# Approximating Digital Shapes by Parametric Surfaces

A. Ardeshir Goshtasby

**Abstract.** A method for parametrizing discrete points sampled from a smooth shape and fitting a rational Gaussian surface to the points with a required accuracy is presented. With the proposed method, a very complex shape can be represented by a single surface, and the surface can be rendered at a desired level of detail by adjusting a smoothness parameter.

## §1. Introduction

Given a set of scattered points in 3-D, $\{\mathbf{p}_i = (x_i, y_i, z_i) : i = 1, \ldots, n\}$, we would like to determine a parametric surface $\mathbf{P}(u, v)$ that approximates the points with a required error tolerance:

$$\max_i ||\mathbf{P}(u_i, v_i) - \mathbf{p}_i|| < \varepsilon \tag{1}$$

$\varepsilon$ is the error tolerance and $(u_i, v_i)$ are the parameters at $\mathbf{p}_i$.

We will consider a special case where the given points are voxels covering an object in a discretized 3-D space. We will call such a data set a *digital shape*. Digital shapes are typically obtained by segmenting tomographic images obtained by industrial or medical scanners. We assume that the given shape is closed and contains no holes.

The objective in this approximation is threefold. First, we want to approximate the points with a parametric surface where the number of control points in the surface is much smaller than the number of points in the shape, and maximum distance between the shape points and the surface is within a required tolerance. Second, we want to have the ability to revise the obtained surface so that a reconstructed shape can be edited. Therefore, the surface formulation used should lend itself to easy editing. Third, the obtained surface should smooth noise among the given points, with the degree of smoothing adjustable by the user.

In the following sections, first, an algorithm to parametrize a set of points by mapping them to a sphere is described. Then, surface fitting using rational Gaussian (RaG) surfaces is discussed. Finally, examples of the proposed surface-fitting method using medical data are presented.

Mathematical Methods in CAGD: Oslo 2000
Tom Lyche and Larry L. Schumaker (eds.), pp. 1–10.


ISBN 0-8265-xxxx-x.

## §2. Definitions

In this section, terminologies used in the paper are defined.

**Digital shape:** A set of points covering a smooth shape in a dicretized 3-D space.

**Point:** A point in a digital shape. Shape points will be denoted by $\mathbf{p}$'s.

**Adjacent points:** Two points, $\mathbf{p}_i$ and $\mathbf{p}_j$, are considered adjacent if $1 \leq ||\mathbf{p}_i - \mathbf{p}_j|| \leq \sqrt{3}$. Two adjacent points are also called **neighbors**.

**Connected points:** Two points are said to be connected if a path can be formed between them by connecting adjacent points.

**Path:** A path between points $\mathbf{p}_i$ and $\mathbf{p}_j$ is a connected set of points starting from $\mathbf{p}_i$ and ending at $\mathbf{p}_j$ where no point is repeated and every point has exactly two neighbors except for $\mathbf{p}_i$ and $\mathbf{p}_j$, which may have only one neighbor. A path is also called a **contour**.

**Triangular mesh approximation of a shape:** A triangular mesh that interpolates some points and approximates the rest in a shape. The mesh vertices will be denoted by $\mathbf{P}$'s. Note that $\mathbf{P}$'s are a subset of the $\mathbf{p}$'s.

**Edge contour:** A contour in a digital shape that is delimited by the end points of a mesh edge and lies in the plane passing through the edge and bisecting the angle between the two triangular faces that share the edge. Note that due to the digital nature of shape points, some points in an edge contour may not fall exactly in the bisecting plane, however, they will be closer to the plane than points in any other path connecting the edge end points.

