Surface Approximation to Scattered Lines

A. Ardeshir Goshtasby

Wright State University, ardy@wright.edu

ABSTRACT

A global weighted-mean approximation to scattered lines is introduced. Rational Gaussian weights are stretched along lines and standard deviations of Gaussians are selected in such a way as to produce surfaces that closely follow data along lines. When the lengths of the lines approach zero, the proposed formulation converges to that of global weighted mean approximation to scattered points. Therefore, the traditional weighted mean approximation becomes a special case of the proposed approximation. Implementation strategies are presented and examples of surface approximation using range data and polygon meshes are given.

Keywords: approximation, global weighted mean, scattered line data, edge-preserving approximation, rational Gaussian weights

1. INTRODUCTION

Surfaces are most widely defined by weighted sum of points [4–6], [10], [13], [17], [24]. Surface formulations based on weighted sum of polynomials also exist in the literature [2] [10]. Surfaces that are defined in terms of points can be considered a weighted sum of polynomials by letting the degree of the polynomials be zero. In parametric surfaces, a component of a surface is defined by a weighted sum of a component of the points, which again can be considered polynomials of degree zero in the parameter space.

In this paper, a surface formulation based on a weighted sum of lines is introduced. The motivation behind this formulation is to produce surfaces that contain sharp edges. The proposed formulation produces surfaces that have high degrees of continuity even at sharp edges.

Man-made scenes often contain flat patches and sharp edges. To represent such geometries from range scans and data generated by computer vision techniques, a representation is needed that can not only represent smoothly varying surfaces but can also represent surfaces containing flat patches and sharp edges. Current surface formulations are not designed to create sharp edges. Subdivision methods have been used to create sharp edges in surfaces by changing the subdivision rules [3], [12]. Subdivision surfaces, however, require that a polygon mesh be given, which is possible to have when working with data generated by computer graphics applications but is rarely obtained when working with data generated by computer vision applications. Stereo depth recovery algorithms can determine depth along scene edges with a rather high degree of accuracy, but they often fail to produce accurate depth values in smooth areas in a scene [1]. Recovered depth will often be along scattered lines in an image and from such scattered line data there is a need to recover the geometry of the scene.

Forcing a surface that is defined by a weighted sum of points to produce a sharp edge in one area may produce high fluctuations in another area. The formulation proposed here is particularly designed to produce sharp edges in a surface. The formulation is written by a weighted sum of lines, which enables the surface to make sharp turns near the lines while approximating them. Providing a smoothness parameter that can be varied interactively, the surface can be made to get as close as desired to the lines and reproduce them, or smooth out the line details and create a smooth surface.

In the following sections, first, the problem to be solved is described. Then, a solution to the problem when the scene represents a single-valued surface is described. Next, the method is extended to parametric surfaces that approximate scattered lines in 3-D. Finally, applications of the proposed formulation in smooth approximation of range data and polygon meshes are discussed.

2. PROBLEM DESCRIPTION

We first solve the problem for scattered line data in 2-D and then extend the method to scattered lines in 3-D. Consider scattered lines in the plane, $\{L_i(x,y) : i = 1,\ldots,N\}$, with endpoints: $\{(x_{i_1}, y_{i_1}),(x_{i_2}, y_{i_2}) : i = 1,\ldots,N\}$. Also, suppose data along a line varies linearly and $L_i(x_{i_1}, y_{i_1}) = Z_{i_1}$ and $L_i(x_{i_2}, y_{i_2}) = Z_{i_2}$ for $i = 1,\ldots,N$. We would like to find a single-valued function $f(x,y)$ that satisfies

$$f(x,y) \approx L_i(x,y), \qquad i = 1,\ldots,N, \tag{1}$$

for points on the lines.

3. APPROACH

The single-valued function approximating a set of scattered line data is obtained by extending a surface formulation that approximates a set of scattered point data. Consider fitting a single-valued surface to scattered point data in the plane, $\{(x_i,y_i,Z_i): i=1,\ldots,N\}$. An example of scattered point data is given in Fig. 1a. Intensities of the points represent the data values at the points. If we use the single-valued rational Gaussian (RaG) surface [7], [8], [15] as the approximating surface, we have

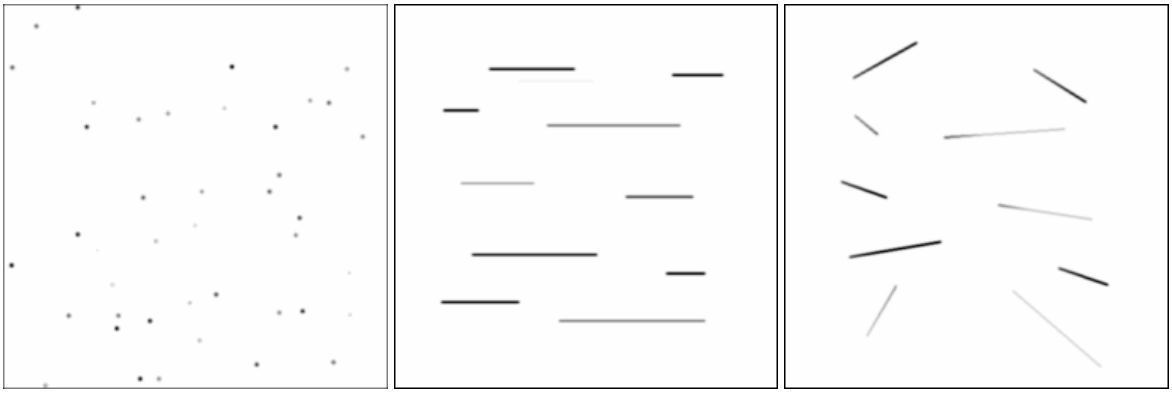
$$f(x,y) = \sum_{i=1}^{N} Z_i g_i(x,y)\,, \tag{2}$$

