

# A Computer-Aided Design Environment for Segmentation of Volumetric Images

*Marcel Jackowski\**, *Martin Satter*<sup>†</sup>, and *Ardeshir Goshtasby\**<sup>‡</sup>

*Abstract*—Due to considerable variations in sensor and tissue characteristics, design of automated methods that can correctly segment all input images is very difficult, if not impossible. Clinicians often find it necessary to revise the result of an automatically determined segmentation. In this paper, a computer-aided design environment is introduced that enables quick and effective revision of segmented volumetric regions like revising 3-D free-form shapes. In the proposed environment, a region obtained by segmenting a volumetric image is approximated by a parametric surface. The user, by observing the overlaid surface and the original volumetric image, moves the control points of the surface to revise the segmentation. With the proposed environment, a user typically needs a few minutes to revise the result of a 3-D segmentation.

*Index Terms*—Image segmentation, computer-aided design, rational Gaussian surface, volumetric image.

## 1 Introduction

Image segmentation is the process of partitioning an image into meaningful regions. For the regions to be meaningful, they should represent objects or their parts. Difficulties arise when properties within objects vary or boundaries of objects are not well defined. The problem is made worse when sensor inaccuracies exist and noise is present in an image. These variations, which are often unpredictable, make it impossible to develop an automatic method that can correctly segment all input images.

Image segmentation is perhaps the most studied topic in image analysis. A large portion of papers published in image analysis journals and presented at conferences annually are devoted to image segmentation. The developed methods often take into consideration various properties of images or objects. When properties of images or objects deviate from those anticipated, errors occur. Even for a limited class of images, for instance MR brain images, various methods have been developed, none of which is guaranteed to correctly segment a

---

\*Department of Computer Science and Engineering, Wright State University, Dayton, OH.

†Wallace-Ketting Neuroscience Institute, Kettering Medical Center, Kettering, OH.

‡Corresponding author.

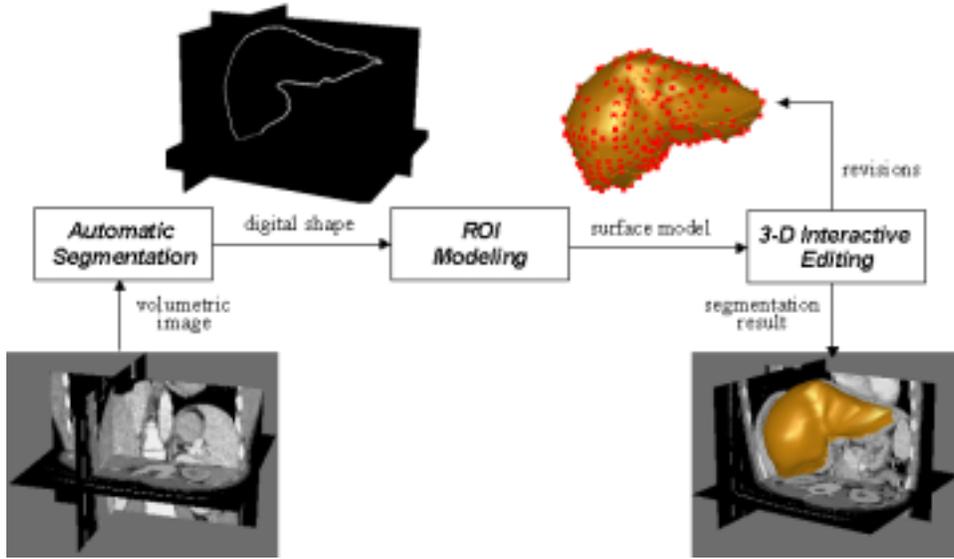


Figure 1: The proposed computer-aided design environment for region editing. The environment receives a region, which is the result of an automatically determined segmentation, as input. It then represents the region by a free-form parametric surface and overlays the surface with the volumetric image. The user then revises the surface while viewing both the volumetric image and the surface. The final result is generated in parametric or digital form.

new image. This may be because there are sensor variations; variations in the brain’s shape, size, and intensity distribution; and variations in intensities of tissues surrounding the brain. Since an error-proof segmentation cannot be guaranteed, user assistance is needed to revise the result of an automatically obtained segmentation. At present, the best possible method is one that can correctly segment most of an image and, in places where it makes a mistake, allow the user to correct the errors.

We have developed a computer-aided design environment that can assist the user revise the result of an automatically determined segmentation. We assume a region obtained by an automatic method is in voxels, representing the bounding surface of an object of interest. The developed environment will fit a parametric surface to the voxels and overlay the surface with the volumetric image. Then, by viewing both the image and the surface, the user will edit the surface until the desired shape is obtained. The concept behind the proposed environment is depicted in Fig. 1.

Attempts to develop user-guided and interactive segmentation methods have been made before. Barrett, Udupa, Mortensen, and others [1] [7] [17] [18] [20] developed a method known as “live-wire” with which a user roughly draws the boundary of a region of interest using a mouse. An automatic process then takes over and revises the boundary by optimizing

a cost function. An alternative method was introduced where the user selects a number of points on the region boundary and the program then automatically finds boundary segments between consecutive points, again by minimizing a cost function. These methods were later optimized for speed [8] and extended to 3-D [6] by creating boundary contours on a few strategically placed slices and by using the obtained boundary points to trace contours in other slices.

Cabral *et al.* [4] developed editing tools associated with a region-growing method that enables a user to add or remove image voxels in a region to revise the region. Hinshaw and Brinkley [12] developed a 3-D shape model that uses prior knowledge of an object's structure to guide the search for the object. Object structure is interactively specified with a graphical user interface.

Höhne and Hanson [13] developed low-level segmentation functions based on morphological operators to interactively delineate regions of interest. Pizer *et al.* [19] developed a method that segments a volumetric image into regions at a hierarchy of resolutions. Then, by pointing to an object in a cross-sectional image at a resolution, selects and revises a region. Welte *et al.* [21] developed an interactive method for separating vessels from each other and from the background in MR angiographic images. To reduce the complexity of the displayed structures in this method during the interactive segmentation, a capability to select substructures of interest was provided.

Energy-minimizing models or "snakes" are another set of tools that have been used to guide a segmentation and revise obtained results [15] [16]. With an energy-minimizing model, a contour or a wireframe is initialized approximately where an object of interest is believed to exist. An optimization process then iteratively revises the contour or the wireframe to minimize a cost function defining the energy of the snake. In this process, the user is allowed to intervene and revise the intermediate or final segmentation results by appropriately moving the snake points.