**Distance of point $\mathbf{p}_i$ to edge $\mathbf{P}_j\mathbf{P}_k$:** Assuming the plane passing through the point and normal to the edge intersects the edge at $\mathbf{P}_l$, if $\mathbf{P}_l$ is between $\mathbf{P}_j$ and $\mathbf{P}_k$, the distance will be $||\mathbf{p}_i - \mathbf{P}_l||$. Otherwise, if $\mathbf{P}_l$ is closer to $\mathbf{P}_j$ than to $\mathbf{P}_k$, the distance will be $||\mathbf{p}_i - \mathbf{P}_j||$, and if $\mathbf{P}_l$ is closer to $\mathbf{P}_k$ than to $\mathbf{P}_j$, the distance will be $||\mathbf{p}_i - \mathbf{P}_k||$.

**Distance of an edge to an edge contour:** Assuming $\mathbf{p}_i$ is a contour point with distance $d_i$ to the associating edge, we will take the maximum distance from points on the contour to the edge as the distance of the contour to the edge. That is, $D_e = \max_i\{d_i\}$.

**Distance of Point $\mathbf{p}_i$ to Triangular Face $\mathbf{P}_j\mathbf{P}_k\mathbf{P}_l$:** Assuming the line passing through $\mathbf{p}_i$ and normal to the triangle interests the triangle at $\mathbf{P}_m$, if $\mathbf{P}_m$ is inside the triangle, the distance will be $||\mathbf{p}_i - \mathbf{P}_m||$. If $\mathbf{P}_m$ is outside the triangle and assuming $\mathbf{P}_n$ is the point on a triangle edge closest to $\mathbf{P}_m$, the distance will be $||\mathbf{p}_i - \mathbf{P}_n||$.

**Triangular patch:** A connected set of points in a digital shape delimited by three contours whose end points are the vertices of a triangle.

**Distance of a triangular patch to the associating triangle:** Assuming point $\mathbf{p}_i$ belongs to the triangular patch and the distance between $\mathbf{p}_i$ and the triangle is $d_i$, the distance to be determined is the maximum of such distances when all points in the patch are tested. That is, $D_t = \max_i\{d_i\}$.

**Major axis of a digital shape:** This is the axis defined by the largest eigenvector of the inertia matrix [2] of points defining the shape.

**Subdividing edge $\mathbf{P}_j\mathbf{P}_k$:** Replacing the edge with edges $\mathbf{P}_j\mathbf{P}_i$ and $\mathbf{P}_i\mathbf{P}_k$, where $\mathbf{P}_i$ is the farthest point on the associating edge contour to the edge and distance of $\mathbf{P}_i$ to the edge is larger than the given tolerance $\varepsilon$.

**Subdividing a triangle:** A triangle may be subdivided in four different ways: 1) If distances between all three edges of the triangle and the corresponding contours are larger than the specified tolerance, then by subdividing each edge into two and connecting the obtained points to each other and to the vertices of the triangle, four smaller triangles are obtained. 2) If distances between two of the edges and corresponding contours are larger then the specified tolerance, then two of the edges are subdivided. By connecting the newly obtained points to each other and to the vertices of the triangle, three new smaller triangles are obtained. 3) If the distance between only one of the edges and the corresponding contour is larger than the required tolerance, then only one of the edges is subdivided into two. By connecting the newly obtained point to the opposing triangle vertex, two smaller triangles are obtained. 4) If distances between all three edges and the corresponding contours do not reach the required tolerance, then the distance between the patch and the triangle is determined, and if it is larger than the required tolerance, the point in the patch farthest from the triangle is connected to the vertices of the triangle to produce three smaller triangles. In this way, a triangle is subdivided into 2, 3, or 4 smaller triangles. Subdivision will take place in the order specified above. That is, only if subdivision by case 1 is not possible subdivision by case 2 will be considered, and subdivision by case 3 will be considered only when subdivision by case 2 is not possible, and so on.

**Parametrizing points in an edge contour:** By knowing the parameters at the end points of an edge, parameters along the edge can be determined by linear interpolation. Parameters at a contour point are then set equal to the parameters at the edge point closest to it. At a coarse resolution, multiple contour points may map to the same edge point, thus receiving the same parameters. However, as the subdivision proceeds, contour points will more likely map to unique edge points, producing unique parameters.