where $g_i(x,y)$ is the ith blending function of the surface defined by

$$g_i(x,y) = \frac{W_i G_i(x,y)}{\sum_{j=1}^{N} W_j G_j(x,y)}\,. \tag{3}$$

$G_i(x,y)$ is a 2-D Gaussian centered at data point (x_i,y_i) and W_i is the weight or the importance of that point with respect to other points on the surface. If the data value at (x_i,y_i) is Z_i, surface $f(x,y)$ will approximate Z_i at (x_i,y_i). The standard deviations of the Gaussians are the free parameters of the surface and can be varied to produce a desired level of local detail in the surface. The standard deviations of all Gaussians can be set to the same parameter σ to control the smoothness of the surface globally.

Now, consider using a data line in the place of a data point. For the sake of simplicity, let's first assume that data along a line does not vary and all lines are parallel to the x-axis. An example of such data lines is given in Fig. 1b. Therefore, instead of point (x_i,y_i) we will have a line with end points (x_{i_1}, y_i) and (x_{i_2}, y_i) and the same data value Z_i everywhere along the line. To fit a surface to these lines, we will stretch the Gaussian associated with a line proportional to the length of the line horizontally.



(a) (b) (c)

Fig. 1: (a) An example of scattered point data in the plane. (b) Scattered line data parallel to the x-axis. Data value along a line does not vary. (c) Scattered line data in the plane. Data value along a line varies linearly.

Assuming the coordinates of the midpoint of the ith line are (x_i,y_i), since a 2-D Gaussian can be decomposed into two 1-D Gaussians, we have

$$G_i(x,y) = exp\left\{-\frac{(x-x_i)^2+(y-y_i)^2}{2\sigma^2}\right\},$$

$$= exp\left\{-\frac{(x-x_i)^2}{2\sigma^2}\right\} exp\left\{-\frac{(y-y_i)^2}{2\sigma^2}\right\},$$

$$= G_i(x)G_i(y). \quad (4)$$

To stretch $G_i(x,y)$ along the *x*-axis, we can scale down the *x* coordinates, or equivalently, scale up σ. Let's denote this scaling by $m_i > 1$. Then, we replace $G_i(x)$ with

$$H_i(x) = exp\left\{-\frac{(x-x_i)^2}{2(m_i\sigma)^2}\right\}, \quad (5)$$

where $m_i = (1 + \varepsilon_i)$ and ε_i is proportional to the length of the *i*th line. After this stretching, relation (2) becomes

$$f(x,y) = \frac{\sum_{i=1}^{N} W_i Z_i H_i(x) G_i(y)}{\sum_{i=1}^{N} W_i H_i(x) G_i(y)}. \quad (6)$$

Now, suppose data along a line varies linearly, but the lines are still parallel to the *x*-axis. To fit a surface to such data lines, instead of using a Gaussian of a fixed height Z_i, we let the height of the Gaussian vary with data along the line. Assuming data at the endpoints of the *i*th line are Z_{i_1} and Z_{i_2}, and assuming Z_i represents the data value at the midpoint of the line, in equation (6) we will replace Z_i with

$$Z_i(x) = Z_i + \frac{(x-x_i)}{(x_{i_2}-x_i)}(Z_{i_2}-Z_i). \quad (7)$$

This formula changes the height of the Gaussian along a line proportional to the data value there. The new approximation formula, therefore, becomes

$$f(x,y) = \frac{\sum_{i=1}^{N} Z_i(x) W_i H_i(x) G_i(y)}{\sum_{i=1}^{N} W_i H_i(x) G_i(y)}. \quad (8)$$

To adapt the approximation to data lines with arbitrary orientations, such as those shown in Fig. 1c, we rotate each line about its center so it becomes parallel to the *x*-axis, use the above formula to find its contribution to the surface, and finally, rotate the computed values back to their original positions. Doing this for each line and adding the contributions from the lines, we obtain the approximating surface. Assuming the *i*th line makes angle θ_i with the *x*-axis, when rotating the coordinate system clockwise about the line's midpoint by θ_i so that it becomes parallel to the *x*-axis, if the coordinates of points on the line before and after this rotation are *(X,Y)* and *(x,y)*, we will have

$$x = (X-X_i)\cos\theta_i - (Y-Y_i)\sin\theta_i + x_i, \quad (9)$$

$$y = (X-X_i)\sin\theta_i + (Y-Y_i)\cos\theta_i + y_i. \quad (10)$$

Substituting relations (9) and (10) into the right side of equation (8), we obtain a relation in *(X,Y)*. This relation defines the surface value at *(X,Y)* in the approximation domain. Renaming the approximating function by $F(X,Y)$, we will have

$$F(X,Y) = \frac{\sum_{i=1}^{N} W_i Z_i(X,Y) H_i(X,Y) G_i(X,Y)}{\sum_{i=1}^{N} W_i H_i(X,Y) G_i(X,Y)} \quad (11)$$

where

$$Z_i(X,Y) = Z_i + \frac{[(X-X_i)\cos\theta_i - (Y-Y_i)\sin\theta_i]}{D_i}(Z_{i_2}-Z_i), \quad (12)$$

$$H_i(X,Y) = exp\left\{-\frac{[(X-X_i)\cos\theta_i - (Y-Y_i)\sin\theta_i]^2}{2(m\sigma)^2}\right\}, \quad (13)$$

and

$$G_i(X,Y) = exp\left\{-\frac{[(X-X_i)\sin\theta_i + (Y-Y_i)\cos\theta_i]^2}{2\sigma^2}\right\}. \quad (14)$$

