



A data reduction scheme for triangulated surfaces

Bernd Hamann^{a,b}

^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*

Received July 1992; revised January 1993

Abstract

Given a surface triangulation in three-dimensional space, an algorithm is developed to iteratively remove triangles from the triangulation. An underlying parametric or implicit surface representation is not required. An order is introduced on the set of triangles by considering curvature at their vertices. Triangles in nearly planar surface regions are prime candidates for removal. The degree of reduction can be specified by a percentage or, in the case of bivariate functions, by an error tolerance.

Key words: Approximation; Curvature; Data reduction; Surface triangulation

1. Introduction

Data reduction schemes are essential for efficient data storage. Storing and processing more data than necessary is rather inefficient. In the context of discretizing curves and surfaces, an efficient scheme should not use more data points than necessary to represent a particular geometrical shape. Using piecewise linear approximation for shape representation, keeping lots of data points in nearly planar regions is rather unsophisticated.

Based on this observation, a data reduction algorithm is developed. Given a surface triangulation in three-dimensional space, each triangle is weighted according to the principal curvatures at its vertices. These curvature values are pre-computed based on the discrete, triangulated representation of the surface. A triangle is associated with a surface region with low curvature if the sum of the absolute curvatures at its vertices is low. This measure is used as a weight to determine a triangle's significance.

The overall strategy consists of the following steps. First, curvature estimates are computed for all vertices in the given triangulation. These estimates determine weights for all triangles. The lower the curvature values at the vertices of a triangle, the lower its weight. The triangle with the lowest weight is identified

and removed from the triangulation. It is replaced by a new point that lies on a local surface approximant. The region affected by a triangle's removal is re-triangulated, and weights are computed for all new triangles introduced.

An iterative algorithm is developed which removes the triangle with the lowest weight in each step. Thus, the triangulation changes iteratively, and the local density of triangles reflects the surface's curvature. At the end, surface regions with low curvature are represented by relatively larger triangles.

It is possible to consider both curvature estimates at the vertices of a triangle and interior angles. A formula combining curvatures and angles is used to avoid undesirable "long" and "skinny" triangles. It must be mentioned that, in some cases, a triangle cannot be removed. This is due to the re-triangulation procedure that must not introduce piercing or overlapping triangles in the case of (locally) planar surfaces.

The method does not require any underlying parametric or implicit surface representation for the given triangulation. The curvature estimates themselves are generated based on the discrete representation only. If an analytical surface representation is given, one should take advantage of it and modify the algorithm appropriately. Here, the entire technique relies on discrete surface information as input.

Commonly, the term data-dependent triangulation is used when function values at data points in the plane are considered for constructing a "good" triangulation. This concept is discussed in (Dyn et al., 1988, 1990). Knot removal strategies for spline curves and tensor product surfaces (in the context of bivariate functions) are described in (Arge et al., 1990) and (Lyche and Mørken, 1987, 1988). Given scattered points in the plane and associated function values, an iterative knot removal algorithm is discussed in (Le Méhauté and Lafranche, 1989). The technique presented in this paper requires a curvature approximation scheme for discrete surface data. The scheme used is discussed in (Hamann, 1993a).

The new data reduction technique is similar to the method described in (Le Méhauté and Lafranche, 1989). However, the method discussed below removes triangles instead of points. Furthermore, it is not restricted to a surface triangulation obtained as the graph of a bivariate function. The new scheme can be applied to more general surface triangulations (e.g., triangulations of parametric surfaces and contours of trivariate functions).

The triangle removal algorithm allows the user to specify a percentage of triangles to be removed or an error tolerance. Currently, the second alternative is implemented for graphs of bivariate functions only. The removal of a triangle (and therefore its three vertices) requires a re-triangulation strategy. It is advantageous to compute a new point that replaces the vertices of a removed triangle. The "hole" created by removing a triangle can also be re-triangulated without introducing a new point. Unfortunately, no efficient re-triangulation strategy could be found without the introduction of a new point.

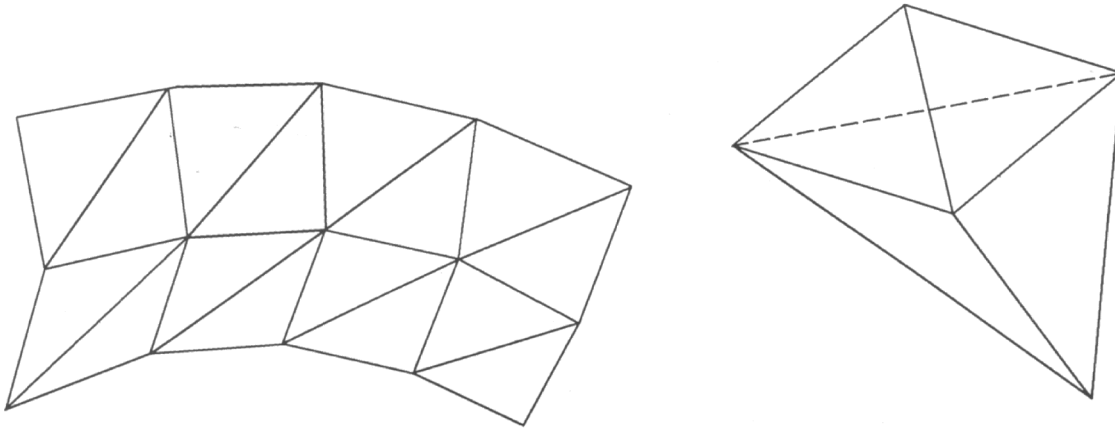


Fig. 1. Connected, knot-to-knot surface triangulations.

2. Definitions and requirements

Some necessary definitions are given before presenting the overall triangle removal algorithm. A certain data stencil is involved when removing a triangle, and certain conditions must hold for this stencil for a triangle to be removable. This is outlined in the following paragraphs. Only surface triangulations belonging to the type defined next are considered.

Definition 2.1. A *connected, knot-to-knot surface triangulation* is a finite set of triangles satisfying the following conditions (cf. Fig. 1):

- Each edge in the triangulation is shared by at most two triangles (no bifurcations).
- A vertex in the triangulation can be shared by any number of triangles.
- Each triangle has at least one point in common with another triangle (connectivity; requires at least two triangles).
- If a vertex of a triangle is shared by a second triangle, then this point is also a vertex of the second triangle (knot-to-knot property).
- No triangle has an intersection with the interior of any other triangle (no piercing/overlapping).

Two triangles are called *neighbors* if they share a common edge.

Definition 2.2. Considering a connected, knot-to-knot surface triangulation, the *triangle platelet* \mathcal{P}_i associated with a triangle T_i (identified with the index triple (v_1^i, v_2^i, v_3^i) specifying its vertices) is the set of all triangles T_j sharing at least one of T_i 's vertices as a common vertex.