**Parametrizing points in a triangular patch:** By knowing parameters at the vertices of a triangle, parameters of points in the triangle and along its edges can be determined from the barycentric coordinates [5, 289–291]. Parameters of a point in a triangular patch are set equal to the parameters of the point colsest to it in the associating triangle. Initially, depending on the complexity of a shape, some points in a patch may receive the same parameters. However, as the subdivision proceeds the probability of such cases decreases, producing unique parameters for points in a patch.

## §3. The Subdivision Algorithm

Using the above definitions, we now describe an algorithm that approximates a digital shape by a triangular mesh with a required accuracy. We start by approximating the shape with an octahedron. Then, we subdivide the triangular faces into smaller triangles until error in the approximation reaches a required tolerance.

**Algorithm 1: Subdivision of a digital shape to a triangular mesh**

1. **Initialization:** Determine the major axis of the shape and approximate the shape with an octahedron whose major axis lies on the shape's major axis and whose vertices lie on the shape. Then, enter the triangular faces of the obtained octahedron into a list.
2. **Main Step:** Remove a triangle from the list. If the distance between the triangle and the corresponding patch is larger than the required tolerance, subdivide it and enter the newly obtained triangles into the list.
3. **Stopping Criterion:** If the list is empty, stop. Otherwise, go to the Main Step.

The reason for orienting the octahedron so that its major axis lies on the major axis of the shape is to maximize overlap between the shape and the octahedron and, thereby, minimize the distance between the approximating mesh and the shape. When subdivision is complete, the maximum distance between the shape and its approximating mesh is guaranteed to be smaller than the required tolerance.

Some of the properties of this approximation are: a) A unique subdivision is obtained independent of the orientation or position of a shape. This is achieved by aligning the major axis of the octahedron with the major axis of the shape. b) The process avoids subdivision into triangles with acute angles or long edges. This is achieved by subdividing the edges of the mesh first. c) Compression rate depends on the complexity of the shape. During subdivision, large triangles are generated at smooth areas and small triangles are created at detailed areas. The process automatically adjusts triangle sizes to reproduce local details in a shape.

## §4. Parametrizing the Shape Points

To parametrize the mesh vertices, first, parameters at the octahedral vertices approximating the shape are determined. This is achieved by fitting an octahedron to a sphere, establishing correspondence between vertices in the shape approximation and in the sphere approximation, and assigning parameters of mesh vertices in the sphere approximation to parameters of mesh vertices in the shape approximation. As a triangle in the shape approximation is subdivided, the corresponding triangle in the sphere approximation is subdivided also and, again, parameters at newly obtained mesh vertices in the sphere are assigned to corresponding mesh vertices in the shape.

When an edge contour in the shape approximation is divided into two, in the sphere approximation, the corresponding arc is divided into two in such a way that the proportion of the lengths of newly obtained arcs would be the same as the proportion of the lengths of contour segments in the shape. At any stage of the process, by knowing parameters of mesh vertices in the sphere, parameters of corresponding mesh vertices in the shape will be known.

Therefore, when the subdivision ends, spherical parameters of all mesh vertices approximating a shape will be known. Algorithm 1, therefore, provides the means to determine the parameters at vertices of the triangular mesh approximating a shape. By knowing parameters at three vertices of a triangle, parameters at points in the triangle can be determined using the barycentric coordinates, and by knowing the parameters of points in a triangle, parameters of points in the associating triangular patch can be determined from the correspondence between the two. In this manner, the spherical parameters of all points in a digital shape can be determined.