Quantity

$$D_i = \sqrt{(x_{i_2}-x_i)^2 + (y_{i_2}-y_i)^2} = \sqrt{(X_{i_2}-X_i)^2 + (Y_{i_2}-Y_i)^2} \quad (15)$$

is half the length of the *i*th line in the xy or *XY* domain. Weight W_i of line L_i is set equal to $1+2D_i$. The 1 in the formula ensures that if instead of lines points are provided, the obtained surface will approximate the points. As the length of a line increases, the volume under the stretched Gaussian increases also. To make the weight function dependent on the length of the line as well as the data values on the line, the following formula is recommended:

$$W_i = 1 + 2D_i = 1 + \sqrt{(x_{i_2}-x_{i_1})^2 + (y_{i_2}-y_{i_1})^2 + (Z_{i_2}-Z_{i_1})^2}\ . \quad (16)$$

Substituting (12–14) and (16) into (11) a single-valued surface will be obtained approximating scattered line data in the plane.

4. APPROXIMATION TO SCATTERED LINES IN 3-D

A simple example demonstrating the proposed idea is given in Fig. 2. Fig. 2a shows seven data lines in the *xy* plane. Intensities of points along a line represent the data values there. The coordinates of the line endpoints and the associating data values are shown in Table 1. Fig. 2b shows the surface approximating the lines according to formula (11). Although the surface approximates the lines, flat spots are obtained along the lines. This is a known property of weighted mean methods. Since the sum of the weights is required to be 1 everywhere in the approximation domain, when the weight functions have rather small widths, flat spots are obtained at and near the data points, and so in our case, at and near the lines. To avoid such flat spots from appearing in the approximating surface, instead of single-valued surfaces, we suggest use of parametric surfaces. Therefore, instead of the single-valued surface given by (11), we use the parametric surface defined by

$$F_x(u,v) = \frac{\sum_{i=1}^{N} W_i X_i(u,v) H_i(u,v) G_i(u,v)}{\sum_{i=1}^{N} W_i H_i(u,v) G_i(u,v)}\ , \quad (17)$$

$$F_y(u,v) = \frac{\sum_{i=1}^{N} W_i Y_i(u,v) H_i(u,v) G_i(u,v)}{\sum_{i=1}^{N} W_i H_i(u,v) G_i(u,v)}\ , \quad (18)$$

$$F_z(u,v) = \frac{\sum_{i=1}^{N} W_i Z_i(u,v) H_i(u,v) G_i(u,v)}{\sum_{i=1}^{N} W_i H_i(u,v) G_i(u,v)}\ . \quad (19)$$

Doing so, we obtain the surface shown in Fig. 2c. F_x, F_y, and F_z are the *X*, *Y*, and *Z* components of the surface, each obtained by varying *u* and *v* from 0 to 1. Parameter coordinates at the line midpoints and line end points are set proportional to the *XY* coordinates of the line midpoints and end points, respectively. That is,

$$u_i = (X_i - X_{min})/(X_{max} - X_{min})\ , \quad (20)$$

$$u_{i1} = (X_{i1} - X_{min})/(X_{max} - X_{min})\ , \quad (21)$$

$$u_{i2} = (X_{i2} - X_{min})/(X_{max} - X_{min})\ , \quad (22)$$

$$v_i = (Y_i - Y_{min})/(Y_{max} - Y_{min})\ , \quad (23)$$

$$v_{i1} = (Y_{i1} - Y_{min})/(Y_{max} - Y_{min})\ , \quad (24)$$

$$v_{i2} = (Y_{i2} - Y_{min})/(Y_{max} - Y_{min})\ , \quad (25)$$

where X_{min}, X_{max}, Y_{min}, and Y_{max} define the approximation domain. The parametric representation makes it possible to fit a surface to scattered lines in 3-D when the lines can be mapped to a plane, a cylinder, a torus, or a sphere.

Formulas (17–19) describe a parametric surface approximating a set of lines and Fig. 2c shows the surface approximating the lines of Fig. 2a using this formulation. Increasing the standard deviation of the Gaussians, which we will call the smoothness parameter, will produce a smoother surface as shown in Fig. 2d, while decreasing the

smoothness parameter will produce a surface that follows the lines more closely as shown in Fig. 2e. Fig. 2f depicts a view of this surface from the opposite side. Portions of the lines hidden from the front view are visible in the rear view.