The new idea introduced in this paper is to use the capabilities of a computer-aided design system to quickly and effectively refine the result of a 3-D segmentation, a method identical to that used to edit a 3-D shape. By having a mental picture of an object of interest and viewing the information present in an image, the user interactively modifies the result of an automatically obtained segmentation. If parts of a region obtained by an automatic method are not satisfactory as judged by the user, those parts are interactively modified, like editing a 3-D free-form shape in a computer-aided design environment. This is achieved by representing the region with a parametric surface and overlaying it with the volumetric image. The surface is then modified by pulling or pushing it locally while viewing both the image and the surface.

We assume an automatic segmentation method that correctly finds most parts of a region of interest in a volumetric image is available. The capability introduced in this paper will enable the user to quickly and effectively revise parts of an object where errors are believed to exist.

## 2 A Computer-Aided Design Environment for Shape Design and Editing

The result of a typical 3-D segmentation is a set of voxels that forms the bounding surface of an object of interest. We will refer to this as a *volumetric region* or a *digital shape*. In this section, a method that fits a parametric surface to a digital shape is described. Since voxels belonging to a shape do not usually form a regular grid, formulations that require a regular grid of control points such as B-splines and Non-Uniform Rational B-Splines (NURBS) are not suitable. We choose the rational Gaussian (RaG) formulation [10] [11] that does not require a regular grid of control points to represent a free-form shape. We will show how to parametrize voxels in a digital shape and how to determine the control points of a RaG surface approximating a digital shape by the least-squares method. The obtained RaG surface is then overlaid with image data and the user is allowed to revise the surface by moving its control points.

### 2.1 RaG Surfaces

Given a set of 3-D points  $\{\mathbf{V}_i : i = 1, \dots, n\}$ , the RaG surface that approximates the points is given by [10]

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

where  $g_i(u, v)$  is defined by

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

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

$$G_i(u, v) = W_i \exp\{-[(u - u_i)^2 + (v - v_i)^2]/2\sigma^2\}. \quad (3)$$

Formulas (1)–(3) are for an open surface. If a surface is closed from one side, like a cylindrical surface,  $G_i(u, v)$  should be replaced by

$$G_i(u, v) = \sum_{k=-l}^l W_i \exp\{-[(u - u_i)^2 + (v - v_i + k)^2]/2\sigma^2\}. \quad (4)$$

The openings at the two ends of a cylindrical surface can be made extremely small so that after resampling to a digital shape a closed region is obtained. In a cylindrical surface, a 2-D Gaussian wraps around the closed side of the surface infinitely. However, since a Gaussian approaches zero exponentially, its effect vanishes after one or two cycles. Therefore,  $l$  in formula (4) is typically 1 or 2. In this formula, the surface closes along parameter  $v$ .

Parameter  $W_i$  is the weight of control point  $\mathbf{V}_i$ . Having the ability to assign weights to the control points enables some control points to influence the reconstructed surface more than others. This capability is quite useful in interactive design. By interactively increasing or decreasing the weights at the control points, the sensitivity of the control point movements can be controlled to effectively modify a shape. For interactive design, the weights at all control points except those being moved are set to one. The weights at the control points being moved are selected interactively to facilitate shape modification.

The standard deviation of Gaussians in formulas (3) and (4) 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 obtained surface.

A node is associated with each control point in a parametric surface, showing the relation of that control point with respect to others. When the control points represent voxels in a closed 3-D region, the region can be represented by a parametric surface if the voxels can be mapped to a sphere. RaG surfaces as formulated by (1), (2), and (4) can be used to represent surfaces with a spherical parametrization also. Assuming parameters  $\phi \in [-\pi/2, \pi/2]$  and  $\theta \in [0, 2\pi]$  represent spherical coordinates of voxels in an object, we will need to set  $u = (\phi + \frac{\pi}{2})/\pi$  and  $v = \theta/2\pi$  in the equations of a RaG surface. In the following section, we will show how to parametrize voxels in a closed digital shape, and in the subsequent section, we will show how to find the control points of a RaG surface approximating a digital shape by the least-squares method.

### 3 Parametrizing Voxels in a Digital Shape

Brechbühler *et al.* [2] [3] described a method for mapping simply connected shapes to spheres through an optimization process. Although this method can find parameters of voxels in various shapes, the process is very time consuming. In the following, a coarse-to-fine parametrization is introduced that starts by approximating a digital shape by an octahedron. At the same time, a sphere is approximated by an octahedron and correspondence is established between triangles in the shape approximation and triangles in the sphere approximation. By knowing parameters of octahedral vertices in the sphere approximation,

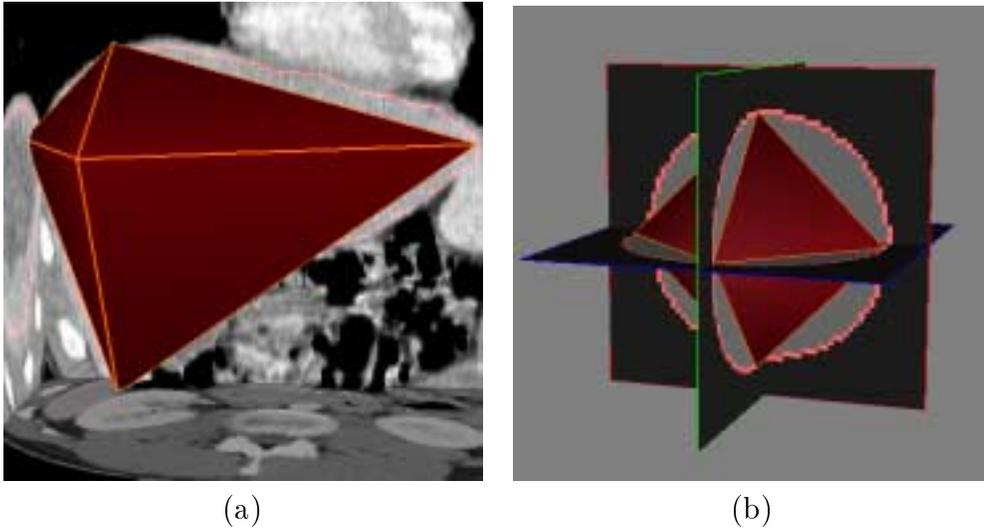


Figure 2: (a) Approximation of a digital shape by an octahedron. (b) Approximation of a sphere by an octahedron.

parameters of octahedral vertices in the shape approximation will be known. This coarse approximation step is depicted in Fig. 2. The major axis of the octahedron is aligned with the major axis of the shape [9] to achieve the highest overlap between the shape and the octahedron.

Next, the voxels associated with each triangle in the octahedral approximation are determined. This is achieved by finding the bisecting plane of each octahedral edge and determining the shape voxels that lie in that plane. In Fig. 3b, the bisecting planes passing through the edges of a triangle and intersecting the shape are shown. The bisecting plane passing through each octahedral edge and intersecting the shape will be an edge contour. A triangle, therefore, will produce three contours sharing the vertices of the triangle and delimiting the shape voxels that belong to that triangular face in the octahedral approximation.