The triangle platelet \mathcal{P}_i is the data stencil in the surface triangulation that is affected by the removal of triangle T_i .

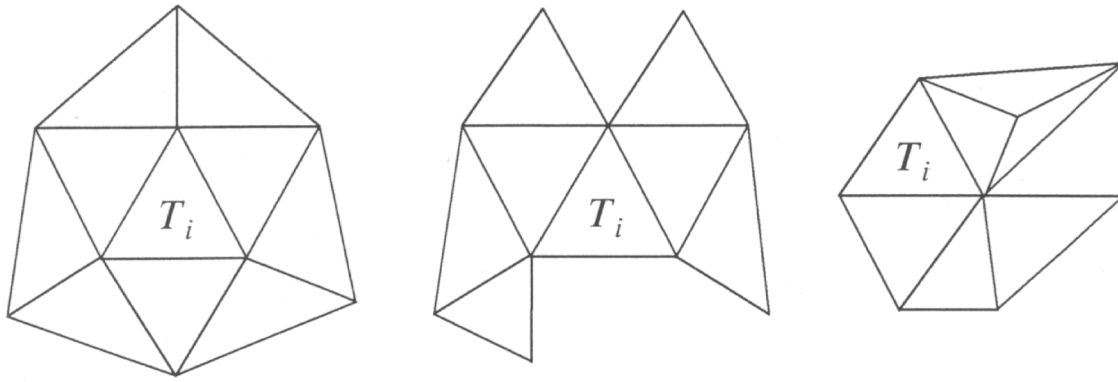


Fig. 2. Triangle platelets with connected/disconnected corona and with cyclic corona.

Definition 2.3. The set of triangles

$$C_i = \mathcal{P}_i \setminus \{T_i\} \quad (1)$$

is called the *corona* of the triangle platelet \mathcal{P}_i . The *corona* C_i is called *connected* if for each pair of triangles $T_{l_1}, T_{l_m} \in C_i$ triangles $T_{l_2}, \dots, T_{l_{m-1}} \in C_i$ exist such that

$$T_{l_i} \text{ and } T_{l_{i+1}} \text{ are neighbors, } i = 1, \dots, m-1; \quad (2)$$

otherwise, the *corona* is called *disconnected*. The *corona* C_i is called *cyclic* if it contains triangles T_{l_0}, T_{l_1} , and T_{l_2} such that

$$T_{l_i} \text{ and } T_{l_{((i+1) \bmod 3)}} \text{ are neighbors, } i = 0, 1, 2; \quad (3)$$

otherwise, the *corona* is called *acyclic*. The *corona* C_i is called *complete* if each triangle in C_i has exactly two neighbors which are also elements of C_i ; otherwise, the *corona* is called *incomplete*.

Only triangles surrounded by a connected and acyclic corona can be removed. Fig. 2 illustrates triangle platelets with connected/disconnected corona and one with a cyclic corona.

Definition 2.4. The set of vertices in the triangle platelet \mathcal{P}_i without the vertices of triangle T_i itself is called the *boundary vertex set* of the triangle platelet \mathcal{P}_i .

Based on the knowledge whether triangles are neighbors or not, it is possible to order the triangles in a corona (e.g., counterclockwise). The edges implied by the boundary vertex set can be ordered such that a polygon describing the polygonal boundary of the triangle platelet is obtained (“platelet boundary polygon”). In the following, it is assumed that both T_i ’s vertices and the vertices of its platelet boundary polygon are oriented in the same way. This order and the platelet boundary polygon can be computed straightforward. An example is illustrated in Fig. 3. Arrows indicate the orientation of edges.

It must be emphasized that certain triangles cannot be removed in nearly planar surface regions if their removal would introduce serious problems for

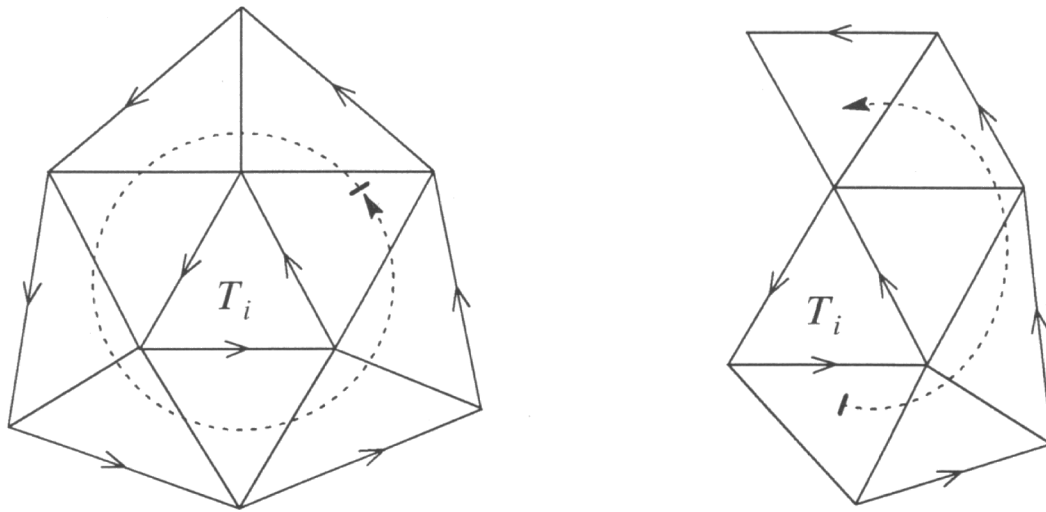


Fig. 3. Triangle platelets with ordered triangles defining their corona and oriented platelet boundary polygons.

the re-triangulation process. Based on this observation, a half-space test is used that determines whether a triangle can be removed or not. This test projects a triangle platelet into the plane implied by triangle T_i being a candidate for removal. The test involves these steps:

- (i) Determine the plane equation of the plane P given by T_i .
- (ii) Define an orthonormal coordinate system in P with T_i 's centroid as origin and two orthogonal unit vectors in P .
- (iii) Compute the distances of all points of the platelet boundary polygon from P .
- (iv) Project all points of the platelet boundary polygon into P and express the projections in terms of the local coordinate system in P .
- (v) Compute line equations L_j in P determined by the projected line segments of the platelet boundary polygon.
- (vi) Test whether the centroid of T_i lies in the (planar) region obtained as the intersection of all half-spaces $L_j > 0$. If this condition is violated, T_i must not be removed.

