Computer Aided Geometric Design
11 (1994) 477–489

COMPUTER AIDED GEOMETRIC DESIGN

Data point selection for piecewise trilinear approximation

Bernd Hamann [a,b,*] Jiann-Liang Chen [c]

[a] *Department of Computer Science, Mississippi State University, P.O. Drawer CS, Mississippi State, MS 39762, USA*

[b] *NSF Engineering Research Center for Computational Field Simulation, Mississippi State University, P.O. Box 6176, Mississippi State, MS 39762, USA*

[c] *Department of Electrical Engineering, Mississippi State University, P.O. Drawer EE, Mississippi State, MS 39762, USA*

Received November 1992; revised August 1993

Abstract

A technique for the iterative selection of 3D points with associated function values (trivariate data) is presented. The selection algorithm is based on assigning weights to the given data and selecting the most important ones. Weights are assigned using a local least square polynomial approximation to the data based on a triangulation of the 3D points. The absolute curvature of the graph of such a local approximant defines the weight for the associated data point. The boundary (convex hull) of the original point set is preserved by keeping boundary points defining the convex hull. Interior data points are selected according to their weights. The insertion of a selected data point requires a local modification of the triangulation.

Keywords: Approximation; Curvature; Data dependent triangulation; Data reduction

1. Introduction

This paper is concerned with the reduction of trivariate, scalar-valued data given as finite sets, denoted by

$$\mathcal{X}^* = \{\boldsymbol{x}_i^* = (x_i, y_i, z_i, f_i) \mid i = 1, \ldots, n\}. \tag{1}$$

Quite often, such data sets are redundant in the sense that function values f_i might vary (nearly) linearly in certain regions. Based on this observation, a data selection algorithm

* Corresponding author. Email: hamann@erc.msstate.edu.

SSDI 0167-8396(93)E0040-K

is developed for trivariate data sets. The algorithm requires an initial triangulation of the given 3D points. In addition, the polyhedron defined by the boundary of this triangulation must be convex. Obviously, a Delaunay triangulation of the given 3D points satisfies this condition. The original point set is denoted by

$$\mathcal{X} = \{\boldsymbol{x}_i = (x_i, y_i, z_i) \mid i = 1, \ldots, n\}, \tag{2}$$

and its initial triangulation is written as

$$\mathcal{T} = \bigcup \{\mathcal{T}_j = (i_{1,j}, i_{2,j}, i_{3,j}, i_{4,j})\}, \tag{3}$$

which is the union of all tetrahedra identified with their index quadruples referring to their vertices in (2).

Given a triangulation of the original point set, one constructs local least square polynomial approximations to the data by considering data in some neighborhood around a particular vertex. Each original data point $\boldsymbol{x}_i^*$ is weighted according to the absolute curvature of the graph of such a local approximant evaluated at $\boldsymbol{x}_i$. Thus, absolute curvature determines a data point's significance. Once the set of triangles defining the convex boundary polyhedron of the initial triangulation is extracted, interior data points are selected in descending order of their weights. At the end of the procedure, the local density of data points reflects the curvature of the graph of an underlying piecewise polynomial approximant. The density of data points is expected to be maximal in regions having maximal curvature.

Triangulation schemes are discussed in detail in (Barth et al., 1992) and (Lawson 1977, 1986). Data dependent triangulations for bivariate data can be found in (Dyn et al., 1990a,b; Quak and Schumaker, 1991; Rippa 1989). Knot removal strategies for spline curves and tensor product surfaces are described in (Arge et al., 1990; Goldman and Lyche, 1993; Lyche and Mørken 1987, 1988). An iterative knot removal algorithm for bivariate data is given in (Le Méhauté and Lafranche, 1989). A data point selection scheme for curves is described in (Hamann and Chen, 1994), and an iterative method for reducing surface triangulations is given in (Hamann, 1994a). The data point selection algorithm developed in this paper requires the approximation of principal curvatures for graphs of trivariate functions. Curvature approximation for trivariate data is discussed in (Hamann, 1994b). For more general topics regarding curve/surface modeling and differential geometry see (do Carmo, 1976; Farin, 1992; Spivak, 1970).

2. Absolute curvature approximation

In this section, the main concepts of differential geometry required for curvature approximation of trivariate data are briefly summarized (see (Hamann, 1994b)). The graph of a trivariate function $f(x, y, z)$, f in class C^m, $m \geqslant 2$, mapping an open set $\mathcal{S} \subset \mathbb{R}^3$ into $\mathbb{R}$ can be viewed as a regular parametric "surface" $\mathcal{X}^* = \{\boldsymbol{s}(\boldsymbol{x}) = (x_1, x_2, x_3, x_4)\} \subset \mathbb{R}^4$ with the parametrization

$$\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{s}(x, y, z) = (x, y, z, f(x, y, z)). \tag{4}$$

The partial derivatives of this surface are

$$\frac{\partial}{\partial x}\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{s}_x = (1,0,0,f_x),$$
$$\frac{\partial}{\partial y}\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{s}_y = (0,1,0,f_y), \quad \text{and} \tag{5}$$
$$\frac{\partial}{\partial z}\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{s}_z = (0,0,1,f_z).$$

