



A Computer-Aided Design System for Refinement of Segmentation Errors

Marcel Jackowski¹ and Ardeshir Goshtasby²

¹ Yale School of Medicine, Diagnostic Radiology Dept., New Haven, CT 06520

² Computer Science & Eng. Dept., Wright State University, Dayton, OH 45435

Abstract. Automatic image segmentation methods often involve errors, requiring the assistance of the user to correct them. In this paper, a computer-aided design system is introduced for correcting such errors. The proposed system approximates each 3-D region by a parametric surface. Region voxels are first parametrized spherically using a coarse-to-fine subdivision method. By using the voxel positions and their parameter coordinates, control points of a rational Gaussian surface are determined through a least-squares method to approximate the region. Finally, this surface is overlaid with the volumetric image and by locally pulling or pushing it with the mouse while viewing image information, the surface is revised as needed. Typically, a few minutes are sufficient to correct errors in a region.

1 Introduction

Image segmentation is the process of partitioning an image into meaningful parts. The objective is to extract objects or their parts in an image. Difficulties arise when noise is present in the image or when properties within objects vary. The problem is worsened when their boundaries become blurred or when boundary information is absent, owing to different tissues having similar properties. These variations, which are often unpredictable, make it impossible to develop an automatic method that can segment all images correctly. At the present, the best one can hope for is to have a method that can correctly delineate most parts of an object of interest, and in areas where it makes a mistake, allow the user to interactively correct the errors.

Various user-guided and interactive segmentation methods have been developed. A popular method described by Falcão, Mortensen, Udupa, and others [6, 1, 10, 11] is known as “live-wire.” In this method, the user selects a number of points on the region boundary, and the program automatically finds optimal boundary segments between consecutive points by minimizing a cost function based on the image gradient information. Minimum-cost segment paths are calculated in real time, allowing the user to interactively snap each segment to the intended boundary to constrain the search space for the optimum solution and improve speed. An extension of it known as “live-lane” was later devised [5] where the optimization process is restricted to a small area around the image

edges, called a “lane.” More recently, Falcão and Bergo introduced a full volumetric interactive segmentation approach based on Image Foresting Transforms (IFT) [4]. The IFT is Dijkstra’s shortest-path algorithm modified to handle multiple sources using more general cost functions. Seed points are added or removed interactively as needed and cost graphs are pruned in real time, thereby allowing revision of the segmentation result.

An interactive segmentation method based on a genetic algorithm was described by Cagnoni *et al.* [2]. In this method, the boundary contour of a region of interest is manually drawn in one of the slices. The boundary contour is then considered an initial contour in the subsequent slice and the contour is refined by a genetic algorithm using image information.

Energy-minimizing models or “snakes” have also been used for segmentation as well as revision of regions [9]. Since some points in a snake may trap in local minima during the optimization process, the globally optimal solution can be missed. To avoid this, often the user is required to intervene and either move some of the snake’s points that are thought to have converged to local minima or guide the snake to the optimal position by interactively controlling the external forces. A survey of interactive segmentation methods providing different levels of user control is given by Olabarriaga and Smeulders [12].

The new idea introduced in this paper is to use the capabilities of a computer-aided design system to quickly and easily refine the result of a 3-D segmentation, just like editing a 3-D geometric model. We assume that a volumetric image has been segmented and regions of interest have been extracted. A region is composed of connected voxels that represent the bounding surface of an object, which we call a *digital shape*. Under these assumptions, we first describe the parametric surface that will be used to represent a given digital shape. Next, we show how to parametrize the shape voxels and optimally determine the control points of the approximating surface. Then we describe the segmentation refinement process, involving interactive free-form surface deformations. Finally, we present experimental results by the proposed method.

2 Approach

2.1 Surface Approximation

Since voxels in a digital shape usually do not form a regular grid, we use a rational Gaussian (RaG) surface [7] to represent the shape. The standard deviation of Gaussians in a RaG surface may be used to control the smoothness of the created surface.