These steps are now discussed in detail. The outward unit normal vector of the plane P is denoted by

$$\mathbf{n} = (n^x, n^y, n^z)^T = \frac{\mathbf{d}_1 \times \mathbf{d}_2}{\|\mathbf{d}_1 \times \mathbf{d}_2\|}, \tag{4}$$

where $\mathbf{d}_1 = \mathbf{a}_1 - \mathbf{a}_0$ and $\mathbf{d}_2 = \mathbf{a}_2 - \mathbf{a}_0$ are defined by T_i 's vertices \mathbf{a}_j , and “ $\| \ \|$ ” is the Euclidean norm. Writing the plane equation for P as

$$\mathbf{n} \cdot (\mathbf{x} - \mathbf{c}) = Ax + By + Cz + D = 0, \tag{5}$$

where \mathbf{c} is T_i 's centroid, two orthogonal unit basis vectors in the plane P are defined by

$$\mathbf{b}_1 = \frac{\mathbf{d}_1}{\|\mathbf{d}_1\|} \quad \text{and} \quad \mathbf{b}_2 = \mathbf{n} \times \mathbf{b}_1. \quad (6)$$

The signed distances d_j , $j = 0, \dots, n_i$, of the $(n_i + 1)$ platelet boundary points $\mathbf{y}_j = (x_j, y_j, z_j)^T$ are

$$d_j = Ax_j + By_j + Cz_j + D. \quad (7)$$

Projecting these platelet boundary points into P yields the points

$$\mathbf{y}_j^P = \mathbf{y}_j - d_j \mathbf{n}. \quad (8)$$

Expressing \mathbf{y}_j^P in terms of the local coordinate system yields the tuples

$$(u_j, v_j) = (\mathbf{d}_j \cdot \mathbf{b}_1, \mathbf{d}_j \cdot \mathbf{b}_2), \quad (9)$$

where $\mathbf{d}_j = \mathbf{y}_j^P - \mathbf{c}$.

The points \mathbf{y}_j^P define a planar polygon which might intersect itself. The line equations of the single segments are expressed using the local coordinate system. The implicit line equation for the (oriented) line $L_j(u, v)$ is given by

$$L_j(u, v) = -\Delta v_j(u - u_j) + \Delta u_j(v - v_j) = 0, \quad (10)$$

where $\Delta u_j = u_{(j+1) \bmod (n_i+1)} - u_j$ and $\Delta v_j = v_{(j+1) \bmod (n_i+1)} - v_j$.

These line equations are used to determine whether a triangle T_i can be removed or not. If the centroid \mathbf{c} (having local coordinates $(0, 0)$) lies in the positive half-space of all lines L_j , the triangle T_i can be removed. This implies that \mathbf{c} must be an element of the solution set to the system of the linear inequalities

$$L_j(u, v) > 0. \quad (11)$$

Only triangles surrounded by a connected and acyclic corona passing this half-space test can be removed. Fig. 4 shows the half-space test applied to a triangle T_i that passes the test.

3. Computing weights for triangles

In order to determine the weight of a triangle in the surface triangulation, the principal curvature estimates at its vertices and its interior angles are considered. The principal curvature estimates are part of the input of the reduction scheme and are based on an approximation scheme for surface triangulations (see (Farin, 1992) and (Hamann, 1993a)).

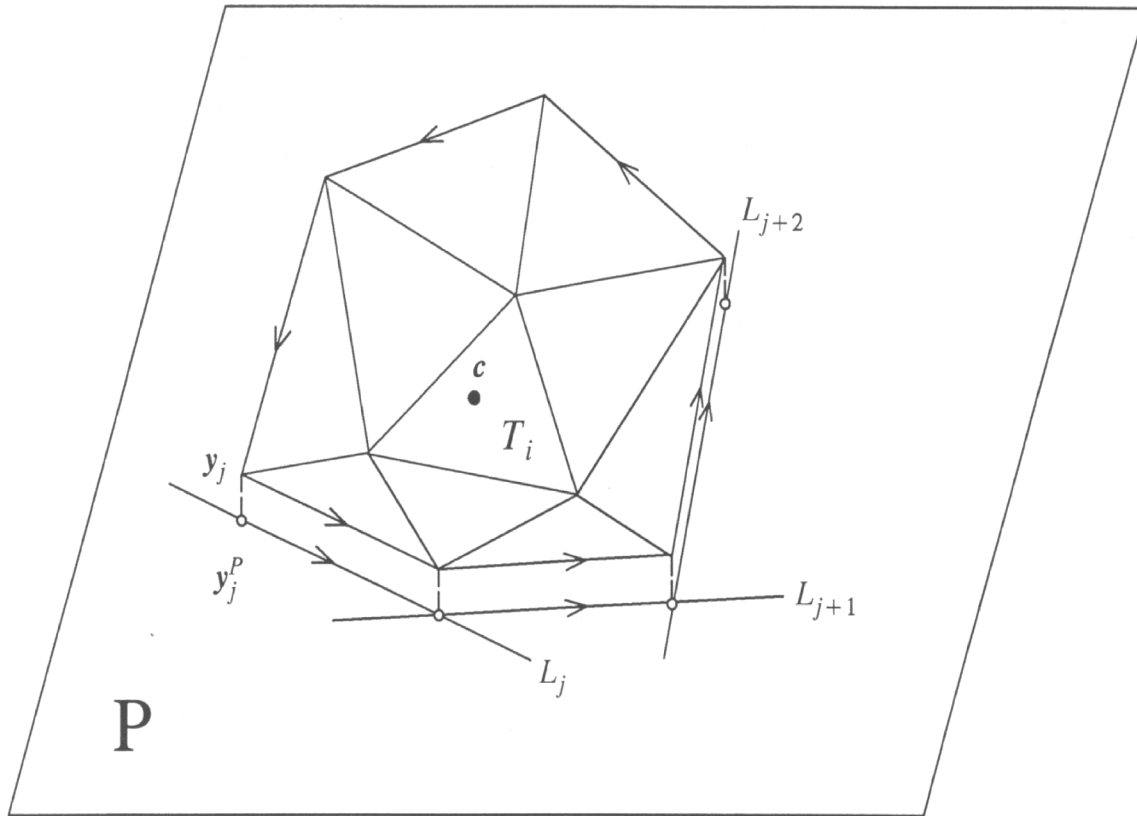


Fig. 4. Boundary vertex set and projection into P ; triangle passing half-space test.

Definition 3.1. The sum A of the absolute values of the principal curvatures κ_1 and κ_2 at a point x in a surface triangulation is called the *absolute curvature*,

$$A = |\kappa_1| + |\kappa_2|. \tag{12}$$

Each triangle is weighted by the three absolute curvatures at its vertices. In order to consider the shape of a triangle, the interior angles are taken into account as well. Equilateral triangles are favored. The combination of absolute curvatures and interior angles as significance measure requires the next definitions.