The tangent space at the point $\boldsymbol{s}_0 = \boldsymbol{s}(\boldsymbol{x}_0)$ is defined as the set of all points $\boldsymbol{y} \in \mathbb{R}^4$ satisfying

$$\boldsymbol{y} = \boldsymbol{s}_0 + \alpha \boldsymbol{s}_x + \beta \boldsymbol{s}_y + \gamma \boldsymbol{s}_z|_{\boldsymbol{x}_0}, \quad \alpha,\beta,\gamma \in \mathbb{R}, \tag{6}$$

and the unit outward normal vector at $\boldsymbol{s}_0$ is

$$\boldsymbol{n}_0 = \boldsymbol{n}(\boldsymbol{x}_0) = \left.\frac{(-f_x,-f_y,-f_z,1)}{\sqrt{1+f_x{}^2+f_y{}^2+f_z{}^2}}\right|_{\boldsymbol{x}_0}. \tag{7}$$

The Gauss–Weingarten map for this particular surface is defined by the matrix $-A$, given as

$$-A = -\langle a_{i,j}\rangle = \frac{1}{l}\begin{pmatrix} f_{xx} & f_{xy} & f_{xz} \\ f_{xy} & f_{yy} & f_{yz} \\ f_{xz} & f_{yz} & f_{zz}\end{pmatrix}\begin{pmatrix} 1+f_x{}^2 & f_xf_y & f_xf_z \\ f_xf_y & 1+f_y{}^2 & f_yf_z \\ f_xf_z & f_yf_z & 1+f_z{}^2\end{pmatrix}^{-1}, \tag{8}$$

where $l = \sqrt{1+f_x{}^2+f_y{}^2+f_z{}^2}$.

It is a well-known fact in differential geometry that the three (real) eigenvalues κ_1, κ_2, and κ_3 of $-A$ are the principal curvatures of the graph of $f(x,y,z)$. Therefore, the principal curvatures are the (real) roots of the characteristic polynomial of $-A$, the cubic polynomial

$$\begin{aligned}\det(-A-\kappa I) = {} & \kappa^3 + (a_{1,1}+a_{2,2}+a_{3,3})\kappa^2 \\ & + (a_{1,1}a_{2,2}+a_{1,1}a_{3,3}+a_{2,2}a_{3,3}-a_{1,2}a_{2,1}-a_{1,3}a_{3,1}-a_{2,3}a_{3,2})\kappa \\ & + a_{1,1}a_{2,2}a_{3,3}+a_{1,2}a_{2,3}a_{3,1}+a_{1,3}a_{2,1}a_{3,2} \\ & - a_{1,1}a_{2,3}a_{3,2}-a_{1,2}a_{2,1}a_{3,3}-a_{1,3}a_{2,2}a_{3,1}.\end{aligned} \tag{9}$$

The sum K of the absolute values of the principal curvatures is the absolute curvature,

$$K = |\kappa_1| + |\kappa_2| + |\kappa_3|. \tag{10}$$

See (do Carmo, 1976) and (Spivak, 1970) for more detail.

The algorithmic aspects of approximating the absolute curvature K for each point in $\mathcal{X}^*$ are described in (Hamann, 1994b). In summary, the curvature approximation scheme is based on constructing a triangulation of all given 3D points, computing a local least square approximation for each data point, deriving the first and second partial derivatives of this approximation (required for the Gauss–Weingarten map), and computing the absolute curvature K. Once absolute curvature estimates are known for each data point, the points are sorted in descending order of their curvatures.

3. Data point selection

There are two alternatives for terminating the data point selection algorithm. Either a specified number/percentage of all data is specified, or an error measure is used to determine the number of data points required to approximate the original data within the allowed tolerance. Original data points lying on the boundary of the domain of the discretized function $f(x, y, z)$ require a different treatment than data lying in the interior of the domain. Here, the term boundary is used as a synonym for convex hull. In order to preserve the convex hull of the original point set, the points defining the convex hull must be preserved by the data point selection algorithm. These points can be extracted directly from a Delaunay triangulation of all original 3D points.

The data selection algorithm assumes that the boundary of the initial triangulation of all 3D points is a convex polyhedron. Obviously, the points on the convex hull of a Delaunay triangulation of the initial point set imply such a convex polyhedron. In order to preserve the boundary of the given point set, a Delaunay triangulation of all points is constructed first (see (Preparata and Shamos, 1990)). The boundary of this triangulation is a convex polyhedron whose faces are divided into triangles. These triangles are extracted from the Delaunay triangulation and kept for the insertion process of selected interior points. The triangles defining the convex boundary polyhedron are not necessarily preserved when inserting selected points into the triangulation at a later time.

Extracting the boundary triangles in this way might lead to redundancy. Redundancy occurs when an n-sided face of the boundary polyhedron is divided into more than $(n-2)$ triangles. In other words, if there are original points lying in the interior of a face or the interior of an edge of the boundary polyhedron, the number of triangles extracted directly from the Delaunay triangulation is not minimal. Unfortunately, it is extremely difficult to detect such collinearities/coplanarities due to numerical problems. Redundant boundary points can be removed without any problem if a 3D point set's convex hull is a cuboid (rectilinear data). The examples given in Section 5 take advantage of this fact.

Once the points/triangles describing the boundary polyhedron are extracted, the interior data point with maximal absolute curvature is determined. This interior point is used in combination with the boundary triangles to construct a first triangulation of the domain. The initial set of tetrahedra is defined by connecting the selected interior point with each point on the boundary polyhedron. This triangulation changes whenever another data point is selected and inserted. Definition 1 is needed for the extraction of the set of triangles defining the convex boundary polyhedron.