## §5. Approximating a Digital Shape by a Rational Gaussian Surface

Knowing the coordinates and the parameters of points in a digital shape, we can find a smooth parametric surface that approximates the shape. We will use the mesh vertices as the control points of the parametric surface. Since the vertices are irregularly spaced, we will need a surface formulation that does not require a regular grid of control points. A rational Gaussian (RaG) surface [3,4] can have irregularly spaced control points and, therefore, will be used in this approximation. Assuming vertices of the triangular mesh obtained by Algorithm 1 are $\{\mathbf{P}_i : i = 1, \ldots, N\}$ and the parameters associated with them are $\{(u_i, v_i) : i = 1, \ldots, N\}$, a RaG surface that approximates the vertices can be written as [3,4]

$$\mathbf{P}(u, v) = \sum_{i=1}^{N} \mathbf{P}_i g_i(u, v), \tag{1}$$

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

$$g_i(u, v) = \frac{G_{\sigma_i}(u - u_i, v - v_i)}{\sum_{j=1}^{N} G_{\sigma_j}(u - u_j, v - v_j)}, \tag{2}$$

and $G_{\sigma_i}(u - u_i, v - v_i) = \exp\{[(u - u_i)^2 + (v - v_i)^2]/2\sigma_i^2\}$.

The standard deviations of Gaussians will be set in such a way to reproduce local shape details. The sizes of triangles obtained in Algorithm 1 contain information about local detaileds in a shape. We will set the standard deviations of Gaussians proportional to the perimeters of the triangles.

If the surface is required to interpolate the vertices, we let $\mathbf{P}(u_i, v_i) = \mathbf{P}_i$ and compute the control points of the surface, $\{\mathbf{V}_i : i = 1, \ldots, N\}$, from three systems of $N$ linear equations:

$$\mathbf{P}_i = \sum_{i=1}^{N} \mathbf{V}_i g_i(u, v), \qquad i = 1, \ldots, N. \tag{3}$$

Because of the nature of the rational Gaussian bases, the obtained matrix of coefficients will be diagonally dominant. For very large standard deviations, however, the system will become unstable because a surface with a desired smoothness may not be possible to fit to points in a very detailed area.

In Algorithm 1, the decision to subdivide a triangle was based initially on distances between edges of the triangle and the associating edge contours, and then on the distance between the triangle and the associating patch. In order to fit a RaG surface to a shape, we redefine the error criteria in Algorithm 1 as follows:

1. Instead of determining the distance between an edge and its corresponding edge contour, we will determine the distance between an edge contour and the approximating/interpolating surface. Since parameters of all points in an edge contour are known, for each point $\mathbf{p}_i$ in an edge contour, we can determine the corresponding point $\mathbf{P}(u_i, v_i)$ in the approximating/interpolating RaG surface. We then let the maximum distance between corresponding points in the contour and the surface be the distance between an edge contour and the RaG surface: $D_E = \max_i ||\mathbf{P}(u_i, v_i) - \mathbf{p}_i||$.
2. Instead of determining the distance between a triangle and its associating patch, we will determine the distance between the patch and the approximating/interpolating RaG surface. This is possible because, by knowing parameters $(u_i, v_i)$ at each point $\mathbf{p}_i$ in the patch, we can determine the corresponding point $\mathbf{P}(u_i, v_i)$ in the surface. We will then define the distance between a triangular patch and its approximating/interpolating RaG surface to be the maximum distance between corresponding points in the patch and the surface: $D_T = \max_i ||\mathbf{P}(u_i, v_i) - \mathbf{p}_i||$.

Replacing error measures $D_e$ and $D_t$ in Algorithm 1 with error measures $D_E$ and $D_T$, respectively, we will obtain an algorithm that fits a RaG surface to a digital shape, ensuring that distance between the given shape and the approximating/interpolating surface is within the required tolerance. We will call this new algorithm, **Algorithm 2**.

Note that $\sigma$'s in the RaG formulation are in the same unit as $u$'s and $v$'s. As the subdivision progresses, the sizes of triangles in the sphere subdivision become smaller. Roughly, the perimeter of a triangle reduces to half its size in each subdivision. Therefore, the $\sigma$ to be assigned at a particular subdivision level will depend on the depth of the subdivision.