If distances of voxels in an edge contour to the associating edge are all within the required tolerance, that edge is not subdivided. Otherwise, the farthest voxel in the contour to the edge is used to produce two smaller edges by connecting the voxel to the end points of the edge. Therefore, a triangular face may be subdivided into 2, 3, or 4 smaller triangles depending on whether 1, 2, or 3 edges of the triangle are replaced with smaller edges. This is depicted in Figs. 4a-c. If distances of voxels in edge contours to corresponding edges are below the required tolerance, a test is performed to determine whether distances of voxels associated with a triangle are within a required tolerance to that triangle or not. If all distances are below the required tolerance, the subdivision is halted. Otherwise, the farthest

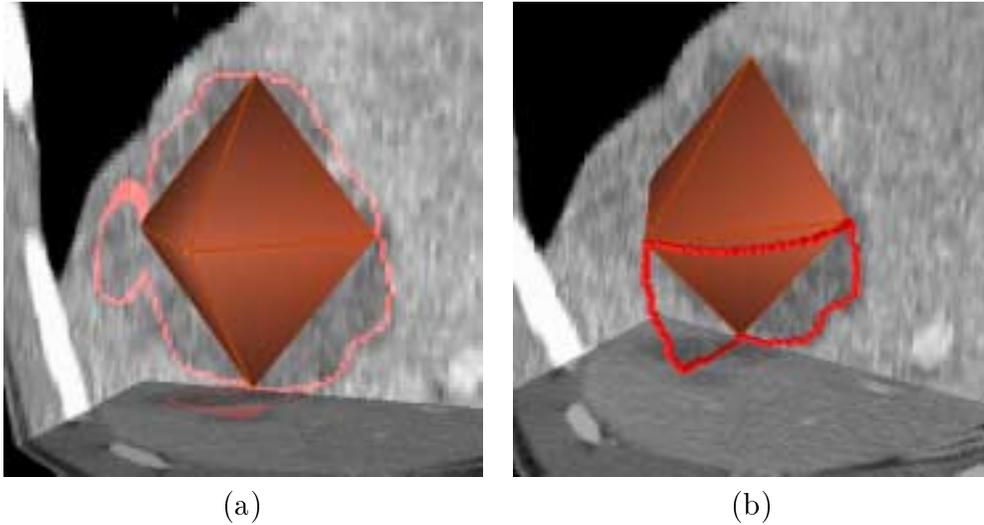


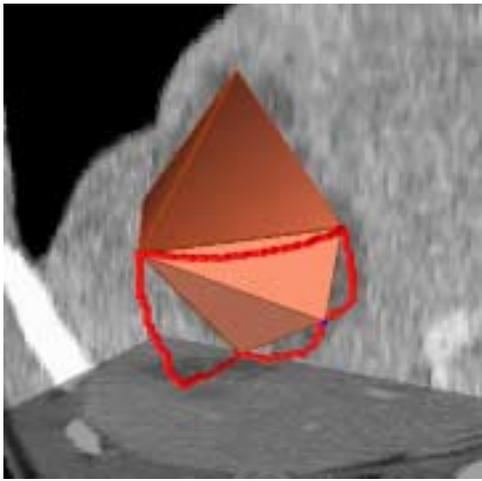
Figure 3: (a) Octahedral approximation of a shape. (b) Edge contours delimiting a triangular patch.

voxel from the triangular face is connected to the three vertices of the triangle and the triangle is subdivided into 3. This is depicted in Fig. 4d. The reason for subdividing the edge contours first is to avoid very long and narrow triangles in the subdivision.

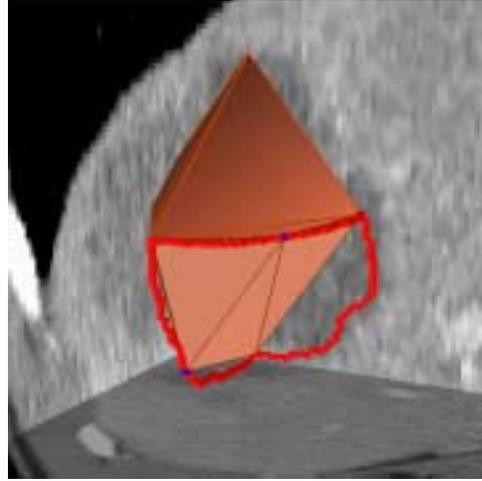
By subdividing existing triangles, finer triangles are obtained. The process is repeated until distances of all shape voxels to the associating triangles become smaller than the required tolerance. Note that this subdivision is performed in parallel in the sphere also. Therefore, whenever a triangle in the shape approximation is subdivided, the corresponding triangle in the sphere approximation is also subdivided. Therefore, 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. This process, in effect, assigns spherical parameters to mesh vertices approximating a digital shape. The only requirement of this parametrization is for the input shape to be closed and have no holes or loops.

We will use the vertices of the triangular mesh approximating a shape as the control points of a RaG surface to approximate the shape, and we will use the spherical parameters obtained at the vertices as the nodes of the surface. Note that by knowing parameters at vertices of a triangle, we will know the parameters at points inside the triangle via barycentric coordinates [14], and parameters of voxels in a triangular patch can be obtained from parameters of points in the associating triangle closest to the voxels.

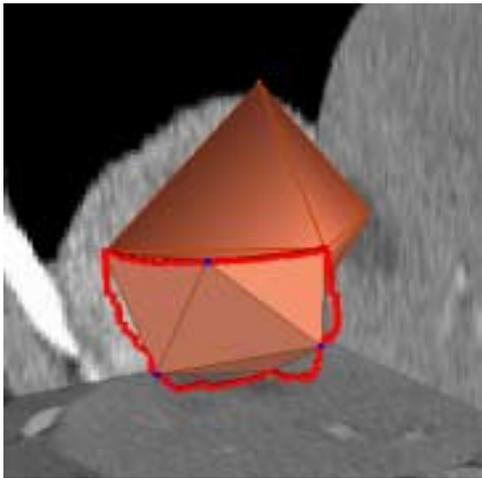
If the vertices of a triangular mesh approximating a digital shape are used as the control



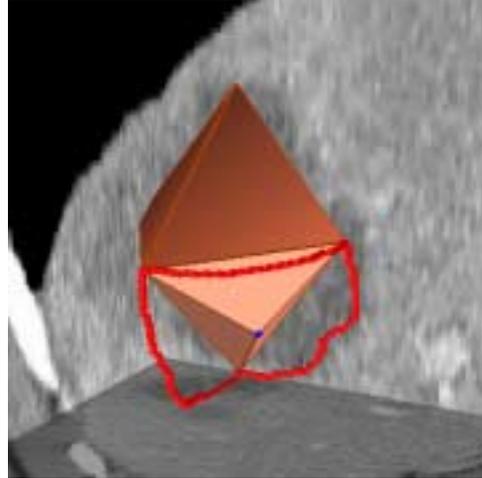
(a)



(b)



(c)



(d)

Figure 4: (a)–(c) Subdividing one, two, or three of the triangular edges. (d) When no triangular edges are subdivided but error between the triangular patch and the triangle is above the required tolerance, the farthest voxel in the patch to the triangle is used to subdivide the triangle.

points of a RaG surface and parameters at mesh vertices are used as the corresponding nodes, a smooth parametric surface will be obtained that approximates the shape. The surface obtained in this manner will approximate the mesh vertices. We can improve this approximation by using all voxels in the shape. In the following, a least-squares method is introduced that determines the control points of a RaG surface using parameters at the mesh vertices as the nodes of the surface.