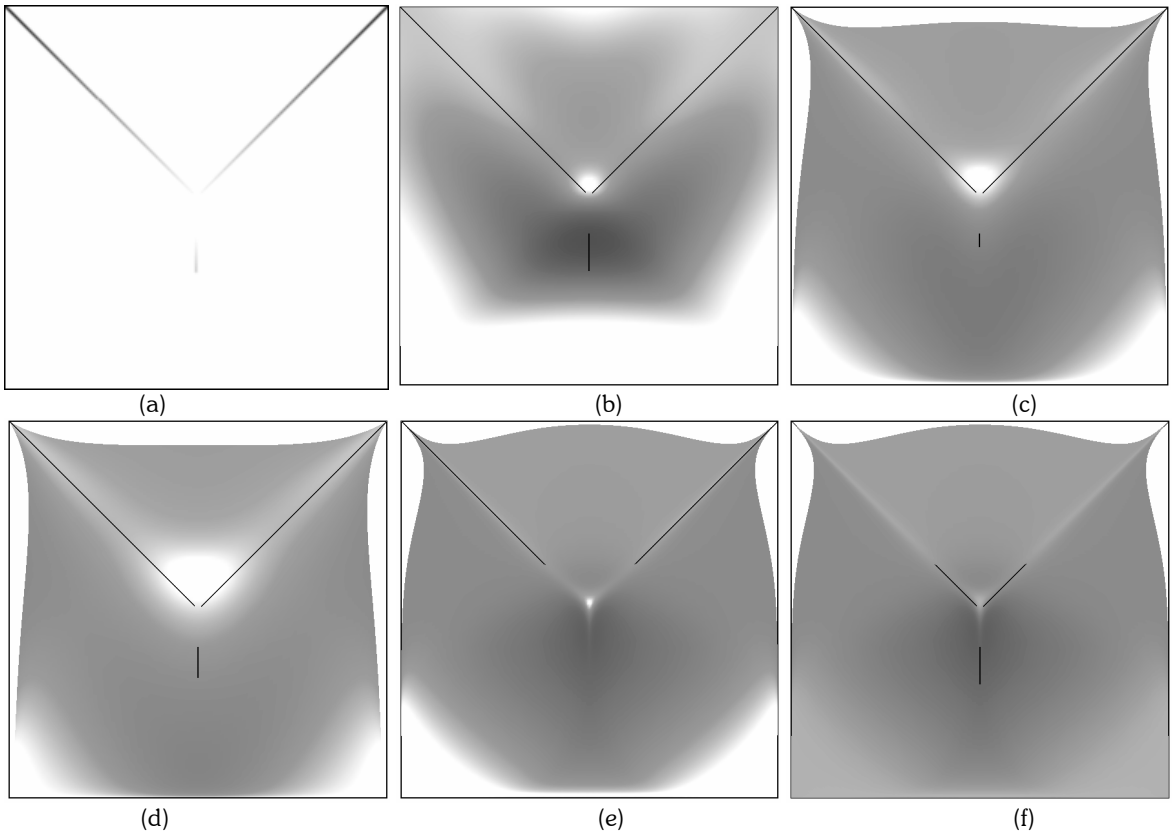


Fig. 2: (a) Data lines of Tab. 1. Higher values are shown with brighter intensities. (b) Single-valued surface defined by (11) approximating the data lines. (c) Parametric surface defined by (17)–(19) approximating the same data lines. (d) Same as (c) but using a larger σ. (e) Same as (c) but using a smaller σ. (f) Same as (e) but viewing from the opposite side.

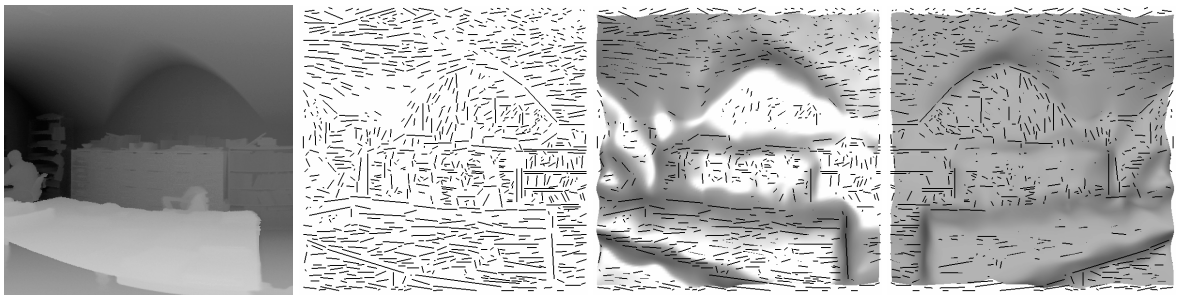
i	1	2	3	4	5	6	7
X_{i_1}	–50	50	50	–50	1	–1	0
Y_{i_1}	–50	–50	50	50	1	1	–10
Z_{i_1}	0	0	0	0	50	50	40
X_{i_2}	50	50	–50	–50	50	–50	0
Y_{i_2}	–50	50	50	–50	50	50	–20
Z_{i_2}	0	0	0	0	0	0	20

Tab. 1: Data lines of Fig. 2a.