Given a set of (control) points $\mathbf{V}_i = \{(x_i, y_i, z_i) : i = 1, \dots, n\}$ and their parameter coordinates $\{(u_i, v_i) : i = 1, \dots, n\}$, the RaG surface approximating the control points is defined by:

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

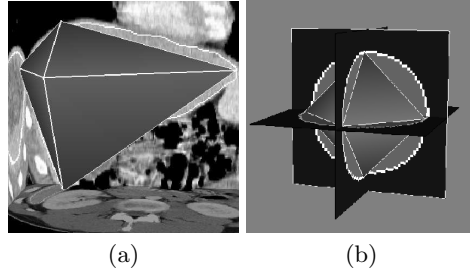


Fig. 1. (a) Approximation of a digital shape by an octahedron. (b) Approximation of a sphere by an octahedron. Parameter coordinates of octahedral vertices in the sphere are assigned to the octahedral vertices in the shape.

where $g_i(u, v)$ is the i th blending function of the surface given by

$$g_i(u, v) = \frac{G_i(u, v)}{\sum_{j=1}^n G_j(u, v)}. \quad (2)$$

$G_i(u, v)$ is the Gaussian centered at (u_i, v_i) in the parameter space. The parameter coordinates determine the adjacency relation between the points. The standard deviation of all Gaussians are set to the same value σ and controlled globally.

When the shape under consideration is closed and has a spherical topology, the parameter coordinates at the shape voxels can be determined by mapping the voxels to a sphere. Assuming that $\phi \in [-\pi/2, \pi/2]$ and $\theta \in [0, 2\pi]$ represent the spherical coordinates of voxels of a digital shape, we will use $u = (\phi + \frac{\pi}{2})/\pi$ and $v = \theta/2\pi$ in the equation of the RaG surface approximating the shape. In the following sections, we show how to spherically parametrize voxels in a closed digital shape and approximate the shape by a RaG surface using the least squares method.

2.2 Shape Parametrization

To approximate the digital shape with a RaG surface, one must first find the parameter coordinates of the shape voxels. Here we employ a coarse-to-fine subdivision method described in [8]. Initially, the given digital shape and a unit sphere are simultaneously approximated by two octahedra. The process involves placing a regular octahedron at the center of gravity of the shape and extending its axes until they intersect the shape (see Fig. 1a). By establishing the correspondence between the triangles in the shape approximation and triangles in the sphere approximation (Fig. 1b), parameters of the shape control points are determined. In the case of irregular shapes, the center of the octahedron is taken as the midpoint of the major axis segment inside the shape.

The projection of each triangular facet onto the shape determines a digital patch. The triangles obtained by the octahedral subdivision are further subdivi-

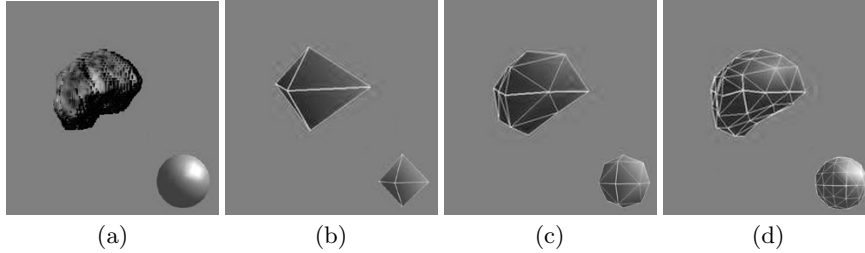


Fig. 2. (a) A digital shape and a unit sphere in the lower right corner. (b-d) Three simultaneous subdivisions of a digital shape and the reference sphere.

vided into smaller facets in a recursive manner [8]. This process continues until the distance between each triangular facet and its associating patch falls below a prescribed tolerance. Whenever a triangle in the shape approximation is subdivided, the corresponding triangle in the sphere approximation is subdivided also. Hence, there always exists a one-to-one correspondence between triangles in the shape approximation and triangles in the sphere approximation. By knowing the parameters of mesh vertices in the sphere, we will know the parameters of corresponding mesh vertices in the shape. The process is graphically shown in Fig. 2. By using the vertices of the triangular mesh as the control points of a RaG surface and the parameters at mesh vertices as its nodes, a smooth parametric surface can be obtained to approximate the shape. Note, however, that the surface obtained in this manner will only approximate the mesh vertices and will not necessarily pass through the vertices. In the next section, we show how to improve the shape recovery process by determining the control points of a RaG surface that interpolates the mesh vertices.