Suppose at level 0 an octahedron is fitted to the shape and another octahedron is fitted to a sphere. If we were to stop the approximation at level 0, we would set all $\sigma$'s to 1 to obtain a smooth approximation to the shape. As the $\sigma$'s are reduced, the obtained surface will resemble the approximating triangular mesh, and as the $\sigma$'s are increased the surface approaches a sphere. In the case of interpolation, for very large values of $\sigma$'s a surface may not be possible to obtain that would pass through the points. Typically, proper values for the $\sigma$'s at level 0 are between 0.2 and 2.

At level 1, perimeters of triangles in the sphere are roughly half the perimeters of triangles at level 0. Therefore, $\sigma$'s at level 1 should be half the $\sigma$'s at level 0. Analogously, $\sigma$'s at level $n$ should be $2^{-n}$ of $\sigma$'s at level 0. Denoting the $\sigma$ assigned to vertices at the $i$th level by $s_i$, we will have $s_i = 2^{-i}s_0$. In this manner, $\sigma$ at all vertices will depend on a single parameter $s_0$. By adjusting this single parameter, the overall smoothness of the reconstructed surface can be controlled. At one extreme when $s_0$ is close to zero, the surface approaches the approximating triangular mesh. At the other extreme, when $s_0$ is very large, the shape will approach a sphere, and if an interpolating surface is required, beyond a certain point a surface may not be possible to obtain that would fit the mesh vertices.

Note that since $\sigma$'s are associated with the control points of a RaG surface, and the control points are the vertices of the approximating triangular mesh, whenever an edge is divided into two, $\sigma$'s associated with the mesh vertices corresponding to the end points of the edge are divided by two also. A vertex may be shared by many triangles obtained at different levels. Assigning a $\sigma$ to the vertex that is proportional to the triangle at the highest level will enable reproduction of details in a shape. This can be explained by examining the spatial frequency characteristics of Gaussians.

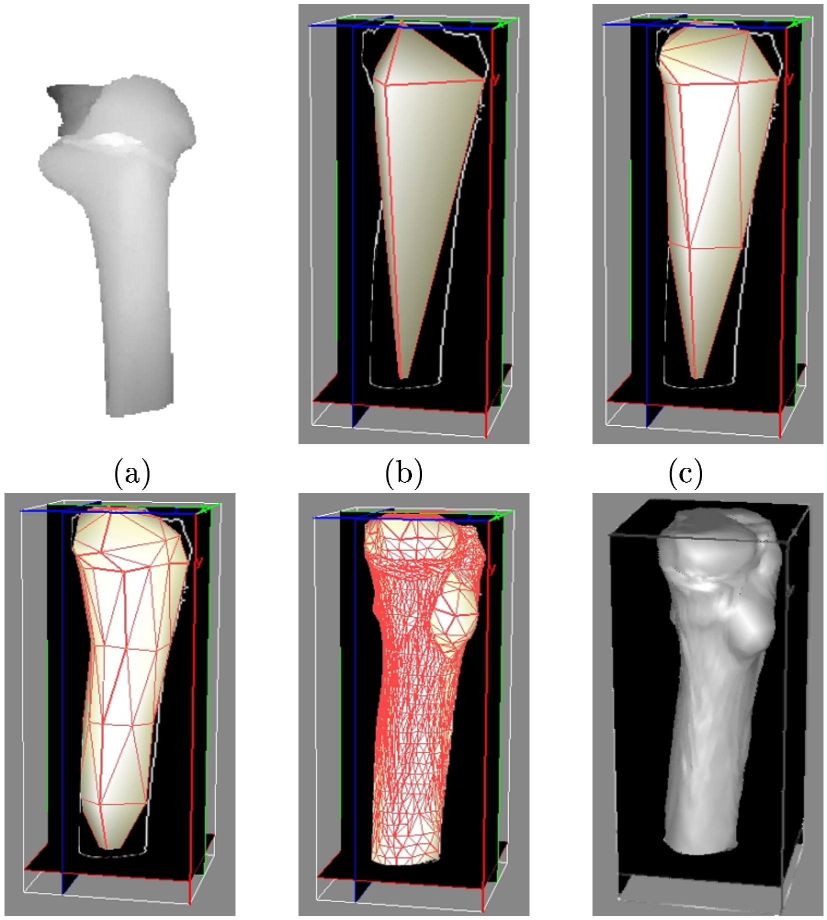
From the signal processing point of view, as the standard deviation of Gaussians in the spatial domain increases (decreases), the Fourier transform of the Gaussians, which are also Gaussians, will become narrower (wider) in the frequency domain. This means, in areas where smaller $\sigma$'s are used, the obtained surface can reproduce more high spatial frequencies (or details) in the surface, and in areas where only large $\sigma$'s are used, high spatial frequencies are not reproduced, thus creating a smooth surface. Narrower Gaussians enable reproduction of both low and high spatial frequencies. For any value of $s_0$, relative details obtained in different areas in a reconstructed surface will depend on relative details of local areas in the original shape.