A more complex example of surface fitting by the proposed method is given in Fig. 3. Fig. 3a shows a depth map of an indoor scene. Closer points are shown brighter. Linear features in the image are detected by computer vision techniques as shown in Fig. 3b and used as the lines. The lines are then used in the parametric formulation to recover the scene geometry by letting the parameter coordinates at the line end points be proportional to their image coordinates. The geometry recovered by the surface defined by (17)–(19) is shown in Fig. 3c. The lines are overlaid

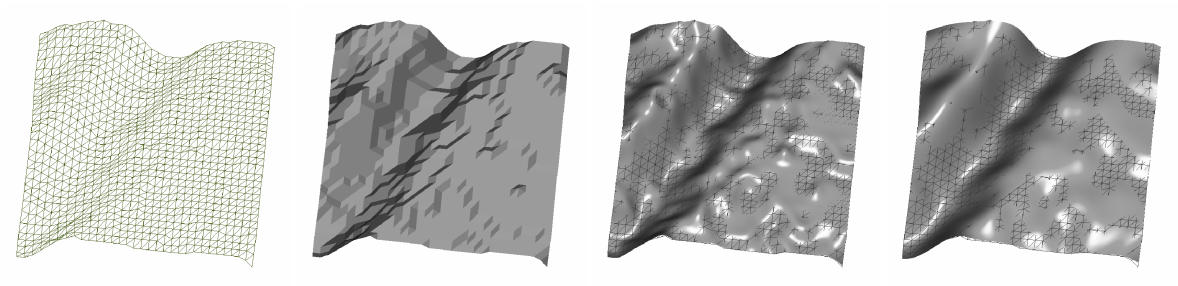
with the recovered surface to show the quality of approximation. Parts of the lines in front of the surface are visible in this image. The remaining lines are behind the surface and can be viewed from the opposite side as shown in Fig. 3d.

An example of the proposed method in approximation of triangle meshes is given in Fig. 4. Fig. 4a shows a mesh obtained by triangulating a digital elevation map of an area over the Grand Canyon. The triangle mesh in shaded form is shown in Fig. 4b. Using the edges of the triangles as input to the proposed approximation, and letting *uv* be proportional to the *xy* coordinates of the mesh points, the surface shown in Fig. 4c is obtained. The triangle mesh is overlaid with the surface to show the quality of approximation. Increasing the smoothness parameter, the surface shown in Fig. 4d is obtained, smoothing the edges. This example demonstrates that the proposed method can be used to smooth a triangle mesh, creating a surface with a very high degree of continuity.



(a) (b) (c) (d)

Fig. 3: (a) A depth map of an indoor scene. This data set is courtesy of University of California, Los Angeles. (b) Lines approximating the edges in the depth map. (c) Estimation of the scene geometry using the data lines. (d) The same geometry when viewed from the opposite side. Some of the lines or their parts that are hidden in (c) are visible in (d).



(a) (b) (c) (d)

Fig. 4: (a) Triangulation of a digital elevation map. This data set is courtesy of USGS. (b) The triangle mesh in shaded form. (c) Approximation of the mesh by the proposed approximation using the mesh edges as the 3-D lines. (d) A smoother approximation to the same mesh.

Since the proposed surface is in parametric form, the lines are not required to represent single-valued data. Rather, they can represent any geometry as long as parameter coordinates at the line end points are known. An example of a cylindrical surface is given in Fig. 5. Table 2 shows the *XYZ* coordinates as well as the *uv* parameters of twelve line end points. These line segments represent the four side edges of a cube. The lines are shown in Fig. 5a and the cylindrical surface obtained from the lines is depicted in Fig. 5b. Reducing the smoothness parameter σ pulls the surface closer to the lines as depicted in Fig. 5c. Another example of a cylindrical surface is given in Figs. 5d–f. Eight more line segments were added to the set of lines in Fig. 5a, connecting the centers of the two open faces to the four corners of the same faces as depicted in Fig. 5d. The *XYZ* and *uv* coordinates at the twenty line end points are given in Table 3. There are two tinyl holes at the two ends of the cylindrical surface, which become invisible when rendered. The approximating surface reproduces the cube closely as depicted in Fig. 5f.

An example of a closed surface is given in Fig. 6. The coordinates of the line endpoints in this data set are given in Table 4. The line segments shown in Fig. 6a have endpoints *(1,2), (2,3), (3,4), (4,1); (5,6), (6,7), (7,8), (8,5); (9,10), (10,11), (11,12), (12,9); (13,14), (14,15), (15,16), (16,13)*. Figs. 6b and 6c show surfaces obtained with different

smoothness parameters. If the four squares shown in Fig. 6a are moved parallel to themselves towards each other so that their centers coincide at *(0,0,0)* without changing their parameter coordinates, a spherical surface will be obtained. Details about parametrizing spherical surfaces with torus parameter coordinates are given elsewhere [9].

i	1	2	3	4	5	6	7	8	9	10	11	12
X_{i1}	0	0	1	1	0	0	1	1	0	0	1	1
Y_{i1}	0	1	1	0	0	1	1	0	0	1	1	0
Z_{i1}	0	0	0	0	1	1	1	1	0	0	0	0
u_{i1}	0	¼	½	¾	0	¼	½	¾	0	¼	½	¾
v_{i1}	0	0	0	0	1	1	1	1	0	0	0	0
X_{i2}	0	1	1	0	0	1	1	0	0	0	1	1
Y_{i2}	1	1	0	0	1	1	0	0	0	1	1	0
Z_{i2}	0	0	0	0	1	1	1	1	1	1	1	1
u_{i2}	¼	½	¾	1	¼	½	¾	1	0	¼	½	¾
v_{i2}	0	0	0	0	1	1	1	1	1	1	1	1

Tab. 2: Coordinates of the line endpoints shown in Fig. 5a.