Definition 1. Given a Delaunay triangulation of a set of 3D points, a face belonging to exactly one tetrahedron is called *boundary triangle*; its edges and vertices are called *boundary edges* and *boundary vertices*.

The set of boundary triangles is extracted from the initial Delaunay triangulation of all given 3D points, and the boundary vertices are kept. The set of preserved boundary data points is denoted by

$$\mathcal{B}^* = \{\boldsymbol{x}^*_{i_j} = (x_{i_j}, y_{i_j}, z_{i_j}, f_{i_j}) \mid j = 1, \ldots, n_B,\ i_j \in \{1, \ldots, n\}\}. \tag{11}$$

The set of interior data points is denoted by

$$\mathcal{I}^* = \mathcal{X}^* \setminus \mathcal{B}^* = \{\boldsymbol{x}^*_{i_k} = (x_{i_k}, y_{i_k}, z_{i_k}, f_{i_k}) \mid k = 1, \ldots, n_I,\ i_k \in \{1, \ldots, n\}\}, \tag{12}$$

Absolute curvature estimates are computed for interior data points only, since all boundary data points are preserved by construction. The fact that the boundary triangles form a convex polyhedron simplifies the iterative insertion of interior points. Knowing absolute curvatures for all elements in $\mathcal{I}^*$, the interior data points are sorted in descending order of these values and inserted according to this order. As already mentioned above, the interior data point with maximal absolute curvature K is inserted first.

4. Data dependent triangulation

The initial data dependent triangulation consisting of all boundary vertices and the first selected interior data point is constructed by connecting the interior point with each boundary vertex. Additional interior data points are inserted until a certain number/percentage of them has been reached or some error criterion is satisfied. When a point is inserted into the triangulation, local modifications (swapping edges) are performed in order to minimize a local approximation error. This process leads to a data dependent triangulation.

The construction of a data dependent triangulation requires an error measure for each tetrahedron refining an intermediate triangulation.

Definition 2. Given a tetrahedron with vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$, and $\boldsymbol{v}_4$ with associated function values f_1, f_2, f_3, and f_4 and a set of points $\boldsymbol{x}_i$, $i = 1, \ldots, N$, with associated function values F_i, $i = 1, \ldots, N$, such that each $\boldsymbol{x}_i$ can be written as a convex combination

$$\boldsymbol{x}_i = \sum_{j=1}^{4} u_{i,j} \boldsymbol{v}_j, \tag{13}$$

where $\sum_{j=1}^{4} u_{i,j} = 1$ and $u_{i,j} \geqslant 0$, $j = 1, \ldots, 4$, the *tetrahedral root-mean-square* (*RMS*) *error* is given by

$$e_{\mathrm{RMS}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (F_i - L(\boldsymbol{u}_i))^2}, \tag{14}$$

and the *tetrahedral maximum error* is given by

$$e_\infty = \max_{1 \leqslant i \leqslant N} |F_i - L(\boldsymbol{u}_i)|, \tag{15}$$

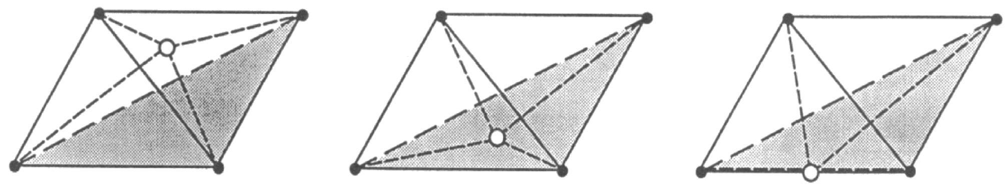

Fig. 1. Inserting point into triangulation.

where $L(\boldsymbol{u}_i) = L(u_{i,1}, u_{i,2}, u_{i,3}, u_{i,4})$ is the linear polynomial

$$L(\boldsymbol{u}_i) = \sum_{j=1}^{4} u_{i,j} f_j. \tag{16}$$

The data dependent triangulation algorithm can use either error measure. Depending on the particular application, one of the two measures might be more appropriate than the other one. In general, when an interior point is inserted into the triangulation, three cases must be considered:

(i) The point to be inserted lies in the interior of a tetrahedron.
(ii) The point to be inserted lies in the interior of a face of at least one tetrahedron (and at most two tetrahedra).
(iii) The point to be inserted lies in the interior of an edge of at least one tetrahedron.

Case (i) yields four new tetrahedra, which are constructed by connecting the point to be inserted with the vertices of the tetrahedron containing it. Case (ii) requires all tetrahedra sharing the face containing the point to be split into three new tetrahedra. This is accomplished by connecting the point with the vertices of all tetrahedra (at most two) sharing the face. Case (iii) makes it necessary to split all tetrahedra sharing the edge containing the point into two tetrahedra by connecting the point with the vertices of all tetrahedra sharing that edge. The three cases are illustrated in Fig. 1.

Once a selected data point has been inserted into the triangulation, the triangulation is still subject to local modifications. The triangulation is changed locally if this results in a smaller error. This is the analog to edge swapping algorithms leading to max–min angle triangulations (see (Lawson 1977, 1986)). A data dependent triangulation algorithm, on the other hand, attempts to minimize the approximation error by considering function values.

Fig. 2 shows the two possible triangulations for a convex region bounded by six triangles (configuration 1). The triangulation of this region is obtained by choosing one among two possible triangulations, the first one defined by the two tetrahedra $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_5\}$, the second one defined by the three tetrahedra $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_4, \boldsymbol{v}_5\}$, $\{\boldsymbol{v}_1, \boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5\}$, and $\{\boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5\}$.