Also note that although some triangles at level $n$ could have the same size as some triangles at levels $m < n$ in the $xyz$ space, but in the $uv$ space the sizes of triangles at level $n$ are smaller than those at level $m$. By appropriately reducing the $\sigma$'s at higher levels, we are in effect preserving information about the sizes of triangles at different levels. When a small portion of a sphere is mapped to a large portion of a shape, the $\sigma$'s assigned to different subdivision levels enable proper reproduction of details in the shape.

## §6. Examples

The digital shape depicted in Fig. 1a shows a femoral stem obtained by segmenting a volumetric X-ray Computer Tomography (CT) image. There were holes at the top and bottom of the original bone. The holes were covered with planar patches to obtain a closed shape. The process of subdividing the closed shape into triangles using Algorithm 1 with error tolerance of 0.5 units is shown in Figs. 1b–1e. It is assumed that the length of each side of elements (voxels) in the shape is 1 unit. Figure 1f shows the rendered femoral stem using the obtained triangular mesh. The triangulation process has placed larger triangles in smoother areas and smaller triangles in more detailed areas.

Since subdivision in the shape and in the sphere are performed in parallel, at any stage of the process, parameters at the vertices of the triangular mesh will be known from the parameters of corresponding points in the sphere.



(a) (b) (c)

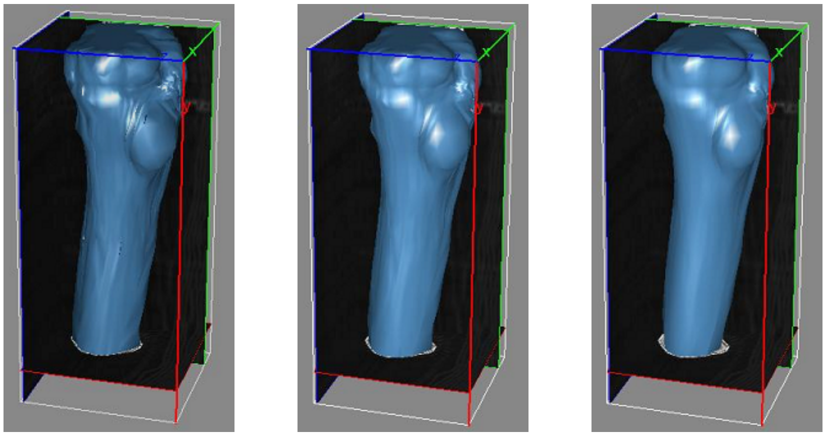
(d) (e) (f)

**Fig. 1.** Approximation of a digital shape by a triangular mesh.

Using Algorithm 2 with the same number of control points as the number of vertices obtained in the mesh approximation, we obtain Figs. 2a–2c when $s_0$ is set to 0.2, 0.5, and 1, respectively.