### 3.1 Least-Squares Computation of RaG Control Points

Suppose a digital shape is obtained by segmenting a volumetric image automatically and the shape voxels are parametrized according to the procedure outlined in the preceding section. Suppose there are  $N$  voxels:  $\{\mathbf{P}_j : j = 1, \dots, N\}$  with parameters  $\{(u_j, v_j) : j = 1, \dots, N\}$ . These parameters determine the adjacency relation between the voxels. We would like to determine a RaG surface with control points  $\{\mathbf{V}_i : i = 1, \dots, n\}$  that can approximate the voxels with least-squares error. Let's suppose  $\mathbf{P}_j = (X_j, Y_j, Z_j)$ ,  $\mathbf{P}(u, v) = [x(u, v), y(u, v), z(u, v)]$ , and  $\mathbf{V}_i = (x_i, y_i, z_i)$ . Then the mean-squared distance between the voxels and the approximating surface can be written as

$$E^2 = \frac{1}{N} \sum_{j=1}^N \{[x(u_j, v_j) - X_j]^2 + [y(u_j, v_j) - Y_j]^2 + [z(u_j, v_j) - Z_j]^2\} \quad (5)$$

$$= \left\{ \frac{1}{N} \sum_{j=1}^N [x(u_j, v_j) - X_j]^2 + \frac{1}{N} \sum_{j=1}^N [y(u_j, v_j) - Y_j]^2 + \frac{1}{N} \sum_{j=1}^N [z(u_j, v_j) - Z_j]^2 \right\} \quad (6)$$

$$= E_x^2 + E_y^2 + E_z^2 \quad (7)$$

Since the three components of the surface are independently defined, to minimize  $E^2$ , we will minimize  $E_x^2$ ,  $E_y^2$ , and  $E_z^2$ , separately. To minimize

$$E_x^2 = \frac{1}{N} \sum_{j=1}^N [x(u_j, v_j) - X_j]^2, \quad (8)$$

since

$$x(u_j, v_j) = \sum_{i=1}^n x_i g_i(u_j, v_j), \quad (9)$$

we minimize

$$E_x^2 = \frac{1}{N} \sum_{j=1}^N \left[ \sum_{i=1}^n x_i g_i(u_j, v_j) - X_j \right]^2. \quad (10)$$

This involves determining the partial derivatives of  $E_x^2$  with respect to the  $x_i$ 's, setting the partial derivatives to zero, and solving the obtained system of linear equations. In the same manner, the  $y$  and the  $z$  components of the control points can be obtained by minimizing  $E_y^2$

and  $E_z^2$ , respectively. Note that the above process positions the  $n$  control points of a RaG surface so as to approximate the  $N$  image voxels by minimizing the sum of squared errors.

Since shape voxels are mapped to a sphere, they have parameters  $u$  and  $v$  associated with them. Assuming the approximating surface is represented by  $\mathbf{P}(u, v)$ , the distance of voxel  $\mathbf{V}_{ij} = (x_{ij}, y_{ij}, z_{ij})$  to the surface is estimated from  $E(u_{ij}, v_{ij}) = \|\mathbf{V}_{ij} - \mathbf{P}(u_{ij}, v_{ij})\|_2$ . The adjacency information between the control points is contained in the  $u$  and  $v$  values of the nodes. Index  $i$  is arbitrary and the control points with their associated nodes can be reordered in equation (1) without having any effect in the obtained surface.

When the standard deviation of the RaG surface is very small, the obtained surface will follow individual voxels and is not desirable. The selected standard deviation of the Gaussians should be large enough to smooth digital and image noise in segmentation. Depending on the standard deviation chosen, a surface with a different level of smoothness will be obtained that approximates the same set of voxels. Usually, the standard deviation of Gaussians in a RaG surface should be made proportional to the average distance between adjacent nodes. The denser the control points, the smaller the distance between their nodes, and thus, a smaller standard deviation should be used in surface fitting. Experimental results show that standard deviations from the average distance between adjacent nodes to five times that are appropriate in surface fitting.

Figure 5a shows voxels in a region that represents the bounding surface of a brain tumor. Subdivision of this region to a triangular mesh with a tolerance of 3 pixels is shown in Fig. 5b. The tolerance shows the maximum distance between the given digital shape and the approximating triangular mesh. Approximation of the tumor with a RaG surface of standard deviation 0.002 is shown in Fig. 5c. Increasing the standard deviation to 0.0025, 0.003, and 0.004, we obtain the results shown in Figs. 5d, 5e, and 5f. Root-mean-squared-error (RMSE) as defined by  $E$  in formula (7) for Figs. 5c–f are 1.9725, 1.9711, 2.1439, and 2.2497 voxels, respectively. The RMSE obtained in a RaG approximation of a digital region is usually much smaller than the tolerance used to estimate the triangular mesh approximating the shape.

At a standard deviation representing the level of detail in a shape, the smallest surface-fitting error is achieved. This minimum error may be determined by a greedy algorithm. However, since the given region is known to contain errors, finding the surface that is closer to the given region than any other surface may not particularly be of 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 user can interactively choose the standard deviation that is believed to best reproduce details in a given region.