Definition 3.2. The *angle weight* σ_i of a triangle T_i is given by

$$\sigma_i = 2 \left(\left(\sum_{j=1}^3 \cos \alpha_j \right) - 1 \right) \in [0, 1], \tag{13}$$

where $\alpha_j, j = 1, 2, 3$, are T_i 's interior angles.

The following lemma proves that the range of σ_i is in fact the interval $[0, 1]$.

Lemma 3.3. Denoting the interior angles of a triangle by α_1 , α_2 , and α_3 , the range of the function

$$f(\alpha_1, \alpha_2, \alpha_3) = \sum_{j=1}^3 \cos \alpha_j \quad (14)$$

is the interval $[1, \frac{3}{2}]$.

Proof. Since $\sum_{j=1}^3 \alpha_j = \pi$, it is sufficient to analyze the bivariate function

$$g(\alpha_1, \alpha_2) = \cos \alpha_1 + \cos \alpha_2 + \cos(\pi - (\alpha_1 + \alpha_2))$$

on the domain $D = \{(\alpha_1, \alpha_2) \mid \alpha_1, \alpha_2 \geq 0, \alpha_1 + \alpha_2 \leq \pi\}$. One easily proves that g equals 1 on D 's boundary:

$$g(0, \alpha_2) = 1 + \cos \alpha_2 + \cos(\pi - \alpha_2) = 1,$$

$$g(\alpha_1, 0) = \cos \alpha_1 + 1 + \cos(\pi - \alpha_1) = 1, \quad \text{and}$$

$$g(\alpha_1, \pi - \alpha_1) = \cos \alpha_1 + \cos(\pi - \alpha_1) + 1 = 1.$$

A critical point must satisfy the equations

$$\frac{\partial g}{\partial \alpha_1}(\alpha_1, \alpha_2) = -\sin \alpha_1 + \sin(\pi - (\alpha_1 + \alpha_2)) = 0$$

and

$$\frac{\partial g}{\partial \alpha_2}(\alpha_1, \alpha_2) = -\sin \alpha_2 + \sin(\pi - (\alpha_1 + \alpha_2)) = 0.$$

Therefore, $\sin \alpha_1 = \sin \alpha_2$ is a necessary condition, which holds for $\alpha_1 = \alpha_2$ and $\alpha_2 = \pi - \alpha_1$. The first case ($\alpha_1 = \alpha_2$) defines the univariate function

$$h(\alpha_1) = 2 \cos \alpha_1 + \cos(\pi - 2\alpha_1)$$

having critical points at $\alpha_1 = 0$ and $\alpha_1 = \frac{1}{3}\pi$, since $h'(0) = h'(\frac{1}{3}\pi) = 0$. Considering $\alpha_1 = \frac{1}{3}\pi$ results in the function value $f(\frac{1}{3}\pi, \frac{1}{3}\pi, \frac{1}{3}\pi) = \frac{3}{2}$. The second case ($\alpha_2 = \pi - \alpha_1$) defines part of D 's boundary where f equals 1. \square

The weight function in Eq. (13) assigns maximum weight to an equilateral triangle and small weights to “long” and “skinny” triangles.

Definition 3.4. The curvature weight ρ_i of a triangle T_i is given by the sum of the absolute curvatures at its vertices,

$$\rho_i = \sum_{j=1}^3 A_j, \quad (15)$$

where $A_j = |\kappa_1^j| + |\kappa_2^j|$, $j = 1, 2, 3$, are the absolute curvatures at T_i 's vertices.

Definition 3.5. The *triangle weight* ω_i of a triangle T_i is given by

$$\omega_i = \sigma_i \rho_i. \quad (16)$$

4. Removing a triangle from the triangulation

Assuming that triangle T_i has the lowest triangle weight among all triangles and that it is removable, triangle T_i is removed from the triangulation by replacing its vertices by a new point p , whose construction is explained next. The new point is connected to each point in \mathcal{P}_i 's boundary vertex set determining a first re-triangulation. The new point is computed by considering the local surface geometry. This is achieved by constructing a local least-squares polynomial approximation to the data and requiring p to lie on this approximant.

Two possibilities are discussed for the construction of the new point p replacing the triangle T_i . The first possibility is to construct a bivariate function $f(u, v)$, using an appropriate coordinate system for the points determining T_i 's triangle platelet (like the one used in the half-space test), and to evaluate f at $(0, 0)$. The second possibility is to compute an implicit function $f(x, y, z) = 0$, considering a similar set of data points, and to generate the new point by intersecting a line with the implicitly defined surface.

The first possibility has been chosen for the implementation and is described in detail. The construction of the bivariate function $f(u, v)$ follows the same principle as the half-space test. The only difference is the choice of an origin o in the plane defined by the triangle T_i . The corona determines where the origin is placed. The rationale for choosing different origins lies in the desire to remain close to the polygonal boundary of the original surface triangulation when the surface triangulation is not closed. Obviously, the five cases listed below are heuristic. The different cases for choosing the origin are as follows:

- If the corona \mathcal{C}_i is complete, the centroid of T_i is chosen.
- If T_i has three neighbors, but its corona is open, the common vertex of T_i and the first and last triangle in \mathcal{C}_i is chosen.
- If T_i has two neighbors, and the first and last triangle in \mathcal{C}_i are both (are both not) neighbors of T_i , the mid-point of T_i 's edge not shared by another triangle is chosen.
- If T_i has two neighbors, and the first (the last) triangle in \mathcal{C}_i is a neighbor of T_i , and the last (the first) triangle in \mathcal{C}_i is not a neighbor of T_i , the vertex only shared by T_i and the first (the last) triangle in \mathcal{C}_i is chosen.
- If T_i has a vertex not shared by another triangle, this vertex is chosen.

These different choices for the origin o are shown in Fig. 5.