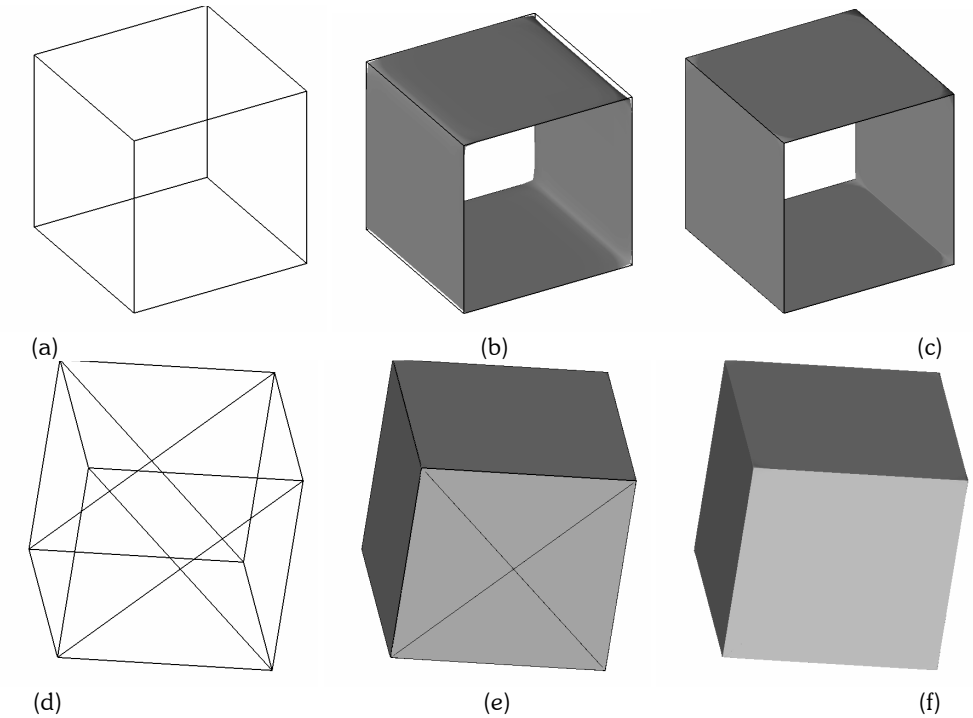


Fig. 5: (a) The line data set of Table 2. (b) The surface approximating the lines. (c) Same as (b) but using a smaller σ. (d) The line data set of Table 3. (e) The surface approximating the line data set of (d). (f) Same as (e) but without showing the lines.

An example of a triangulated cylindrical surface is given in Fig. 7. Fig. 7a shows triangulation of the range scan of a person's knee. The mesh in shaded form is shown in Fig. 7b. Using the edges of the mesh as input to the proposed weighted linear method and using cylindrical parametrization, the surface shown in Fig. 7c is obtained. Increasing the smoothness parameter reduces the noisy details in the surface and captures the global shape of the knee as depicted in Fig. 7d. The triangle mesh is overlaid with the surface to demonstrate the quality of approximation. This example demonstrates reconstruction of a smooth surface, containing no sharp edges. The lines used in the input simply represent small linear segments approximating a smooth surface.

To approximate a surface to single-valued linear data, parameters u and v of the line endpoints can be set proportional to the x and y coordinates of the line end points. For more complex geometries, determination of parameters at line end points may become a challenge. Algorithms to parametrize geometries that can be mapped to a plane, cylinder, sphere, or torus have been developed [14], [19]. The weakness of the proposed surface formulation compared to the subdivision methods [16], [20], [23] can be considered the need for finding the parameter coordinates at the line endpoints. The proposed method, however, has advantages over subdivision methods when parameter coordinates at line endpoints are known. For instance, to create a surface, a complete triangle mesh is not needed. Parts of some of the triangle edges can be missing and the process will still work. In addition, the proposed method has a smoothness parameter that can be changed to suppress noisy details in the constructed surface. This function, if not impossible, is very difficult to achieve by subdivision methods.

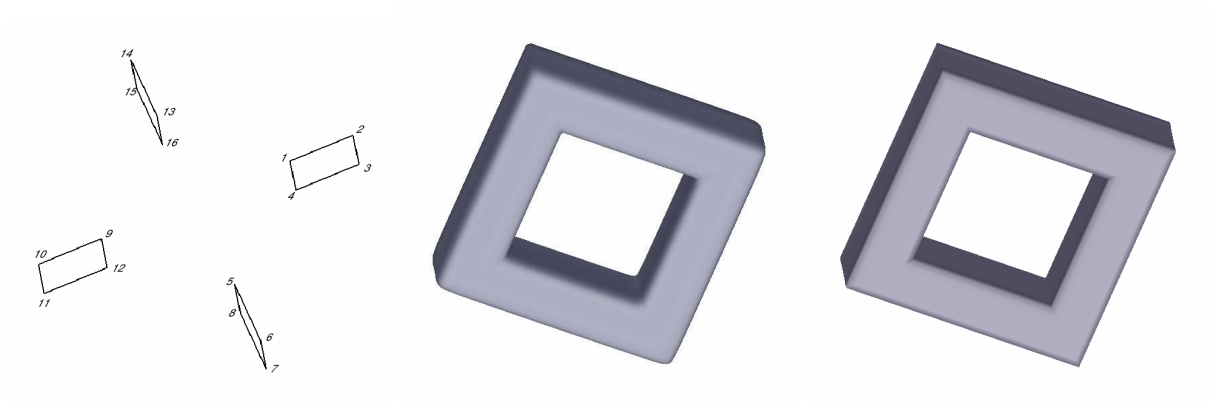
Related to the proposed method are the discontinuous regression method of Qiu [20] and the moving least-squares method of Shen et al. [22]. These methods, which work quite well when dense range data are available, either fail or perform poorly when data lie along scattered lines.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
X_{i1}	0	0	1	1	0	0	1	1	0	0	1	1	1/2	½	½	½	½	½	½	1/2
Y_{i1}	0	1	1	0	0	1	1	0	0	1	1	0	½	½	½	½	½	½	½	1/2
Z_{i1}	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
u_{i1}	0	¼	1/2	¾	0	¼	½	¾	0	¼	½	¾	0	¼	½	¾	0	¼	½	¾
v_{i1}	1/3	1/3	1/3	1/3	2/3	2/3	2/3	2/3	1/3	1/3	1/3	1/3	0	0	0	0	1	1	1	1
X_{i2}	0	1	1	0	0	1	1	0	0	0	1	1	0	0	1	1	0	0	1	1
Y_{i2}	1	1	0	0	1	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0
Z_{i2}	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
u_{i2}	¼	½	¾	1	¼	½	¾	1	0	¼	½	¾	0	¼	½	¾	0	¼	½	¾
v_{i2}	1/3	1/3	1/3	1/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	1/3	1/3	1/3	1/3	2/3	2/3	2/3	2/3