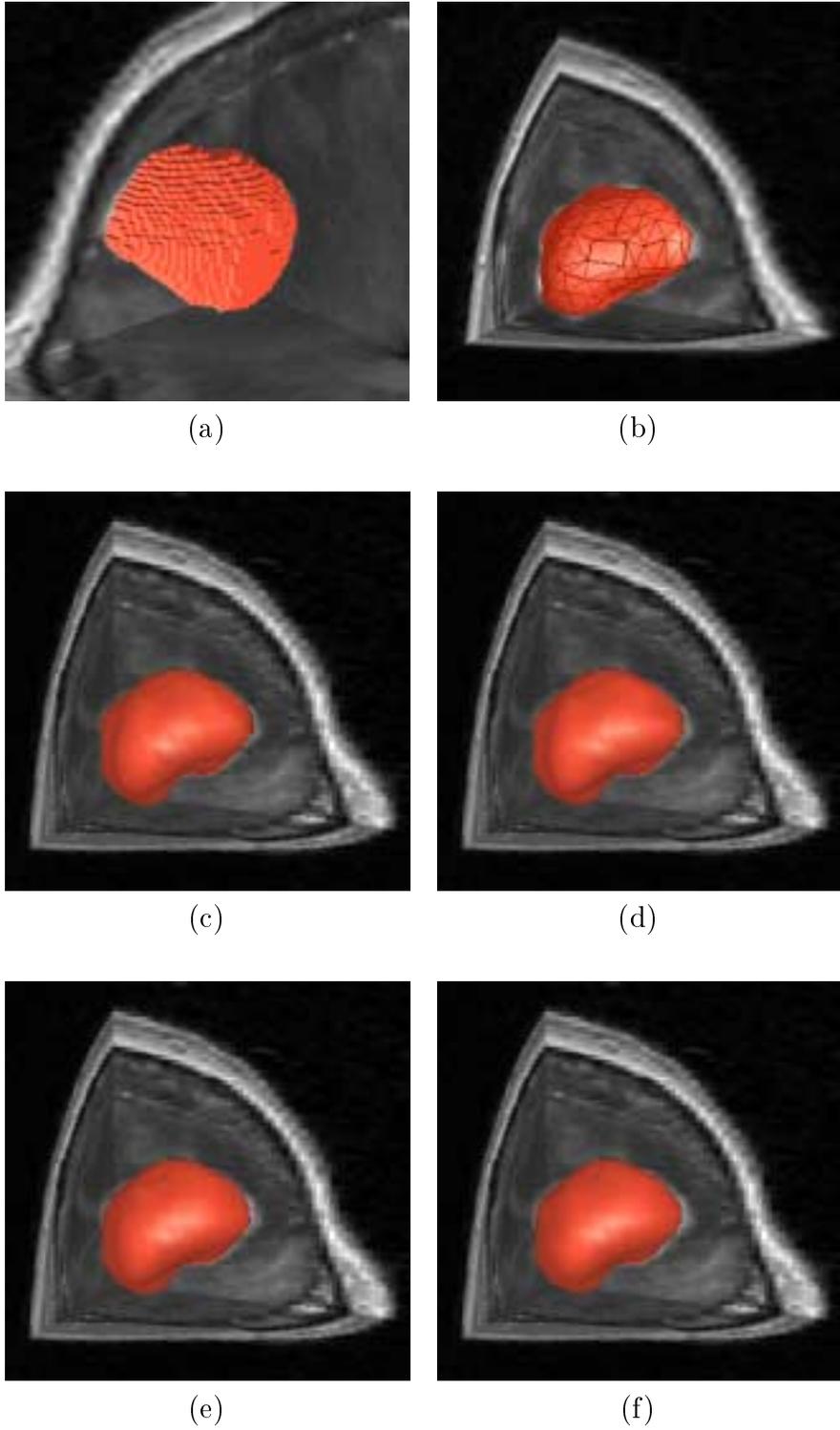


Figure 5: (a) A segmented brain tumor in an MR image. (b) Approximation of the tumor by a triangular mesh with a tolerance of 3 voxels. (c)–(f) RaG Surfaces approximating the tumor with standard deviations 0.002, 0.0025, 0.003, and 0.004, resulting in RMSE equal of 1.9725, 1.9711, 2.1439, and 2.2497, respectively.

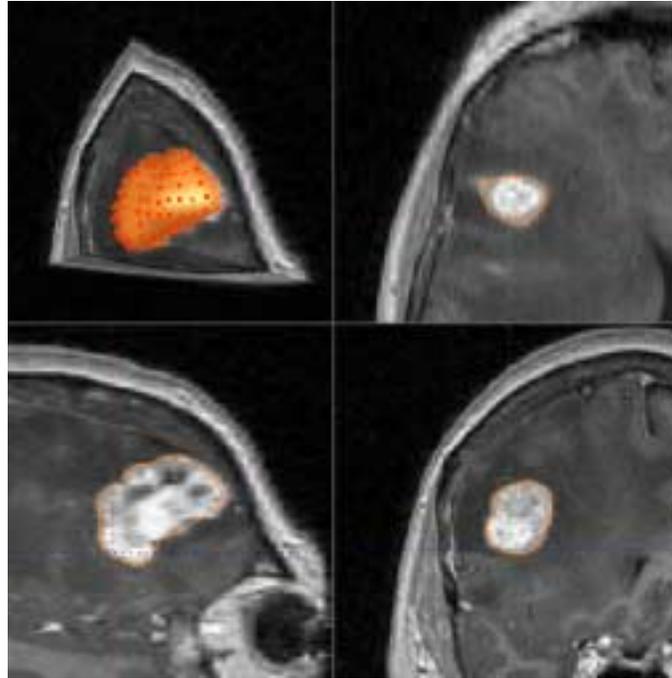
## 3.2 Local Shape Modification

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 a desired direction, the user will visually observe the intersections of the surface with the image slices and verify the correctness of the segmentation. When an error is observed, one or more of the control points are moved in the neighborhood of the error to revise the segmentation. As the control points are moved, the user will observe changes in the surface immediately. Movement of the control points is parallel to the plane of the slices. This allows modification of a surface with a 2-D mouse. The user has the option of choosing different slice directions when viewing and editing a surface.