Denoting the vertices in T_i 's triangle platelet by x_1, \dots, x_{n_i} (which includes both T_i 's vertices and its boundary vertex set), they have local coordinates (u_j, v_j) in the plane P . Considering these points' distances d_j from P , a

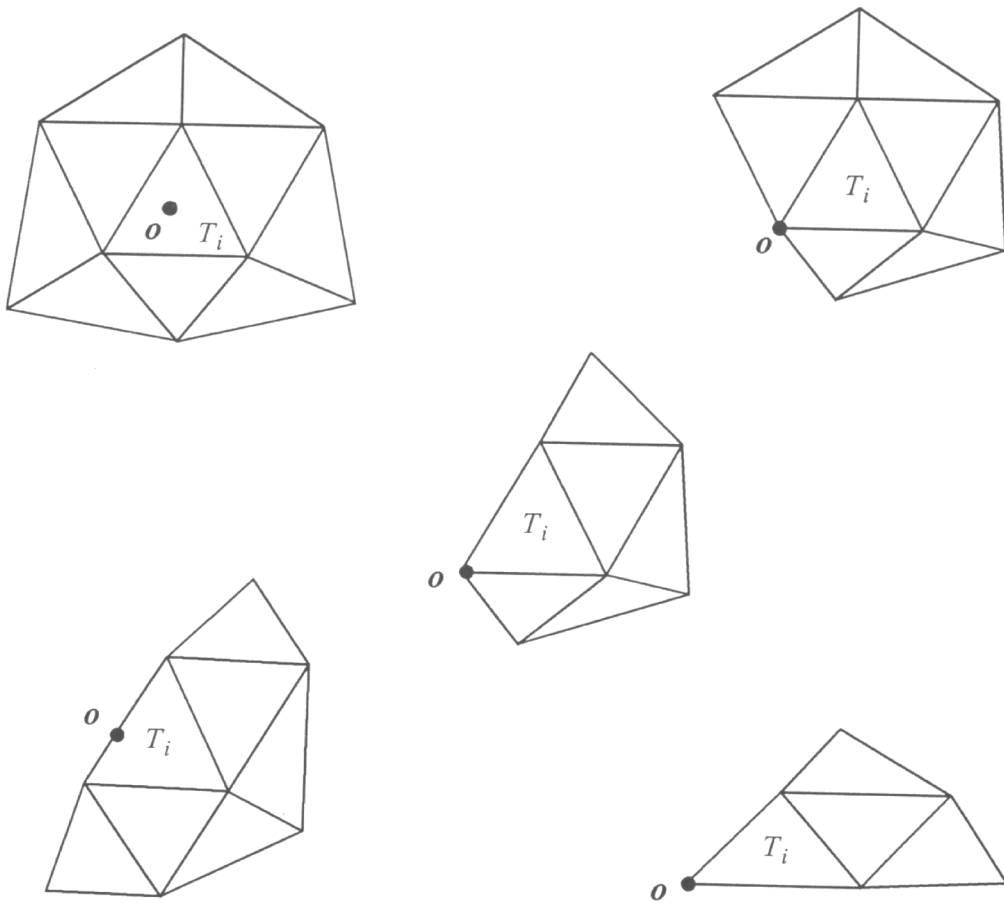


Fig. 5. Different choices for origin o of local approximant.

least-squares, degree-two polynomial is constructed using the conditions

$$\begin{pmatrix} u_1^2 & u_1 v_1 & u_1 & v_1^2 & v_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{n_i}^2 & u_{n_i} v_{n_i} & u_{n_i} & v_{n_i}^2 & v_{n_i} & 1 \end{pmatrix} \begin{pmatrix} c_{2,0} \\ c_{1,1} \\ c_{1,0} \\ c_{0,2} \\ c_{0,1} \\ c_{0,0} \end{pmatrix} = U \begin{pmatrix} c_{2,0} \\ c_{1,1} \\ c_{1,0} \\ c_{0,2} \\ c_{0,1} \\ c_{0,0} \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_{n_i} \end{pmatrix}. \quad (17)$$

Solving the normal equations determines a local approximant, provided that the determinant of $U^T U$ does not vanish. If it does vanish, the number of points used in the construction is increased appropriately. Finally, the point

$$p = o + f(0,0)n, \quad (18)$$

where n is T_i 's outward unit normal vector (ordinate direction of f), is used to replace triangle T_i . Since this new point becomes a vertex in the triangulation, principal curvature estimates must be computed for it. These estimates are obtained directly from the local approximant f using the following lemma.

Lemma 4.1. *The two principal curvatures κ_1 and κ_2 of the graph $(u, v, f(u, v))^T \subset \mathbb{R}^3$, $u, v \in \mathbb{R}$, of the bivariate polynomial*

$$f(u, v) = \sum_{\substack{i+k \leq 2 \\ i, k \geq 0}} c_{i,k} u^i v^k \tag{19}$$

at the point $(0, 0, f(0, 0))^T$ are given by the two real roots of the quadratic equation

$$\det \begin{pmatrix} 2c_{2,0}(1 + c_{0,1}^2) - c_{1,1}c_{1,0}c_{0,1} - \kappa & -2c_{2,0}c_{1,0}c_{0,1} + c_{1,1}(1 + c_{1,0}^2) \\ c_{1,1}(1 + c_{0,1}^2) - 2c_{0,2}c_{1,0}c_{0,1} & -c_{1,1}c_{1,0}c_{0,1} + 2c_{0,2}(1 + c_{1,0}^2) - \kappa \end{pmatrix} = 0 \tag{20}$$

Proof. The principal curvatures of f 's graph are the eigenvalues of the matrix

$$-A = \frac{1}{l_1} \begin{pmatrix} f_{uu} & f_{uv} \\ f_{uv} & f_{vv} \end{pmatrix} \begin{pmatrix} 1 + f_u^2 & f_u f_v \\ f_u f_v & 1 + f_v^2 \end{pmatrix}^{-1},$$

where $l_1 = \sqrt{1 + f_u^2 + f_v^2}$ (see (do Carmo, 1976)). Evaluating $-A$ for $u = v = 0$, one obtains the matrix

$$\frac{1}{l_1} \begin{pmatrix} 2c_{2,0} & c_{1,1} \\ c_{1,1} & 2c_{0,2} \end{pmatrix} \begin{pmatrix} 1 + c_{1,0}^2 & c_{1,0}c_{0,1} \\ c_{1,0}c_{0,1} & 1 + c_{0,1}^2 \end{pmatrix}^{-1},$$

where $l_1 = \sqrt{1 + c_{1,0}^2 + c_{0,1}^2}$, having the characteristic polynomial given in (20). \square

The two roots of (20), κ_1 and κ_2 , determine the absolute curvature at the point \mathbf{p} and therefore the curvature weights for all triangles sharing \mathbf{p} as a common vertex.

The second possibility to generate the point \mathbf{p} is the construction of an implicitly defined surface, e.g., a quadric surface $f(x, y, z) = 0$ using the conditions

$$f(x_j, y_j, z_j) = \sum_{\substack{i+k+l \leq 2 \\ i, k, l \geq 0}} c_{i,k,l} x_j^i y_j^k z_j^l = 0, \quad j = 1, \dots, n_i \geq 9,$$

and one additional linear condition, e.g.,

$$f(1, 1, 1) = \sum_{\substack{i+k+l \leq 2 \\ i, k, l \geq 0}} c_{i,k,l} = 1, \tag{21}$$

where $(x_j, y_j, z_j)^T$ is a vertex in T_i 's triangle platelet and $(x_j, y_j, z_j) \neq (1, 1, 1)$. If the triangle platelet does not provide a sufficient number of conditions, additional points are necessary.

