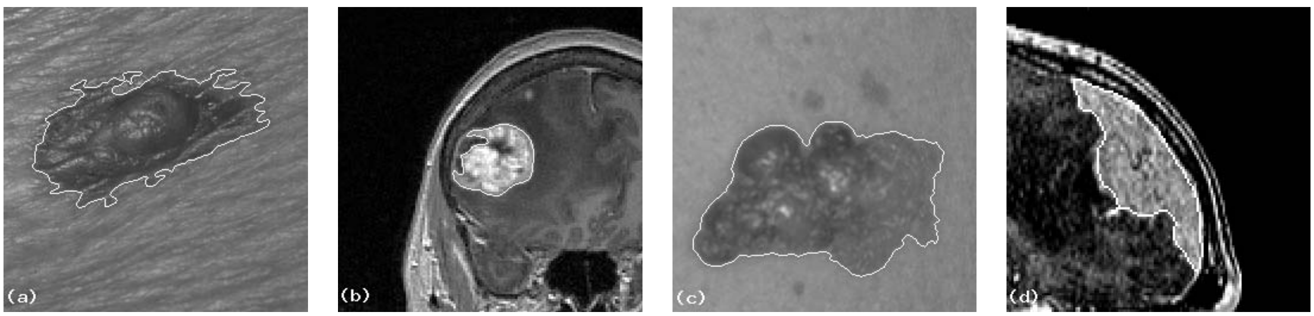

Figure 5. More initial segmentation results obtained from the iterative thresholding method.

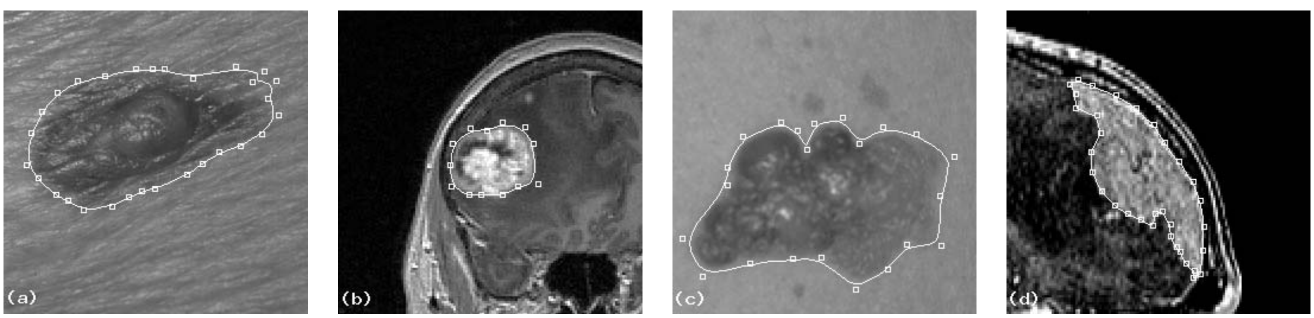

Figure 6. Final segmentation results after interactive revision.

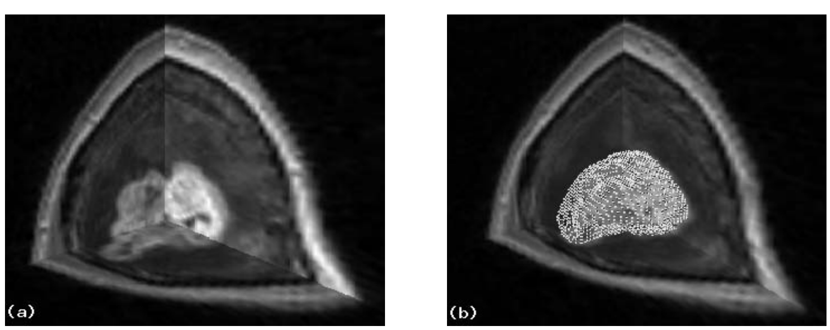

Figure 7. (a) A 3-D brain image and (b) result of iterative thresholding.