2.3 Least-Squares Computation of the Control Points

Suppose a digital shape is available and the shape voxels are parametrized according to the procedure outlined in the preceding section. Also, suppose the shape is composed of N voxels: $\{\mathbf{P}_j : j = 1, \dots, N\}$ with parameter coordinates $\{(u_j, v_j) : j = 1, \dots, N\}$. We would like to determine a RaG surface with control points $\{\mathbf{V}_i : i = 1, \dots, n\}$ that can approximate the shape points optimally in the least-squares sense. Let $\mathbf{P}_j = (X_j, Y_j, Z_j)$, $\mathbf{P}(u, v) = [x(u, v), y(u, v), z(u, v)]$, and $\mathbf{V}_i = (x_i, y_i, z_i)$. Then the sum of squared distances between the voxels and the approximating surface can be written as:

$$E^2 = \sum_{j=1}^N \{[x(u_j, v_j) - X_j]^2 + [y(u_j, v_j) - Y_j]^2 + [z(u_j, v_j) - Z_j]^2\} \quad (3)$$

$$= E_x^2 + E_y^2 + E_z^2. \quad (4)$$

Since the three components of the surface are independently defined, to minimize E^2 , we minimize E_x^2 , E_y^2 , and E_z^2 , separately. This involves determining

the partial derivatives of E^2 with respect to each variable, setting the partial derivatives to zero and solving the obtained system of equations. For E_x^2 , the following system of n linear equations are obtained:

$$E_x^2 = \sum_{j=1}^N g_k(u_j, v_j) \sum_{i=1}^n [x_i g_i(u_j, v_j) - X_j] = 0, \quad k = 1, \dots, n, \quad (5)$$

which can be solved for $x_i, i = 1, \dots, n$. Components y_i and z_i are determined similarly. The standard deviation of all Gaussians is set to the same value σ and controlled globally. Since RaG basis functions monotonically decrease from a center point, if σ is not very large, these equations will have diagonally dominant matrices of coefficients, ensuring a solution. Note that the above process positions the n control points of a RaG surface so that the surface will approximate the N image voxels with the least sum of squared errors. n depends on the size and complexity of the shape being approximated (n is typically a few hundred). The smallest surface-fitting error will be obtained when the standard deviation of Gaussians matches the level of detail of the shape. This minimum error can be determined by a steepest-descent algorithm. However, since the given region is known to contain errors, finding the surface that is very close to the region may not be of particular interest. Currently, after the control points of an approximating surface are determined, the user is allowed to interactively vary the smoothness (standard deviation) of the surface and view the obtained surface as well as the associating RMSE. In this manner, the standard deviation of Gaussians is interactively selected to reproduce a desired level of detail in the constructed shape.

2.4 Shape Editing

Once the result of an automatic segmentation is represented by a free-form parametric surface, the surface can be revised to a desired geometry by appropriately moving its control points. In the system we have developed, an obtained surface is overlaid with the original volumetric image. Then by going through different image slices along one of the three orthogonal directions, the user visually observes the intersection of the surface with the image slices and verifies the correctness of the segmentation. When an error is observed, one or more control points are appropriately moved to correct it. As the control points are moved, the user will observe changes in the surface in real-time.

Figure 3a shows the surface approximating a brain tumor within the original volumetric image. By picking the surface at a point near where the error occurred (shown by white arrow), a spherical attractor is activated and the control points (darker dots) falling inside the attractor are selected. By changing the radius of action, different numbers of control points are selected. Each point is translated in the appropriate direction by connecting it to the center of the attractor and by using the amount proportional to the cosine of the angle between that direction and the direction of motion of the mouse. Only those control points falling inside the hemisphere with positive cosines are moved. This avoids the movement of