The solution is a quadric surface that locally approximates the vertices in the triangle platelet \mathcal{P}_i . This approach has several drawbacks. First, it requires

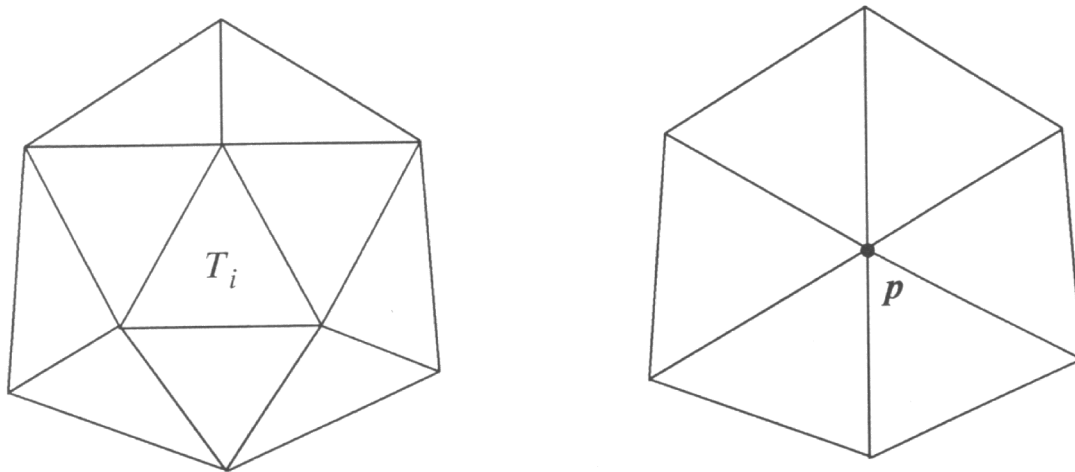


Fig. 6. Removal of triangle T_i and first re-triangulation.

at least nine points to compute the local approximant. Second, one has to take care of the special case when all points belong to the zero set of a quadric.

Once a quadric has been computed, the point p replacing T_i is obtained by intersecting the quadric with the line $x(t) = o + tn$, $t \in \mathbb{R}$ (see equation (18)). The resulting quadratic equation

$$\sum_{\substack{i+k+l \leq 2 \\ i,k,l \geq 0}} c_{i,k,l} (x(t))^i (y(t))^k (z(t))^l = 0$$

might (or might not) have a real solution. The real solution associated with the point closest to o is chosen as the new point p . The fact that this point might not be unique and the possibility of having no intersection at all are additional disadvantages of the implicit approach.

Principal curvatures must be computed for the new point p on the quadric. It is well known in differential geometry how to compute the principal curvatures in this case, and an appropriate reference is (O'Neill, 1969).

5. Local re-triangulation and the reduction algorithm

A first local re-triangulation of the boundary vertex set and the new point p is constructed by introducing edges from each vertex in B_i to p . Fig. 6 shows this process.

In order to obtain a good re-triangulation in the sense of optimizing the angle weights (see Definition 3.2), an iterative edge swapping algorithm is applied to the set of new triangles (see (Lawson, 1977) and (Choi et al., 1988)). The main idea is to swap diagonals of (non-planar) quadrilaterals until angle weights can no longer be improved. The diagonal of a quadrilateral is swapped only if the region obtained by projecting it into the plane P is convex. Fig. 7 shows the improvement of angle weights by swapping a diagonal.

The overall triangle reduction algorithm can be summarized as follows:

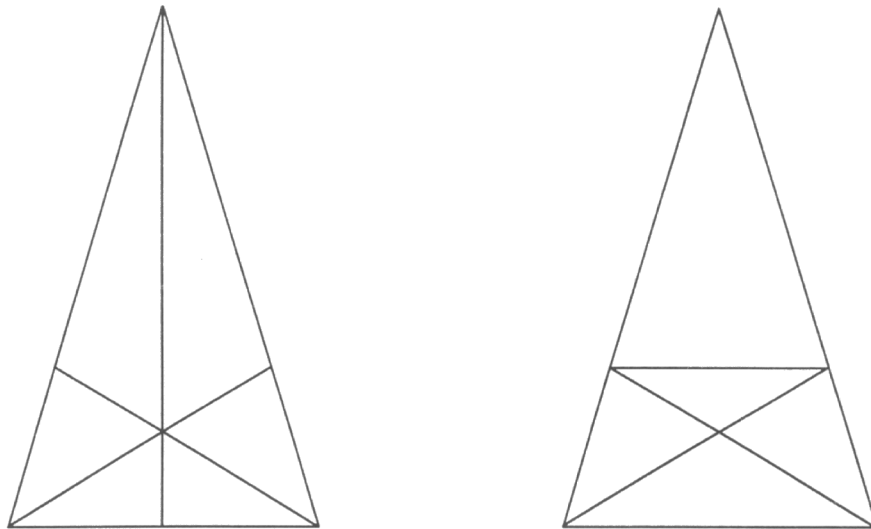


Fig. 7. Increasing angle weights of triangles.

Algorithm (*Triangle reduction by iterative triangle removal*).

Input: set \mathcal{T} of N triangles (including neighborhood information),
 set \mathcal{V} of vertices (including principal curvatures), and
 a percentage $p \in [0, 100]$.

Output: reduced set $\hat{\mathcal{T}}$ of triangles and reduced set $\hat{\mathcal{V}}$ of vertices.

compute weights for each triangle in \mathcal{T} ;

while number of triangles is greater than $(p/100)N$

(among all triangles having a connected, acyclic corona and passing the
 half-space test, determine the triangle T_i with minimal weight;

remove triangle T_i from triangulation (using either the bivariate or
 trivariate, implicit least squares approximation);

compute a first re-triangulation;

improve the re-triangulation by

maximizing the minimum angle weight;

compute weights for all new triangles;

)

Triangles not surrounded by a complete corona, such as triangles containing boundary vertices or boundary edges of the original triangulation, can be kept by marking them at the beginning. This, however, will preserve the original density of points on the polygonal boundary. In the case of closed surface triangulations, this problem does not arise.

The termination criterion in the algorithm (using a percentage) can be modified in the case of the graph of a bivariate function. A (discrete) root-mean-square (RMS) error can be computed by computing distances between points on the original piecewise linear function and the reduced one. The algorithm stops when a prescribed error tolerance is exceeded.

It is possible that the algorithm cannot find a triangle with a connected,

acyclic corona passing the half-space test. In this case, the algorithm stops.