Tab. 3: The coordinates of line endpoints in Figs. 5d.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
X_i	4.5	5.5	5.5	4.5	0	0	0	0	-4.5	-5.5	-5.5	-4.5	0	0	0	0
Y_i	0	0	0	0	-4.5	-5.5	-5.5	-4.5	0	0	0	0	4.5	5.5	5.5	4.5
Z_i	0.5	0.5	-0.5	-0.5	0.5	0.5	-0.5	-0.5	0.5	0.5	-0.5	-0.5	0.5	0.5	-0.5	-0.5
u_i	0	0.25	0.5	0.75	0	0.25	0.5	0.75	0	0.25	0.5	0.75	0	0.25	0.5	0.75
v_i	0	0	0	0	0.25	0.25	0.25	0.25	0.5	0.5	0.5	0.5	0.75	0.75	0.75	0.75

Tab. 4: Coordinates of line end points shown in Fig. 6a.



(a) (b) (c)

Fig. 6: An example of a closed surface. (a) The line segments defining the surface. The coordinates of the line endpoints are shown in Table 4. (b), (c) Surfaces obtained with different values of the smoothness parameter σ.

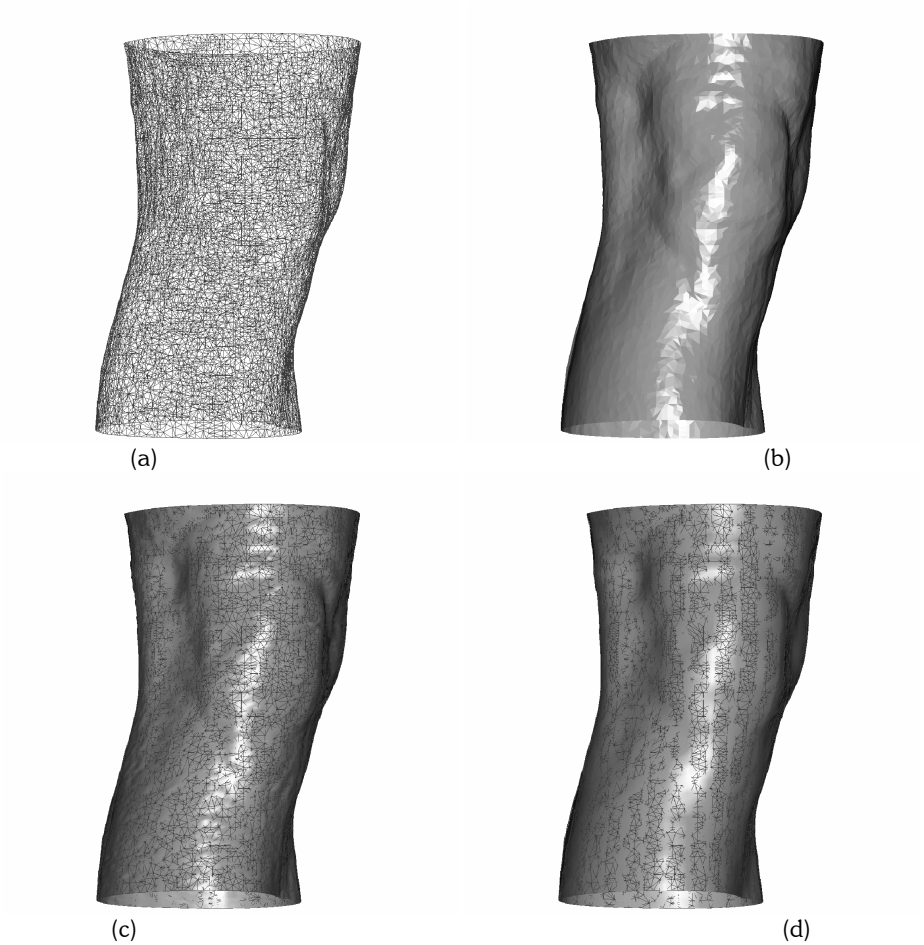


Fig. 7: (a) Triangulation of a range data set of a person's knee. This data set is courtesy of SGI. (b) The triangle mesh in shaded form. (c) Approximation of the mesh by the proposed method. (d) Same as (c) but using a larger smoothness parameter. The triangle mesh is overlaid with the surfaces in (c) and (d) to show the approximation quality.