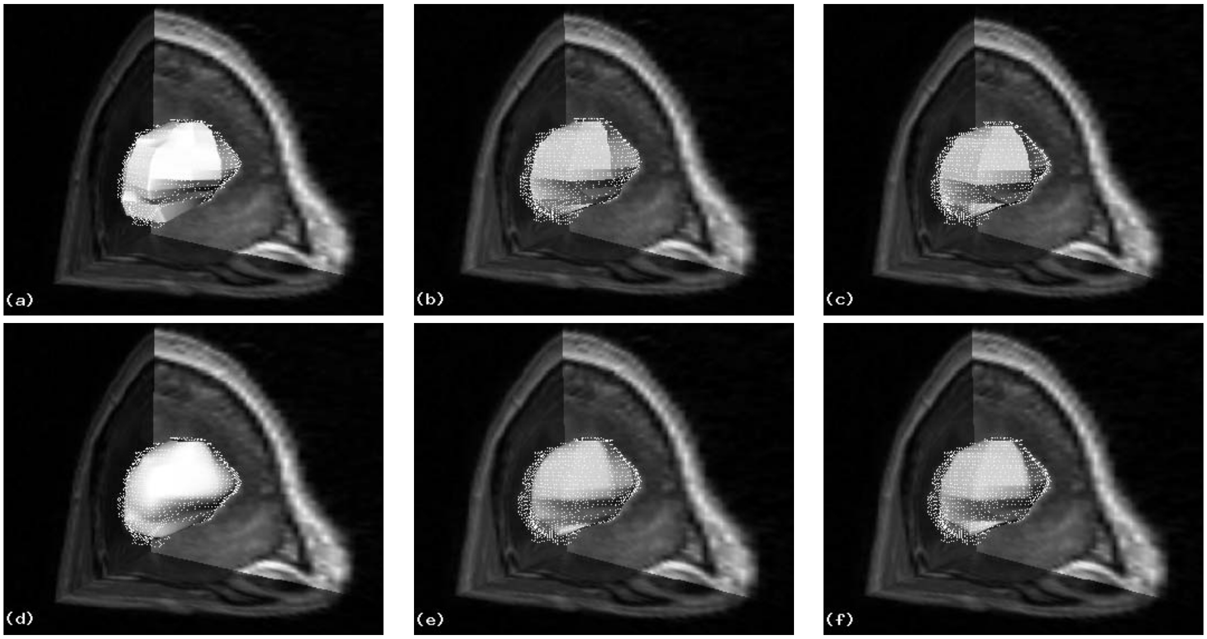

Figure 8. (a-c) Approximating surfaces with E=2,3,4, σ=0.02 and (d-f) E=2,3,4, σ=0.03.

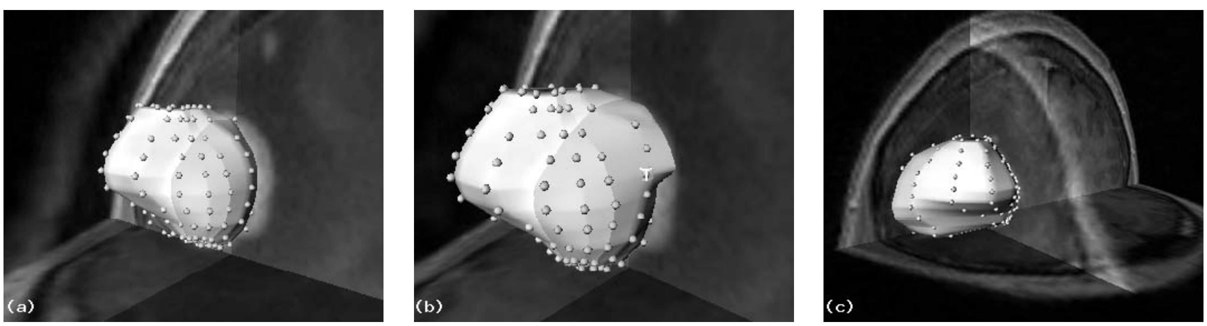

Figure 9. (a) Initial segmentation result. (b) Interactive revision. (c) Final result.

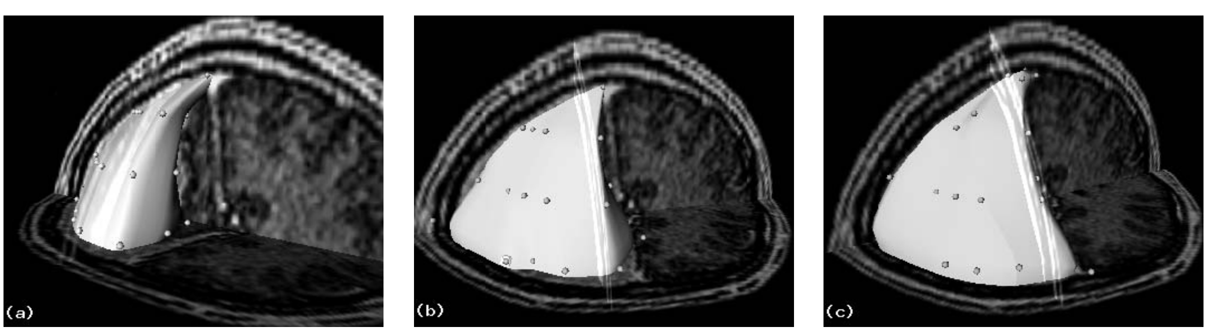

Figure 10. (a) Initial segmentation result. (b) Interactive revision. (c) Final result.