Fig. 3 shows the three possible triangulations for a convex region bounded by eight triangles (configuration 2). The triangulation of this region is determined by choosing one among the three possible edges $\overline{\boldsymbol{v}_1\boldsymbol{v}_2}$, $\overline{\boldsymbol{v}_3\boldsymbol{v}_4}$, and $\overline{\boldsymbol{v}_5\boldsymbol{v}_6}$.

When inserting a point into the triangulation leads to configuration 1, the edge $\overline{\boldsymbol{v}_4\boldsymbol{v}_5}$

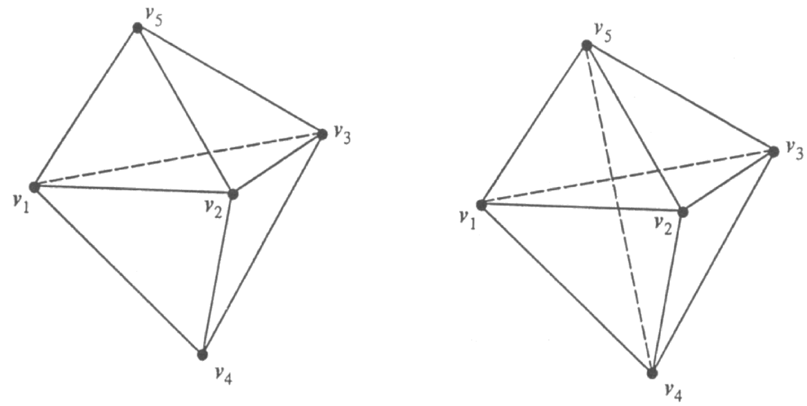

Fig. 2. Triangulations of convex region (configuration 1).

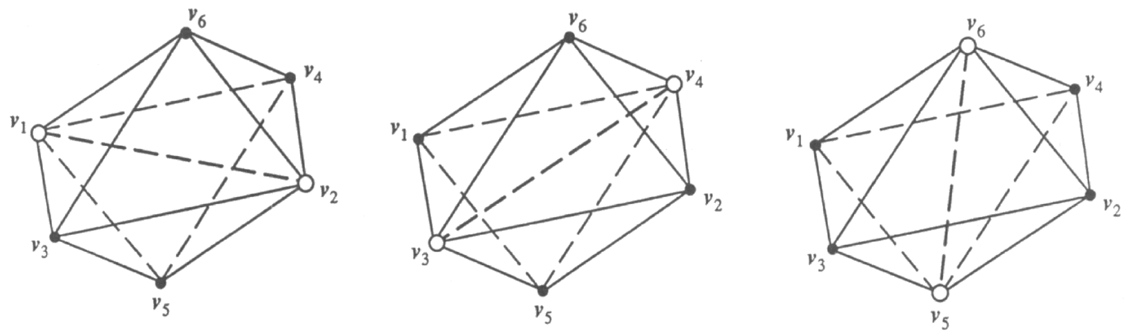

Fig. 3. Triangulations of convex region (configuration 2).

might be inserted into or taken out of the local triangulation. When the insertion of a point leads to configuration 2, the common edge of the four tetrahedra might get replaced by one of the other two possible edges. The algorithm searches for these two configurations and, depending on the resulting errors, constructs the (locally) optimal triangulation. Both configurations are different generalizations of convex quadrilaterals, which are identified in planar edge swapping algorithms. The edge swapping procedure is restricted to a certain neighborhood, which requires the following definition.

Definition 3. Given a triangulation, the *neighbor set* $\mathcal{N}_i$ associated with tetrahedron $\mathcal{T}_i$ is the set of tetrahedra $\mathcal{T}_j$ sharing exactly one face with $\mathcal{T}_i$, formally

$$\begin{aligned}\mathcal{N}_i = \bigcup \{ \mathcal{T}_j = (i_{1,j}, i_{2,j}, i_{3,j}, i_{4,j}), i \neq j \mid \{i_{1,j}, i_{2,j}, i_{3,j}\} \subset \{i_{1,i}, i_{2,i}, i_{3,i}, i_{4,i}\} \\ \vee \{i_{1,j}, i_{2,j}, i_{4,j}\} \subset \{i_{1,i}, i_{2,i}, i_{3,i}, i_{4,i}\} \vee \{i_{1,j}, i_{3,j}, i_{4,j}\} \subset \{i_{1,i}, i_{2,i}, i_{3,i}, i_{4,i}\} \\ \vee \{i_{2,j}, i_{3,j}, i_{4,j}\} \subset \{i_{1,i}, i_{2,i}, i_{3,i}, i_{4,i}\} \}. \qquad (17)\end{aligned}$$

The tetrahedra in $\mathcal{N}_i$ are called *neighbors* of $\mathcal{T}_i$.

When inserting a selected point into the triangulation, an existing tetrahedron $\mathcal{T}_i$ is split into several new tetrahedra. The potential swapping of edges is restricted to its neighbor set $\mathcal{N}_i$ and the new tetrahedra replacing $\mathcal{T}_i$. Configuration 1 and configuration 2 are identified in this set of tetrahedra and considered for edge swapping. Overall, the edge swapping process is characterized by these steps:

(i) Determine the union of the neighbor set $\mathcal{N}_i$ and the set of new tetrahedra replacing $\mathcal{T}_i$; denote the union by $\mathcal{O}_i$.