6. Numerical test results and examples

The triangle reduction strategy has been tested numerically for graphs of bivariate functions. All test functions $f(x, y)$ are defined over $[-1, 1] \times [-1, 1]$ and evaluated on a 51-by-51 grid with uniform spacing,

$$(x_i, y_j) = \left(-1 + \frac{i}{25}, -1 + \frac{j}{25} \right), \quad i, j = 0, \dots, 50,$$

determining a set of points on their graphs,

$$\{(x_i, y_j, f(x_i, y_j))^T \mid i, j = 0, \dots, 50\}.$$

A graph's original triangulation is obtained by splitting each quadrilateral identified with its index quadruple

$$((i, j), (i + 1, j), (i + 1, j + 1), (i, j + 1))$$

into the two triangles

$$((i, j), (i + 1, j), (i + 1, j + 1)) \quad \text{and}$$

$$((i, j), (i + 1, j + 1), (i, j + 1)).$$

Original and reduced triangulation, which are both piecewise linear functions, are compared at the original knots (x_i, y_j) , $i, j = 0, \dots, 50$. The discrete RMS error is given by

$$\frac{1}{51} \sqrt{\sum_{j=0}^{50} \sum_{i=0}^{50} (f(x_i, y_j) - \hat{f}(x_i, y_j))^2}, \quad (22)$$

where \hat{f} denotes the piecewise linear function implied by the reduced triangulation.

Table 1
RMS errors for triangle reduction of graphs of bivariate functions

Function		50%	80%	90%
1. Cylinder:	$\sqrt{2 - x^2}$.00049	.00100	.00211
2. Sphere:	$\sqrt{4 - (x^2 + y^2)}$.00045	.00123	.00229
3. Paraboloid:	$.4(x^2 + y^2)$.00061	.00159	.00362
4. Hyperboloid:	$.4(x^2 - y^2)$.00025	.00079	.00201
5. Monkey saddle:	$.2(x^3 - 3xy^2)$.00036	.00084	.00185
6. Cubic polynomial:	$.15(x^3 + 2x^2y - xy + 2y^2)$.00038	.00103	.00216
7. Exponential function:	$e^{-\frac{1}{2}(x^2 + y^2)}$.00034	.00092	.00208
8. Trigonometric function:	$.1(\cos(\pi x) + \cos(\pi y))$.00036	.00109	.00205

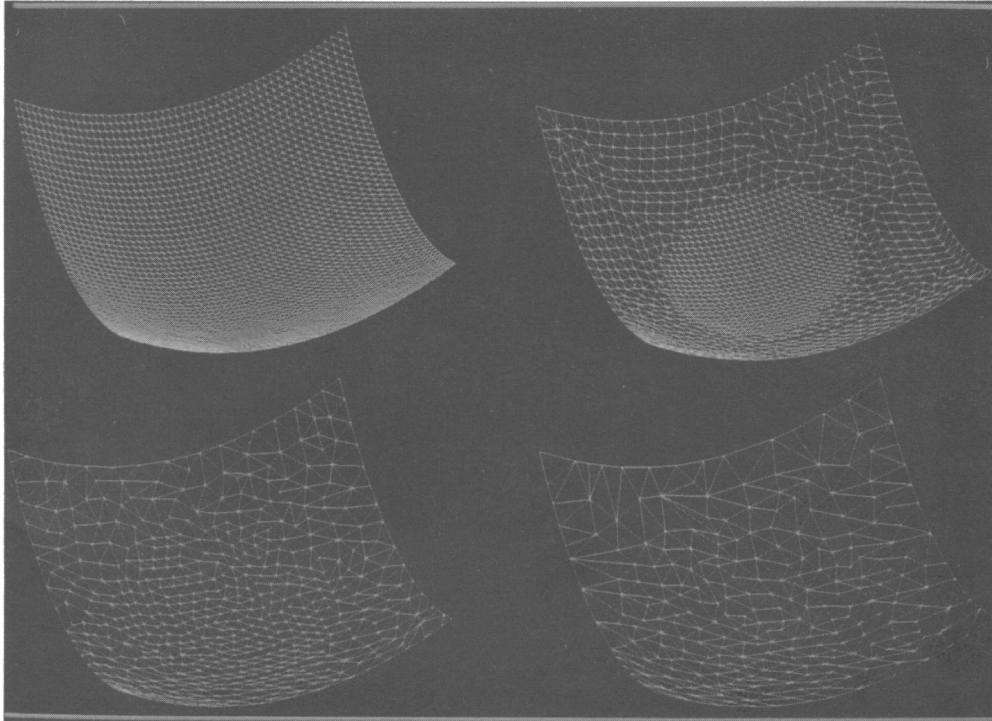


Fig. 8. Triangle reduction of 50%, 80%, and 90% for graph of $f(x, y) = .4(x^2 + y^2)$, $x, y \in [-1, 1]$.

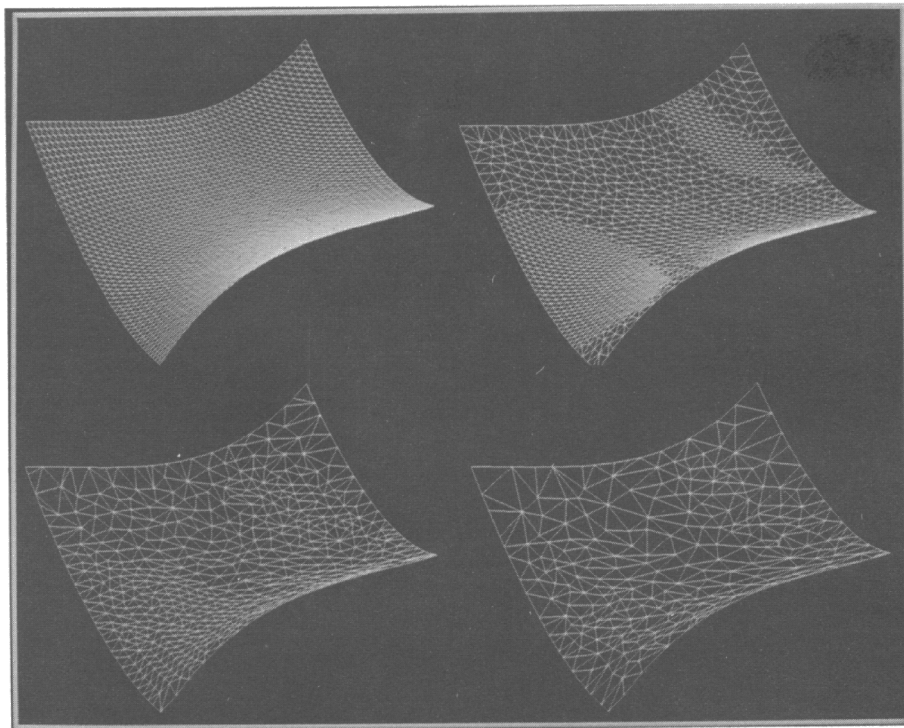


Fig. 9. Triangle reduction of 50%, 80%, and 90% for graph of $f(x, y) = .15(x^3 + 2x^2y - xy + 2y^2)$, $x, y \in [-1, 1]$.