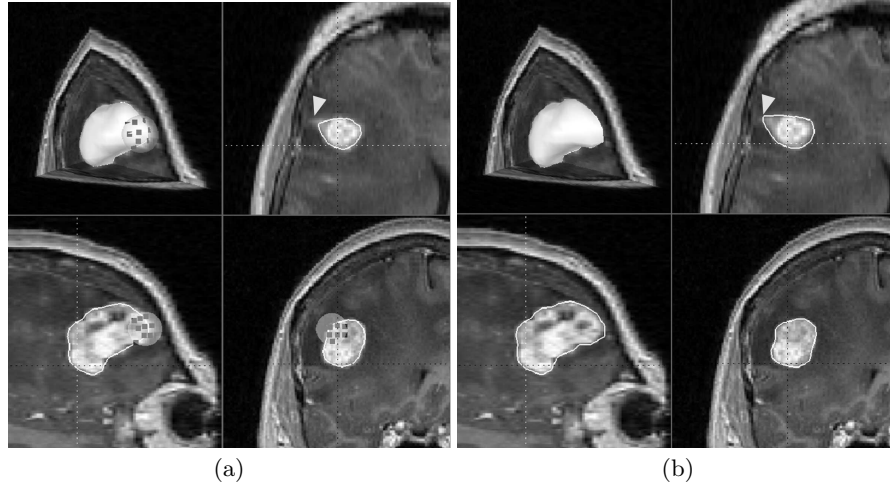


Fig. 3. (a) Overlaying of the approximated tumor surface and the volumetric image. Dots show the selected control points during surface editing. The upper-left window shows the 3-D view of the image volume with all three orthogonal slices. Arrow points to the revision area. (b) The tumor after necessary modifications. This is the final result in parametric form.

control points with negative cosines in the opposing direction. It also ensures that discontinuities will not occur between points that are moved and points that are not. Fig. 3b shows the resulting surface after revision. Surface revision can be performed gradually and repeatedly while observing the image information. The sensitivity of the surface to the motion of the mouse can be changed by increasing or decreasing the weights assigned to the control points.

3 Results

Examples of the proposed segmentation revision method are shown in Fig. 4. The first column shows the original images, the second column shows the initial segmentation results, and the third column shows the results after the necessary revisions. Both the ventricular blood pool (Fig. 4, first row) and the brain tumor (Fig. 4, second row) were initially segmented by applying a smoothing operation and an optimal intensity thresholding method. At the optimal threshold value, a small change in threshold value will change the segmentation result minimally. This threshold value corresponds to the intensity at object boundaries where intensities change sharply. Therefore, a slight error in estimation of the threshold value will not change the segmentation result drastically. The liver (Fig. 4, third row) was segmented by a 3-D Canny [3] edge detector. Weak edges were removed by interactively varying the gradient threshold value and observing the resulting edges. In these figures, results of the initial segmentation are shown after RaG