(ii) If $\mathcal{O}_i$ contains configuration 1 or configuration 2, perform (iii) and (iv); otherwise, retain the triangulation and stop.

(iii) Construct the other possible triangulations for the identified convex polyhedra and select the triangulation minimizing the sum of the single tetrahedral RMS (or maximum) errors.

(iv) If the triangulation has been altered in (iii), update the set $\mathcal{O}_i$ and goto (ii); otherwise, stop.

The selection and insertion of data points terminates when a user-specified number/percentage of all data points has been selected or an error criterion for the implied piecewise trilinear approximation is satisfied. At the end, it is possible to apply the edge swapping algorithm to the set of all tetrahedra.

It must be emphasized that the geometrical characteristics of the resulting tetrahedra (angles, edge lengths, face areas, volumes) are not considered in swapping algorithm. The goal of this triangulation scheme is error minimization, not shape optimization. Obviously, the data dependent triangulation can contain "long", "skinny" tetrahedra, which might cause numerical problems for further processing of the data. On the other hand, the shape of tetrahedra is not that important in applications directly utilizing the piecewise trilinear interpolant.

5. Examples

The data point selection scheme is tested for different trivariate functions. All functions $f(x, y, z)$, $x, y, z \in [-1, 1]$, are evaluated on a $(n+1) \times (n+1) \times (n+1)$ grid with

Table 1
Total RMS errors, $n = 16$ ("*" indicating total RMS errors without considering boundary data)

Function	Selected points (percentage)		
	50%	20%	10%
1. Quadratic polynomial:	0.141	0.350	0.406
$0.4(x^2 + y^2 + z^2)$	0.076*	0.191*	0.313*
2. Quadratic polynomial:	0.115	0.220	0.245
$0.4(x^2 - y^2 - z^2)$	0.057*	0.120*	0.177*
3. Cubic polynomial:	0.080	0.137	0.163
$0.15(x^3 + 2x^2y - xz^2 + 2y^2)$	0.085*	0.140*	0.168*
4. Exponential function:	0.044	0.143	0.165
$e^{-0.5(x^2+y^2+z^2)}$	0.017*	0.062*	0.117*
5. Trigonometric function:	0.030	0.060	0.071
$0.1(\cos(\pi x) + \cos(\pi y) + \cos(\pi z))$	0.036*	0.067*	0.079*