5. CONCLUSIONS

If a polygon mesh is given, a smooth approximation to the mesh can be obtained by 1) approximating the mesh vertices, 2) approximating the mesh faces, or 3) approximating the mesh edges. This paper described a method for approximating the mesh edges. Methods that are based on points tend to produce overall smooth surfaces. Methods that are based on planes [10] and lines can reproduce flat faces and sharp edges in created surfaces. In the proposed approximation, by sufficiently lowering a smoothness parameter, the surface can be made to closely follow the provided lines and reproduce sharp edges in the surface, while by setting the smoothness parameter sufficiently high, the surface can be made to smooth out noisy line data and produce an overall smooth surface. The method described here does not require that all edges in a polygon mesh be provided or that entire edge segments be given. Rather, a partial edge list is sufficient to create a surface that approximates the edges.

If a line data set can be mapped to a plane, a cylinder, a torus, or a sphere, the lines can be approximated by the proposed surface. The obtained surface will have a high degree of continuity everywhere, enabling very accurate generation of surface normals, and consequently, generation of very high-quality images. The proposed method is particularly suitable in applications where the smoothness of the surface can be interactively varied to reproduce sharp edges in data or smooth out noisy edge details and capture the global geometry hidden amongst the data.

REFERENCES

[1] Bay, H.; Ferrari, V.; Van Gool, L.: Wide-baseline stereo matching with line segments, Proc. Computer Vision and Pattern Recognition, 1, 2005, 329–336.

[2] Catmull, E.; Rom, A.: A class of interpolating splines, in Computer Aided Geometric Design, R. Barnhill and R. Riesenfeld (eds.), Academic press, New York, 1974, 317–326.

[3] DeRose, T.; Kass, M.; Truong, T.: Subdivision surfaces in character animation, Proc. ACM SIGGRAPH, 1998, 85–94.

[4] Farin, G.: Curves and Surfaces for Computer Aided Geometric Design, Academic Press, New York, 1988.

[5] Franke, R.: Scattered data interpolation: test of some methods, Mathematics of Computation, 38(157), 1982, 181–200.

[6] Franke, R.; Schumaker, L. L.: A bibliography of multivariate approximation, in Topics in Multivariate Approximation, Academic Press, New York, 1987, 275–335.

[7] Goshtasby, A.: Design and recovery of 2-D and 3-D shapes using rational Gaussian curves and surfaces, Int'l J. Computer Vision, 1993, 233–256.

[8] Goshtasby, A.: Geometric modeling using rational Gaussian curves and surfaces, Computer-Aided Design, 1995, 363–375.

[9] Goshtasby, A.: Parametric circles and spheres, Computer-Aided Design, 35, 2003, 487–494.

[10] Goshtasby, A.: A weighted mean approach to smooth parametric representation of polygon meshes, The Visual Computer, 20(5), 2004, 344–359.

[11] Grosse, E.: A catalogue of algorithms for approximation, in Algorithms for Approximation II, J. Mason and M. Cox (eds.), Chapman and Hall, 1990, 479–514.

[12] Hoppe, H.; DeRose, T.; Duchamp, T.; Halstead, M.; Jin, H.; McDonald, J.; Schweitzer, J.; Stuetzler, W.: Piecewise smooth surface reconstruction, Computer Graphics, 28(3), 1994, 295–302.

[13] Hoschek, J.; Lasser, D.: Fundamentals of Computer Aided Design, translated by L. L. Schumaker, A. K. Peters, Wellesley, MA, 1993.

[14] Khohodakovsky, A.; Litke, N.; Schröde, P.: Globally smooth parametrization with low distortion, Proc. SIGGRAPH, ACM Press, 2003, 350–357.

[15] Jackowski, M.; Satter, M.; Goshtasby, A.: Approximating digital 3-D shapes by rational Gaussian surfaces, IEEE Trans. Visualization and Computer Graphics, 9(1), 2003, 56–69.

[16] Ma, W.: Subdivision surfaces for CAD – An overview, Computer Aided Design, 37, 2005, 693–709.

[17] Mortenson, M. E.: Geometric Modeling, 2nd Edition, Wiley Computer Publishing, 1997.

[18] Oswald, P.: Designing composite triangular subdivision schemes, Computer Aided Geometric Design, 22, 2005, 659–679.

[19] Praun, E.; Hoppe, H.: Spherical parametrization and remeshing, Proc. SIGGRAPH, ACM Press, 2003, 340–349.

[20] Qiu, P: Discontinuous regression surfaces fitting, The Annals of Statistics, 26(6), 1998, 2218–2245.

[21] Schaefer, S.; Warren, J.: On C^2 triangle/quad subdivision, ACM Trans. Graphics, 24(1), 2005, 28–36.

[22] Shen, C.; O'Brien, J. F.; Shewchuk, J. R.: Interpolating and approximating implicit surfaces from polygon soup, Proc. SIGGRAPH, 2004, 896–904.

[23] Warren, J.; Weimer, H.: Subdivision methods for Geometric Design, Morgan Kaufmann Publishers, San Francisco, 2002.

[24] Weiss, V.; Andor, L.; Renner, G.; Varady, T.: Advanced surface fitting techniques, Computer Aided Geometric Design, 19, 2002, 19–42.