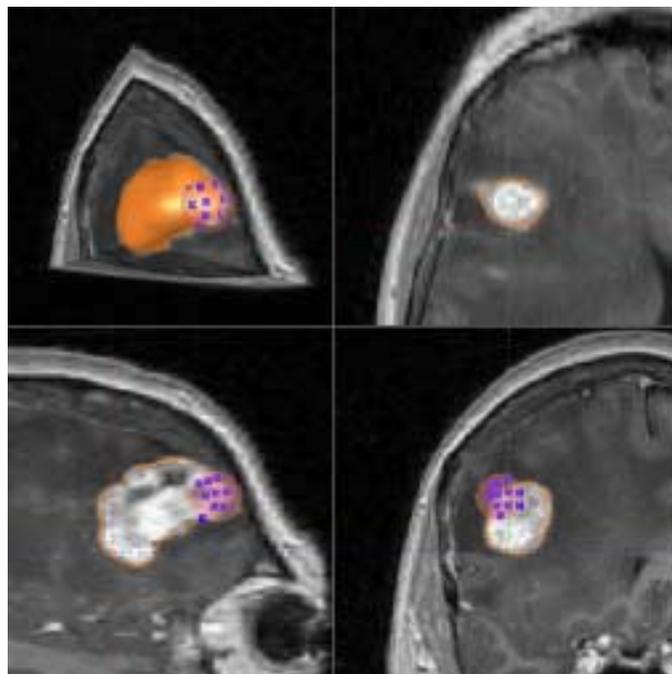
An example of shape editing by the proposed method is shown in Figure 6. Figure 6a shows overlaying of the surface approximating a brain tumor with the original volumetric image. The control points of the approximating surface are shown by small dots in this figure. The user selects a number of control points using a small sphere that is attached to the cursor and whose center lies in the image slice being viewed. By changing the radius of sphere, the number of control points selected for movement is changed. Control points selected by the sphere are then moved with the motion of the mouse. Control points inside the sphere are not all moved by the same amount and in the same direction. A point is moved in the direction obtained by connecting the point to the center of the sphere and by the amount proportional to the cosine of the angle between that direction and the direction of the mouse motion. Only those control points falling inside the hemisphere with positive cosines are moved. This avoids motion of control points with negative cosines in the opposing direction and also ensures that discontinuities will not occur between points that are moved and points that are not. Intermediate results in surface modification are shown in Figure 6b. Surface revision can be performed gradually and repeatedly while observing the image information. The sensitivity of the surface to the movement of the mouse can be controlled by increasing or decreasing the weights assigned to the control points. Figure 7a shows the tumor of Figure 6a after the necessary modifications. This surface can now be digitized as shown in Figure 7b to produce the revised segmentation result in digital form.

## 4 Results

A few examples of image segmentation by the proposed method are shown in Figure 8. The first column shows the original images, the second column shows initial segmentation results,

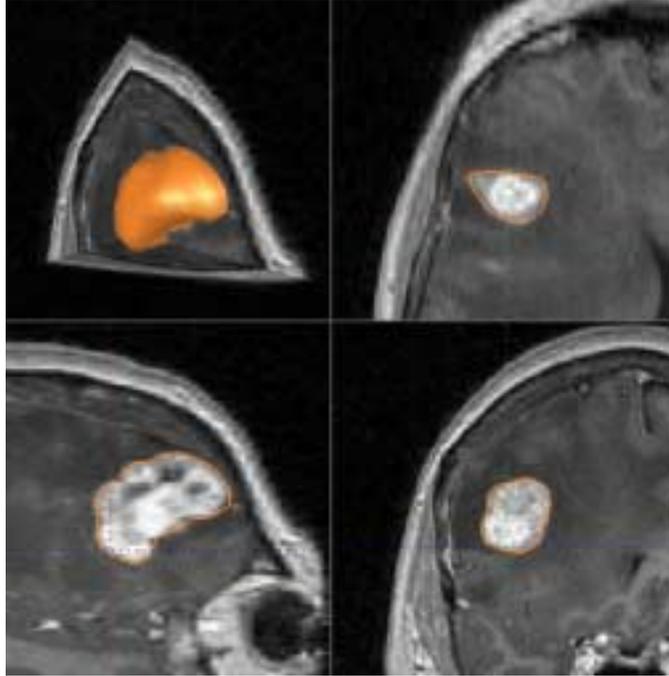


(a)

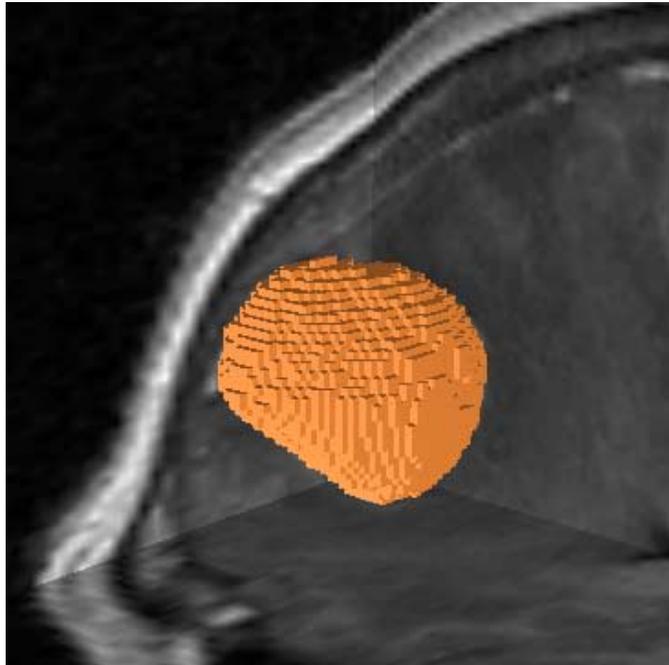


(b)

Figure 6: (a) Overlaying of the approximated tumor surface with the original volumetric image. The dots in the upper-left window show the control points of the surface. Four simultaneous displays provide the interface to the program. The user can interactively revise the surface using any or all of the windows. (b) The surface during a local modification. A small sphere attached to the cursor selects an appropriate number of control points for editing. As the user performs editing in one window, result of editing will appear in all four windows.



(a)



(b)

Figure 7: (a) The tumor after all modifications. (b) Digitization of the surface in (a). This is the final segmentation result in digital form.

and the third column shows results after interactive modifications. The images represent a short-axis cardiac MR image (first row), an MR brain image containing a tumor (second row), an MR image containing only the brain (third row), and a PET image of the head (fourth row). The ventricular blood pool, the brain tumor, and the brain were initially determined by smoothing and intensity thresholding. The PET image was initially segmented by our 3-D implementation of the Canny edge detector [5]. In these figures, results of the initial segmentation are shown after fitting RaG surfaces to the obtained regions by the least-squares method as outlined above. RaG surfaces were then interactively revised as needed while viewing the overlaid surfaces and the original volumetric images. Final segmentation results are shown in the third column of Figure 8.

Another set of examples is shown in Figures 9 and 10. Figure 9 is a chest CT image. Segmentation of this image via a combination of thresholding and edge detection, subdivision of obtained regions into triangular meshes, and fitting of RaG surfaces to the regions are shown in Figure 10. Regions corresponding to the liver, one of the kidneys and spleen were selected one at a time and after representing each by a RaG surface was edited to remove inaccuracies in segmentation. Final segmentation results are shown in the right column in Figure 10.

The time needed to obtain an initial segmentation and modify the initial segmentation to obtain the final result varies from image to image. In images shown in Figs. 8 and 10, 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. 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.

The final result of a segmentation obtained by the proposed environment is user dependent. In a typical image, the user judges what a correct segmentation is based on his/her past knowledge and experience in image segmentation and also on the information present in an image. Since users have different experiences and knowledge in imaging, results obtained by different users will be different. Even the same user may segment an image differently at different times. The intra-user variability and inter-user variability are not characteristics of the proposed environment, but those of a user or users. The proposed environment will provide to a user tools with which he/she can modify the result of a segmentation in any way desired. There are no limitations in shape, size, or complexity of a region under consideration. The only requirements are that the region be closed and have no holes or loops.