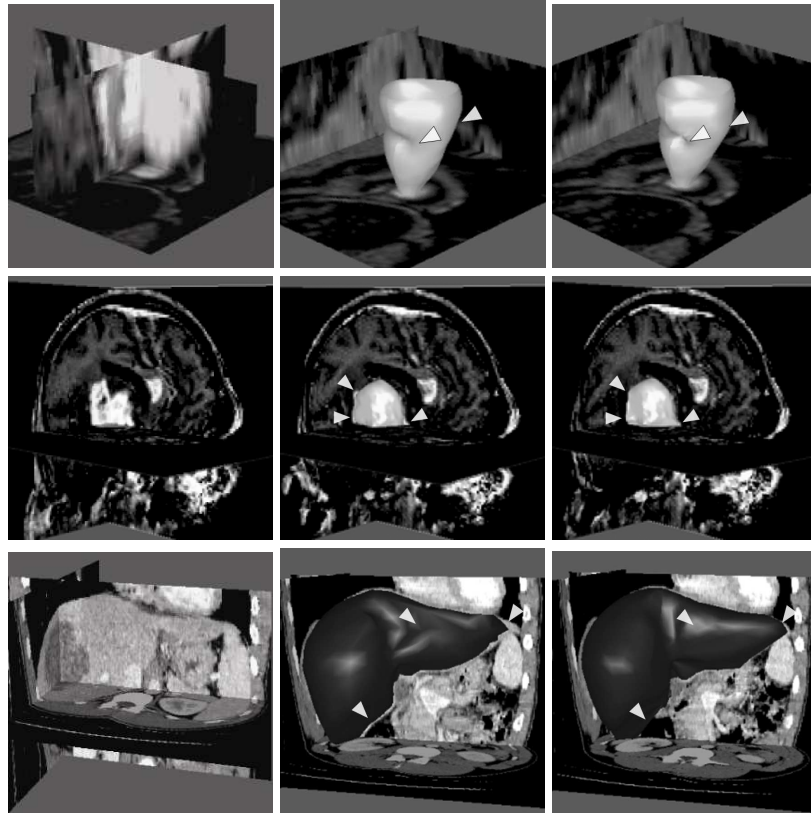


Fig. 4. First row: A short-axis cardiac MR image and segmentation of the left ventricular cavity. Second row: An MR brain image and segmentation of the tumor. Third row: An abdominal CT image and segmentation of the liver. The first column shows the original images, the middle column shows the initial segmentation results, and the right column shows the results after the necessary modifications (white arrows).

surface fitting by the least-squares method. RaG surfaces were then interactively revised as needed while viewing the overlaid surface and volumetric image. Final segmentation results are shown in the third column of Fig. 4.

Approximation of the initial regions by triangular meshes took from 10 to 30 seconds and approximation of the regions with RaG surfaces by the least-squares method took from 40 to 60 seconds. Interactive revision of the initial surfaces to obtain the final surfaces took from 1 to 2 minutes. All these times are measured on an SGI Octane computer with R10000 processor. Although the time to subdivide a region into a triangular mesh and the time to fit a RaG surface to a volumetric region are fixed for a given region, the time needed to revise an initial surface to a desired one depends on the speed of the user and the severity of errors in the initial segmentation. The final result of a segmentation obtained by the

proposed system is thus user-dependent. Since users have different experiences in image interpretation, results obtained by different users may differ.

4 Conclusions

In this paper, the idea of using a computer-aided design system to refine the result of an automatically determined segmentation was introduced. In the proposed system, a RaG surface is fitted to voxels covering a 3-D region by the least-squares method. The surface and the original volumetric image are then overlaid and the surface is interactively revised until the desired segmentation is achieved. Because a 3-D shape is represented by a parametric surface, the surface may be sent to a computer-aided manufacturing system for construction of an actual 3-D model of the shape. The proposed system provides tools with which a user can modify a segmentation result freely. There are no limitations in shape, size, or complexity of a region except that it should have a spherical topology.

References

1. W. Barrett and E. Mortensen. Fast, accurate, and reproducible live-wire boundary extraction. *Proc. Visualization in Biomedical Computing*, 1996.
2. S. Cagnoni, A. B. Dobrzeniecki, R. Ploi, and J. C. Yanch. Genetic algorithm-based interactive segmentation of 3-D medical images. *Image and Vision Computing*, 17:881–895, 1999.
3. J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
4. A. X. Falcao and F. P. G. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. Medical Imaging*, 23(9):1100–1108, 2004.
5. A. X. Falcao and J. K. Udupa. A 3-D generalization of user-steered live-wire segmentation. *Medical Image Analysis*, 4:389–402, 2000.
6. A. X. Falcao, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. A. Lotufo. User-steered image segmentation paradigms: live-wire and live-lane. *Graphical Models and Image Processing*, 60(4):233–260, 1998.
7. A. Goshtasby. Design and recovery of 2-D and 3-D shapes using rational Gaussian curves and surfaces. *Intl. Journal of Computer Vision*, 10(3):233–256, 1993.
8. M. Jackowski, M. Satter, and A. Goshtasby. Approximating digital 3-D shapes by rational Gaussian surfaces. *IEEE Trans. Visualization and Computer Graphics*, 9(1), 2003.
9. A. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Intl. Journal of Computer Vision*, 1:321–331, 1987.
10. E. N. Mortensen and W. A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384, 1998.
11. E. N. Mortensen, B. S. Morse, W. A. Barrett, and J. K. Udupa. Adaptive boundary detection using live-wire two-dimensional dynamic programming. *IEEE Proc. Computers in Cardiology*, pages 635–638, 1992.
12. S. D. Olabarriaga and A. W. M. Smeulders. Interaction in the segmentation of medical images: A survey. *Medical Image Analysis*, 5:127–142, 2001.