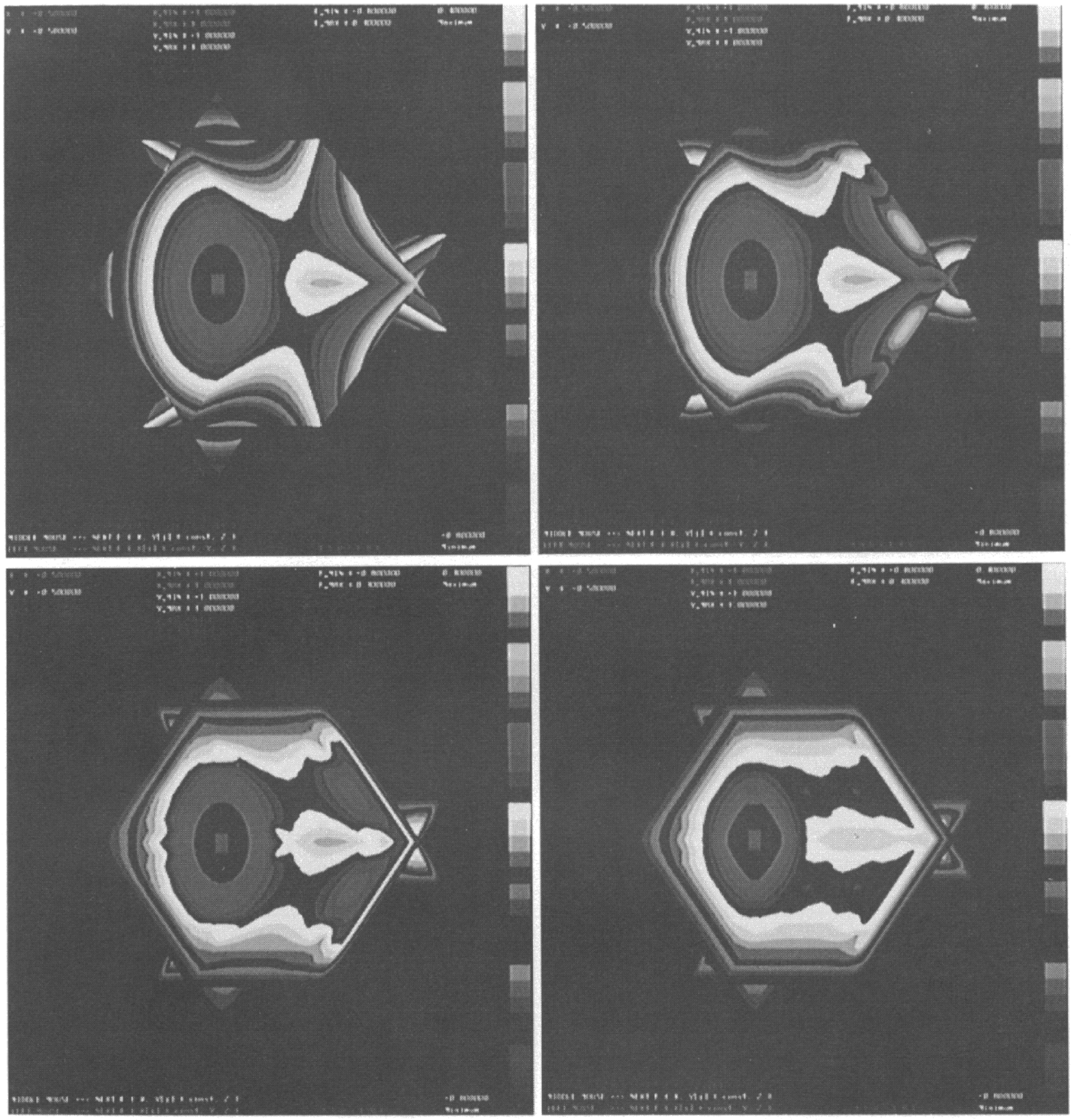

Fig. 4. Original and reduced data sets for $f(x, y, z) = 0.4(x^2 - y^2 - z^2)$, $x, y, z \in [-1, 1]$.

equidistant spacing, i.e.,

$$\mathcal{X}^* = \left\{ \boldsymbol{x}^*_{i,j,k} = \left(\frac{2i-n}{n}, \frac{2j-n}{n}, \frac{2k-n}{n}, f(\boldsymbol{x}_{i,j,k}) \right) \middle| \; i, j, k = 0, \ldots, n \right\}. \qquad (18)$$

The initial triangulation of all $(n+1)^3$ points is defined by splitting each cuboid given by the eight vertices in $\{\boldsymbol{x}_{i+I,j+J,k+K} \mid I, J, K \in \{0, 1\}\}$, $i, j, k = 0, \ldots, n-1$, into six tetrahedra yielding a Delaunay triangulation (see (Nielson et al., 1991)). Since the given data set (18) is rectilinear, only the eight corner points $\boldsymbol{x}_{i,j,k}$, $i, j, k \in \{0, n\}$, are kept. The boundary polyhedron of the data set is defined by twelve triangles.

The total RMS and total maximum approximation errors are defined as

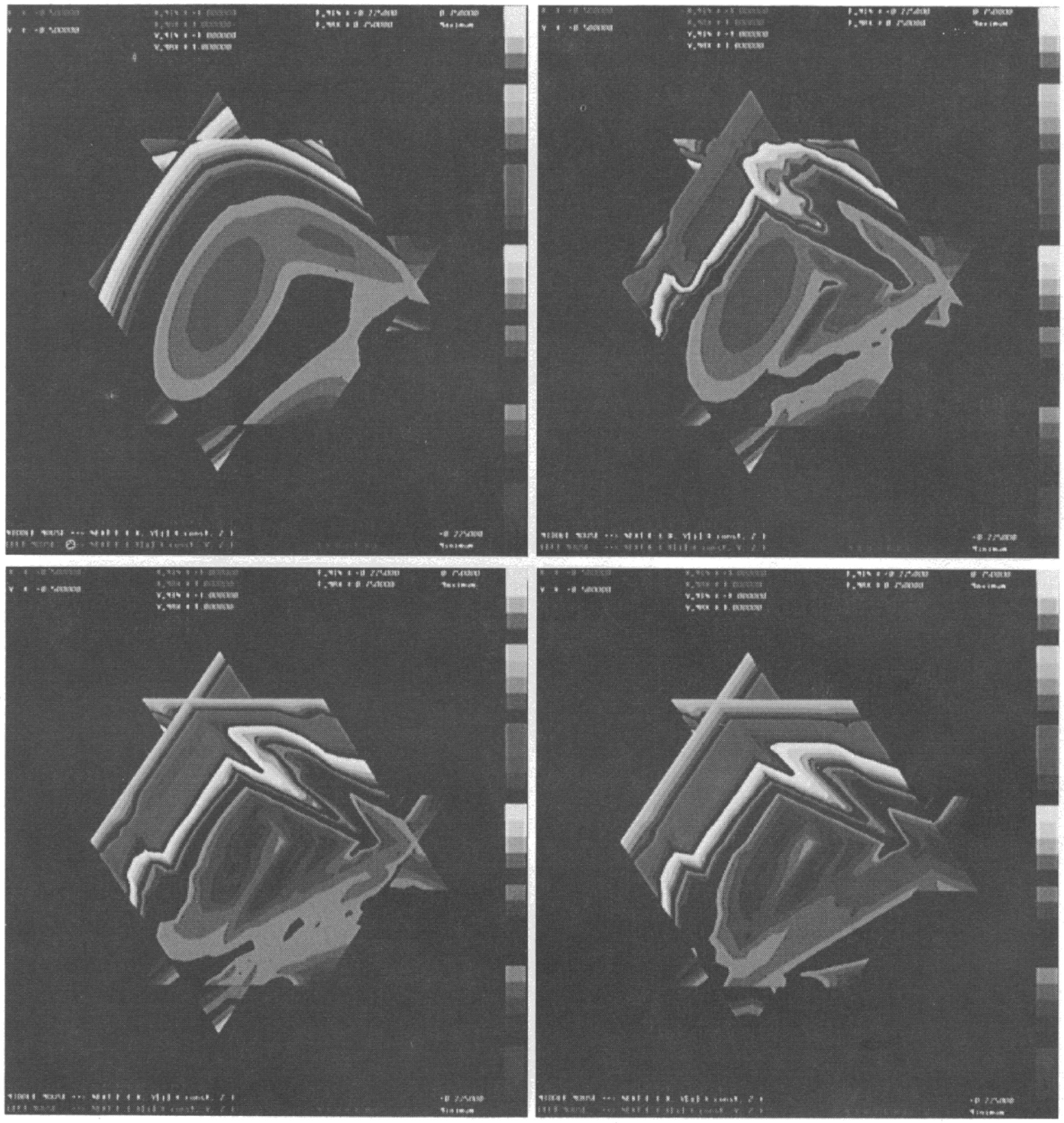

Fig. 5. Original and reduced data sets for $f(x,y,z) = 0.15(x^3 + 2x^2y - xz^2 + 2y^2)$, $x, y, z \in [-1, 1]$.