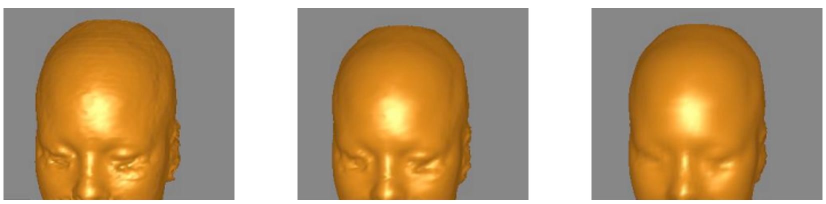
A second set of examples is shown in Figs. 3a–3c. The data set used in this experiment was obtained by segmenting a Magnetic Resonance (MR) image of a person's head. The segmentation has extracted the skin of the head. Using Algorithm 2 with error tolerance equal to 0.5, we obtain the surface shown in Fig. 3a with $s_0 = 0.2$. Increasing $s_0$ to 0.5 we obtain the surface shown in Fig. 3b. Increasing $s_0$ further to 1, we obtain the surface shown in Fig. 3c.

Figs. 2 and 3 show two examples of the proposed surface-fitting method. The number of control points obtained in an approximation is not a mere function of the error tolerance; it is also a function of the smoothness of the required surface. Preliminary results show that to achieve a high compression rate, when a large error tolerance is given a large $s_0$ should be used, and when a small error tolerance is given a small $s_0$ should be used.



(a) (b) (c)

**Fig. 2.** Approximating a digital femoral stem by RaG surfaces.



(a) (b) (c)

**Fig. 3.** Approximating a digital head by RaG surfaces.

## §7. Concluding Remarks

An algorithm to subdivide a digital shape into a triangular mesh, parametrize the mesh vertices as well as the shape points, and fit a rational Gaussian surface to the points was presented. Attempts to parametrize mesh vertices have been made before. Lee *et al.* [6] simplified a mesh to a base mesh, assigned parameters to the vertices of the base mesh, and determined parameters at the original mesh vertices through conformal mapping of the base mesh to the original mesh. Rogers and Fog [7] developed a nonlinear optimization method for determining the parameters of a mesh to be approximated by B-spline patches. Brechbühler *et al.* [1] developed an optimization method for mapping vertices of a simple polyhedron into a sphere and thereby parametrizing the polyhedral vertices.

Once the shape points or the mesh vertices are parametrized, a single RaG surface can be fitted to the points to reconstruct the shape. A RaG surface enables editing of a shape by moving its control points just like a NURBS surface. This representation is especially useful when a noisy data set is given and there is a need to smooth noise in the data. The RaG formulation has a smoothness parameter that can be varied to obtain surfaces at different levels of detail.

**Acknowledgments.** The author would like to thank the National Science Foundation for partially funding this work under grant IIS-9906340. The assistance of Marcel Jakowski in preparation of images presented in this paper is also greatly appreciated. Images used in this work were provided by the Kettering Medical Center, Kettering, OH.

## References

1. Ch. Brechbühler, G. Gerig, and O. Kübler, Parametrization of closed surfaces for 3-D shape description, Computer Vision and Image Understanding **61**:2 (1995), 154–170.
2. J. M. Galvez and M. Canton, Normalization and shape recognition of three-dimensional objects by 3-D moments, Pattern Recognition **26**:5 (1993), 667–682.
3. A. Goshtasby, Design and recovery of 2-D and 3-D shapes using rational Gaussian curves and surfaces, Int'l J. Computer Vision **10**:3 (1993), 233–256.
4. A. Goshtasby, Geometric modeling using rational Gaussian curves and surfaces, Computer-Aided Design **27**:5 (1995), 363–375.
5. J. Hoschek and D. Lasser, *Computer Aided Geometric Design,* A. K. Peters, 1989.
6. A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, MAPS: Multiresolution adaptive parametrization of surfaces, Computer Graphics Proceedings (1998), 95–104.
7. D. F. Rogers and N. G. Fog, Constrained B-spline curve and surface fitting, Comput. Aided Geom. Design **21** (1989), 641–648.

A. Ardeshir Goshtasby
Computer Science and Engineering Department
Wright State University
Dayton, OH 45435
`ardy@cs.wright.edu`
`http://www.cs.wright.edu/~agoshtas/`