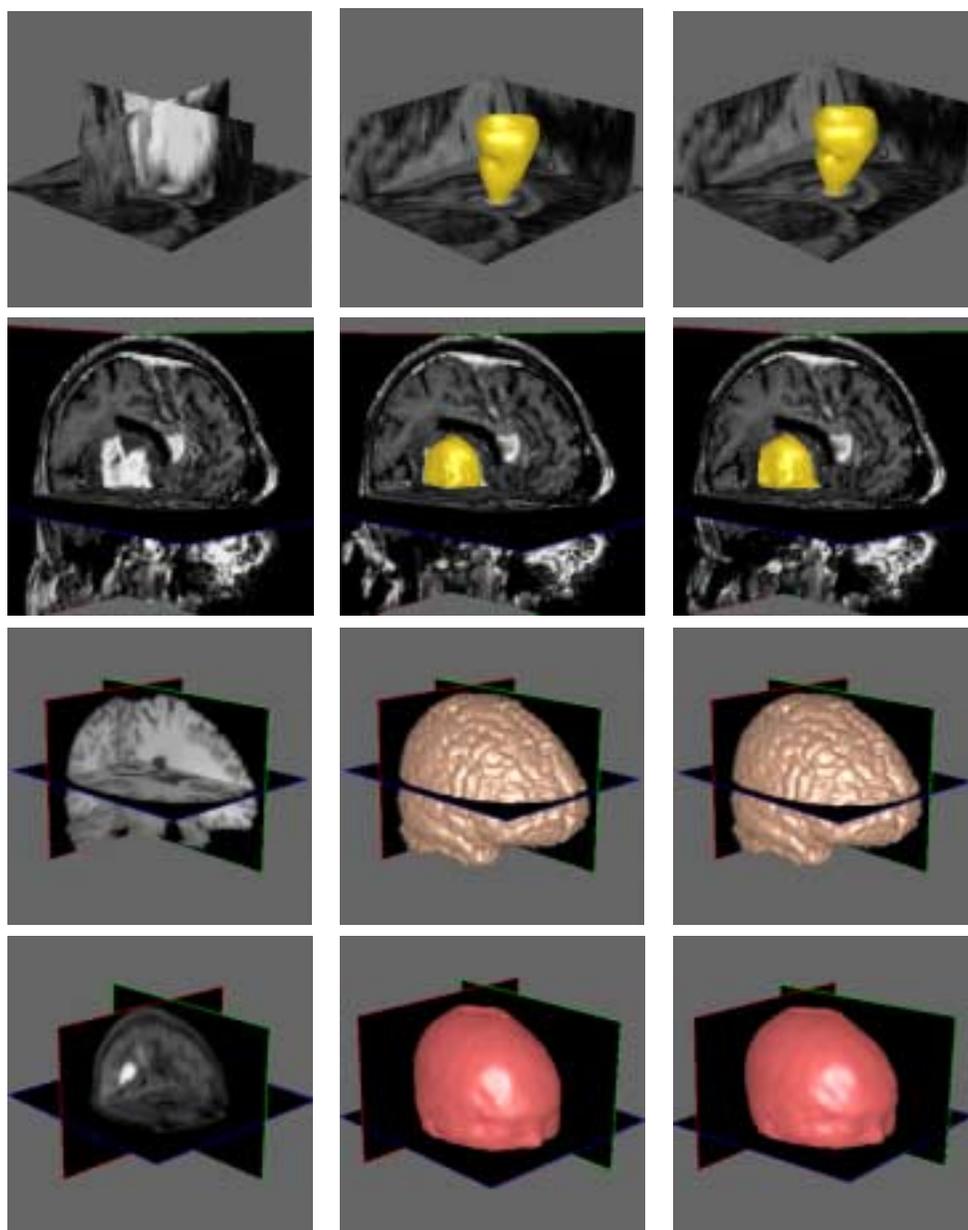


Figure 8: First row: Short-axis cardiac MR image and segmentation of left ventricular cavity; Second row: MR brain image and segmentation of the tumor; Third row: MR brain image and segmentation of the brain; Fourth row: PET image and extraction of the surface of the head. The first column shows the original images, the middle column shows the initial segmentation results, and the right column shows the results after necessary modification.

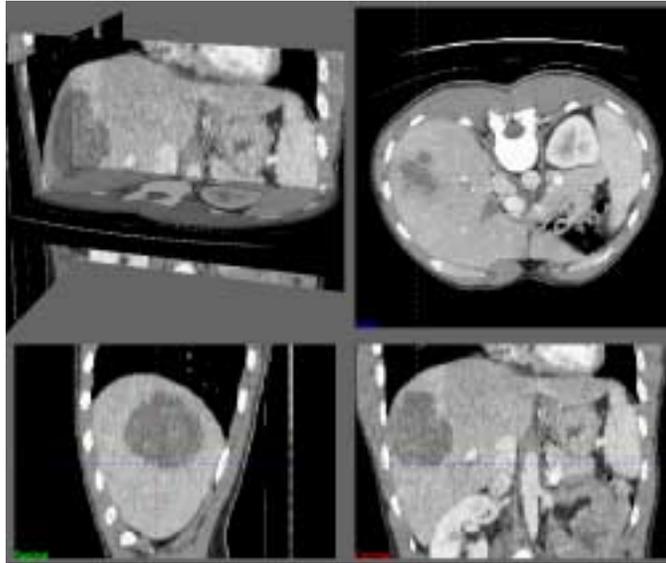


Figure 9: A chest CT image.

## 5 Conclusions

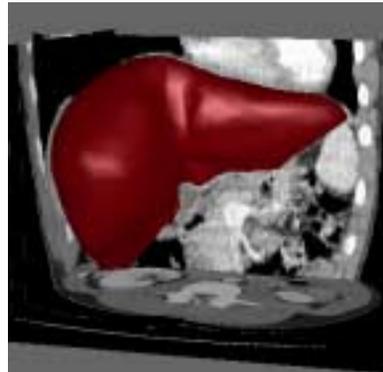
Image segmentation is the single most important component of any image analysis system. In medical images, it is essential that an image be accurately segmented so that quantification and analysis of the image can be successfully achieved. In this paper, the idea of using a computer-aided design environment to effectively revise the result of a segmentation obtained by an automated method was introduced. In the proposed environment, a RaG surface is fitted to voxels in a region by the least-squares method. The RaG surface and the original volumetric image are then overlaid, and the surface is interactively revised until desired segmentation result as judged by the user is achieved.