$$E_{\text{RMS}} = \sqrt{\frac{1}{(n+1)^3} \sum_{k=0}^{n} \sum_{j=0}^{n} \sum_{i=0}^{n} {d_{i,j,k}}^2} \tag{19}$$

and

$$E_\infty = \max_{0 \leqslant i,j,k \leqslant n} |d_{i,j,k}|, \tag{20}$$

where $d_{i,j,k} = f_{i,j,k} - L(\boldsymbol{x}_{i,j,k})$, and L is the trilinear polynomial implied by the tetrahedron containing the point $\boldsymbol{x}_{i,j,k}$. Obviously, $E_{\text{RMS}} = E_\infty = 0$ if all data points originate from a single trilinear polynomial. Table 1 shows the total RMS errors for different trivariate functions and different percentages of selected data points. Since the resolution is small ($n = 16$), the boundary data has a rather strong influence on the resulting

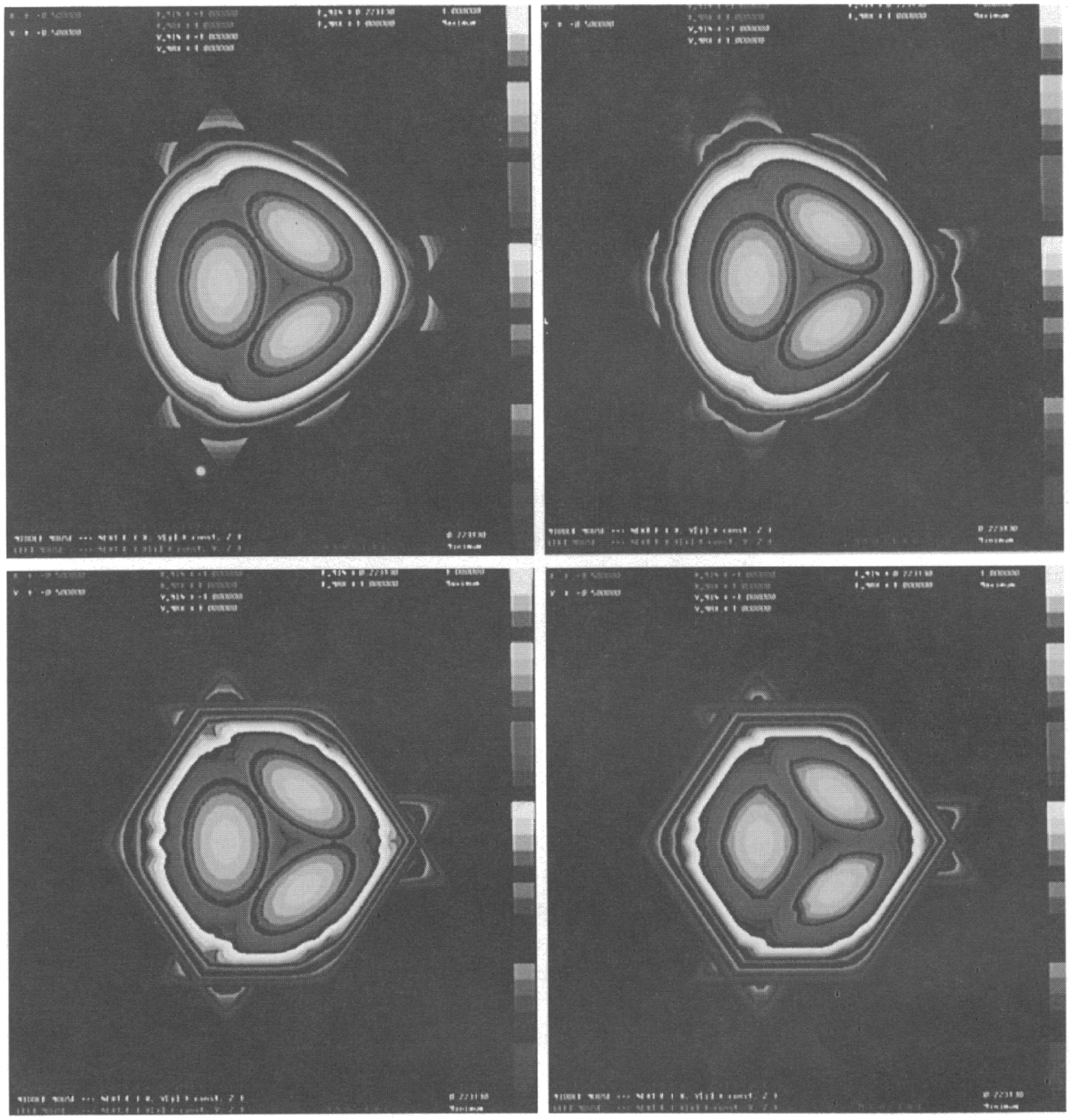

Fig. 6. Original and reduced data sets for $f(x, y, z) = e^{-0.5(x^2+y^2+z^2)}$, $x, y, z \in [-1, 1]$.

errors. Therefore, two total RMS errors are computed. The first one considers data on the boundary, the second one does not.