The described computer-aided design environment requires only a fraction of the time needed to segment a 3-D segmentation slice by slice. Although modification is done in 3-D, the user can go through image slices in a desired direction and review the result as it is being modified. Since the final result is in surface form, a smooth rendering of the segmented region is possible. This representation also enables rendering of a segmented region in high resolution.

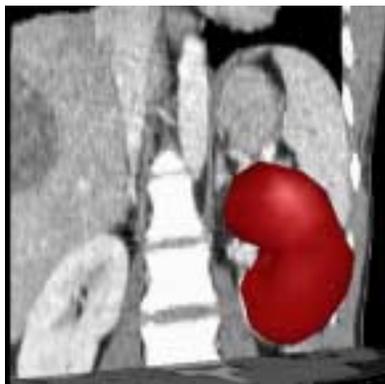
The system provides the option of using the output of an automatic segmentation method as its input or manually creating an initial segmentation by selecting a number of 3-D points in the given image volume. In the latter case, an initial surface is created from the points and overlaid with the image. The surface is then revised to a desired shape while observing the image data. Because a region of interest is represented by a parametric surface, the



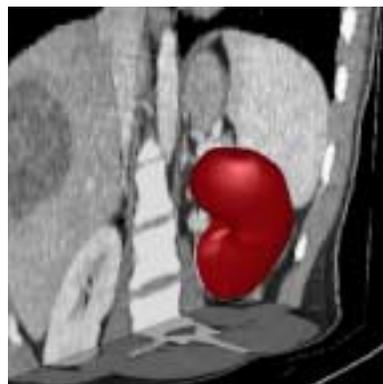
(a)



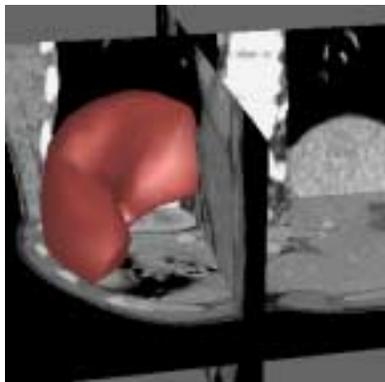
(b)



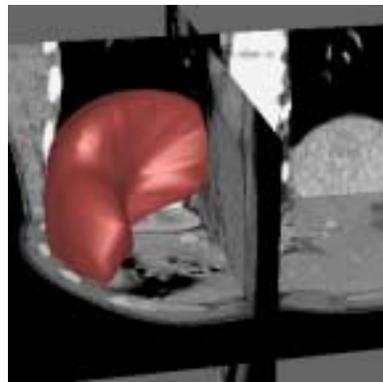
(c)



(d)



(e)



(f)

Figure 10: First row: Polygon mesh and RaG approximation of the liver. Second row: Polygon mesh and RaG approximation of the kidney. Third row: Polygon mesh and RaG approximation of the spleen.

surface can be sent to a computer-aided manufacturing system for construction of an actual 3-D model of the region.

## Acknowledgements

This work was supported in part by the Air Force Research Laboratory (AFRL/HEOP) Air Force Material Command, USAF, under cooperative agreement F33615-98-2-6002 and by the Wallace-Kettering Neuroscience Institute. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The image used in Figure 9 is from Milan Sonka, University of Iowa. The cardiac MR image in Figure 8 is from David Turner, Presbyterian-St. Luke's Hospital, Chicago, IL. The MR brain image containing only the brain in Figure 8 is prepared by Terry Oaks, University of Wisconsin, and the rest of the images were provided by Kettering Medical Center, Kettering, OH. We appreciate their contributions.

## References

- [1] W. Barrett and E. Mortensen, Fast, accurate, and reproducible live-wire boundary extraction, *Proc. Visualization in Biomedical Computing*, Hamburg, 1996.
- [2] Ch. Brechbühler, G. Gerig, and O. Kübler, Surface parametrization and shape description, *SPIE Workshop on Visualization in Biomedical Computing*, vol. 1808, 1992, pp. 80–89.
- [3] Ch. Brechbühler, G. Gerig, and O. Kübler, Parametrization of closed surfaces for 3-D shape description, *Computer Vision and Image Understanding*, vol. 61, no. 2, 1995, pp. 154–170.
- [4] 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*, 1993, pp. 171–179.
- [5] J. Canny, A Computational Approach to Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, 1986, pp. 679–698.
- [6] A. X. Falcão and J. K. Udupa, A 3-D generalization of user-steered live-wire segmentation, *Medical Image Analysis*, vol. 4, 2000, pp. 389–402.
- [7] A. X. Falcão, 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*, vol. 60, no. 4, 1998, pp. 233–260.
- [8] A. X. Falcão, J. K. Udupa, and F. K. Miyazawa. An ultra-fast user-steered image segmentation paradigm: live-wire-on-the-fly, *SPIE Medical Imaging Conference, Proc. Image Processing*, vol. 3661, San Diego, CA, 1999, pp. 184–191.
- [9] J. M. Galvez and M. Canton, Normalization and shape recognition of three-dimensional objects by 3-D moments, *Pattern Recognition*, vol. 26, no. 5, 1993, pp. 667–682.
- [10] A. Goshtasby, Design and recovery of 2-D and 3-D shapes using rational Gaussian curves and surfaces, *Int. J. Computer Vision*, vol. 10, no. 3, 1993, pp. 233–256.
- [11] A. Goshtasby, Geometric modeling using rational Gaussian curves and surfaces, *Computer-Aided Design*, May 1995, pp. 363–375.

- [12] 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.), 1997, pp. 236–242.
- [13] K. H. Höhne and W. A. Hanson, Interactive 3D segmentation of MRI and CT volumes using morphological operations, *Journal of Computer Assisted Tomography*, vol. 16, no. 2, 1992, pp. 258–294.
- [14] J. Hoschek and D. Lasser, *Computer Aided Geometric Design*, A. K. Peters, 1989, pp. 289–291.
- [15] A. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active contour models, *Int'l J. Computer Vision*, vol. 1, 1987, pp. 321–331.
- [16] T. McInerney and D. Terzopoulos, A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4-D image analysis, *Computerized Medical Imaging and Graphics*, vol. 19, no. 1, 1995, pp. 69–83.
- [17] E. N. Mortensen and W. A. Barrett, Interactive segmentation with intelligent scissors, *Graphical Models and Image Processing*, vol. 60, no. 5, 1998, pp. 349–384.
- [18] 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*, 1992, pp. 635–638.
- [19] 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, 1990, pp. 83–105.
- [20] J. K. Udupa, S. Samarasekera, and W. A. Berrett, Boundary detection via dynamic programming, *Proc. SPIE*, vol. 1808, 1992, pp. 33–39.
- [21] D. Welte, T. Grunert, U. Klose, D. Petersen, and E. Becker, Interactive 3D segmentation and visualization of vessels, *Computer Assisted Radiology*, 1996, pp. 329–335.