A trivariate function can be visualized by slicing its domain with planes, evaluating the function in these planes, and representing function values according to some color map (see (Nielson et al., 1991)). The original and reduced data sets for the functions 2, 3, and 4 of Table 1 are visualized by slicing. In Figs. 4–6, the upper-left corner shows the original data set, the upper-right corner 50% selection, the lower-left corner 20% selection, and the lower-right corner 10% selection.

6. Conclusions

A new method for the reduction of trivariate, scalar-valued data has been presented. The method is based on selecting the most important data, not on removing data.

Significant data are identified by large absolute curvatures obtained by a local least square approximation method. The method can be generalized to multivariate, scalar-valued data sets $\{(x_{1,i}, x_{2,i}, \ldots, x_{N,i}, f(x_{1,i}, x_{2,i}, \ldots, x_{N,i})) \mid i = 1, \ldots, n\}$.

The boundary polyhedron/convex hull of the original data can sometimes be preserved by choosing an appropriate subset of all points lying on the convex hull. It will be investigated how to remove redundant data lying on the convex hull without destroying it. Scattered data approximation methods and volume visualization techniques (e.g., ray casting) should benefit significantly from this data reduction strategy.

Acknowledgements

This research was supported by Mississippi State University through its Research Initiation Program (Research Initiation Award 1992) and by the National Science Foundation (Research Initiation Award 1992, grant number ASC-9210439). The reviewers' comments are gratefully acknowledged.

References

Arge, E., Dæhlen, M., Lyche, T. and Mørken, K. (1990), Constrained spline approximation of functions and data based on constrained knot removal, in: Mason, J.C. and Cox, M.G., eds., *Algorithms for Approximation II*, Chapman and Hall, New York, 4–20.

Barth, T.J., Wiltberger, N.L. and Gandhi, A.S. (1992), Three-dimensional unstructured grid generation via incremental insertion and local optimization, in: *Software Systems for Surface Modeling and Grid Generation*, NASA Conference Publication 3143, 449–461.

do Carmo, M.P. (1976), *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ.

Dyn, N., Levin, D. and Rippa, S. (1990a), Data dependent triangulations for piecewise linear interpolation, IMA J. Numer. Analysis 10, 137–154.

Dyn, N., Levin, D. and Rippa, S. (1990b), Algorithms for the construction of data dependent triangulations, in: Mason, J.C. and Cox, M.G., eds., *Algorithms for Approximation II*, Chapman and Hall, New York, 185–192.

Farin, G. (1992), *Curves and Surfaces for Computer Aided Geometric Design*, 3rd ed., Academic Press, San Diego, CA.

Goldman, R.N. and Lyche, T. (1993), *Knot Insertion and Deletion Algorithms for B-Spline Curves and Surfaces*, SIAM, Philadelphia, PA.

Hamann, B. (1994a), A data reduction scheme for triangulated surfaces, Computer Aided Geometric Design 11, 197–214.

Hamann, B. (1994b), Curvature approximation of 3D manifolds in 4D space, Computer Aided Geometric Design 11, to appear.

Hamann, B. and Chen, J.L. (1994), Data point selection for piecewise linear curve approximation, Computer Aided Geometric Design 11, 289–301.

Lawson, C.L. (1977), Software for C^1 surface interpolation, in: Rice, J.R., ed., *Mathematical Software III*, Academic Press, New York, 161–193.

Lawson, C.L. (1986), Properties of n-dimensional triangulations, Computer Aided Geometric Design 3, 231–246.

Le Méhauté, A.J.Y. and Lafranche, Y. (1989), A knot removal strategy for scattered data in $\mathbb{R}^2$, in: Lyche, T. and Schumaker, L.L., eds., *Mathematical Methods in Computer Aided Geometric Design*, Academic Press, San Diego, CA, 419–426.

Lyche, T. and Mørken, K. (1987), Knot removal for parametric B-spline curves and surfaces, Computer Aided Geometric Design 4, 217–230.

Lyche, T. and Mørken, K. (1988), A data-reduction strategy for splines with applications to the approximation of functions and data, IMA J. Numer. Analysis 8, 185–208.

Nielson, G.M., Foley, T.A., Hamann, B. and Lane, D.A. (1991), Visualizing and modeling scattered multivariate data, IEEE Computer Graphics and Applications 11, 47–55.

Preparata, F.P. and Shamos, M.I. (1990), *Computational Geometry*, Third Print., Springer, New York.

Quak, E. and Schumaker, L.L. (1991), Least squares fitting by linear splines on data dependent triangulations, in: Laurent, P.J., Le Méhauté, A. and Schumaker, L.L., eds., *Curves and Surfaces,* Academic Press, San Diego, CA, 387–390.

Rippa, S. (1989), *Piecewise Linear Interpolation and Approximation Schemes over Data Dependent Triangulations*, Ph.D. Thesis, School of Mathematical Sciences, Tel-Aviv University, Tel Aviv.

Spivak, M. (1970), *Comprehensive Introduction to Differential Geometry*, Vols. 1–5, Publish or Perish, Waltham.