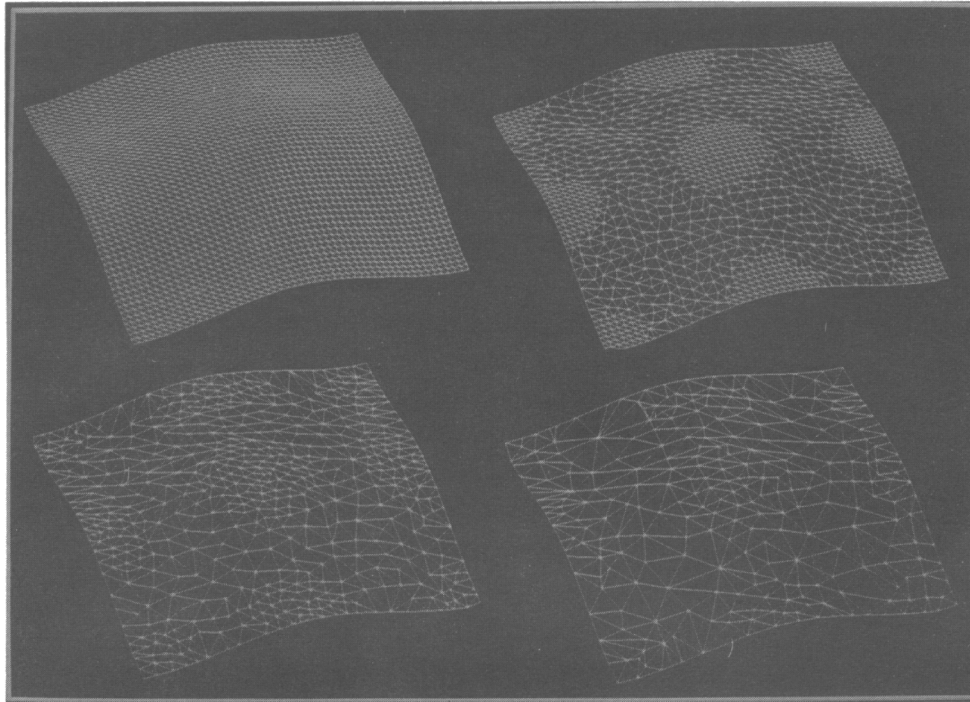


Fig. 10. Triangle reduction of 50%, 80%, and 90% for graph of $f(x, y) = .1(\cos(\pi x) + \cos(\pi y))$, $x, y \in [-1, 1]$.

In Table 1, the original and reduced triangulations are compared for different degrees of reduction. The number of triangles has been reduced by 50%, 80%, and 90%. During reduction it is ensured that the reduced triangulation still covers the whole domain $[-1, 1] \times [-1, 1]$.

The examples shown in Figs. 8–10 are based on the bivariate approach for replacing a triangle by a new point. Each figure shows the original (upper-left) and three reduced triangulations for different degrees of reduction (50% upper-right, 80% lower-left, and 90% lower-right). The functions chosen in these examples are the functions 3, 6, and 8 from the table above. The curvature input needed is pre-computed using the approximation scheme described in (Hamann, 1993a).

Fig. 11 shows the reduction algorithm applied to a torus, and Fig. 12 shows the original triangular approximation of a human skull (left, about 60,000 triangles) and the result after a reduction of 90% (right). All triangles are flat-shaded.

7. Conclusions

A new method for data reduction has been presented. This technique is based on an iterative triangle removal principle. The degree of reduction can be specified either by a percentage or, in the case of graphs of bivariate functions, by an error tolerance. The test results confirm the quality of this approach.

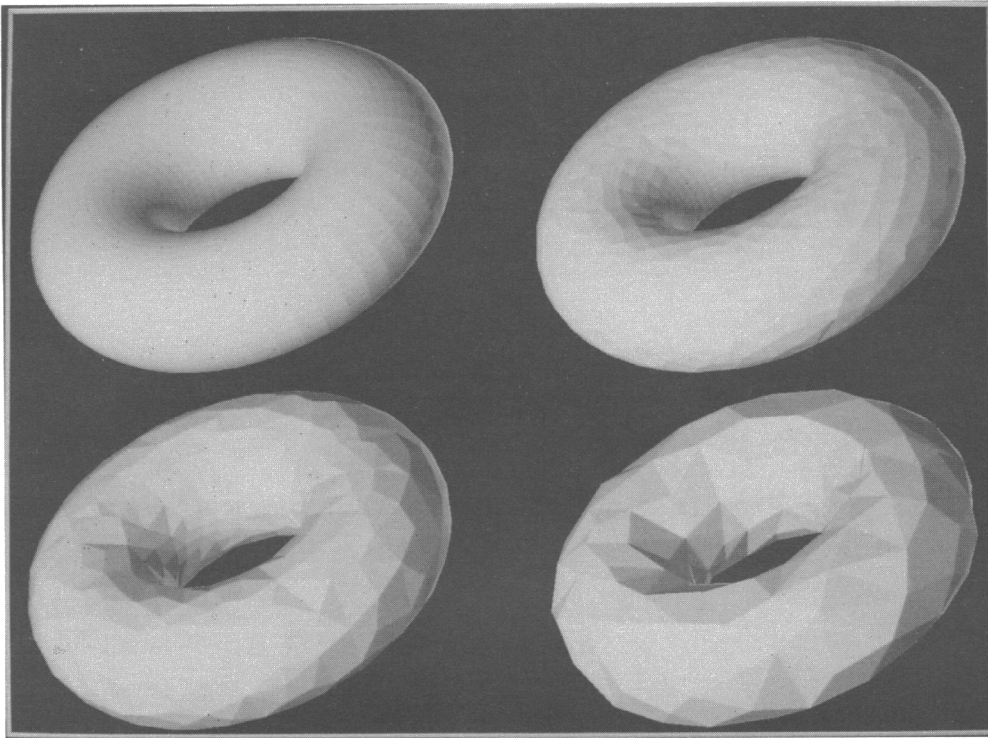


Fig. 11. Triangle reduction of 50%, 80%, and 90% for torus $((2 + \cos u) \cos v, (2 + \cos u) \sin v, \sin u)^T, u, v \in [0, 2\pi]$.

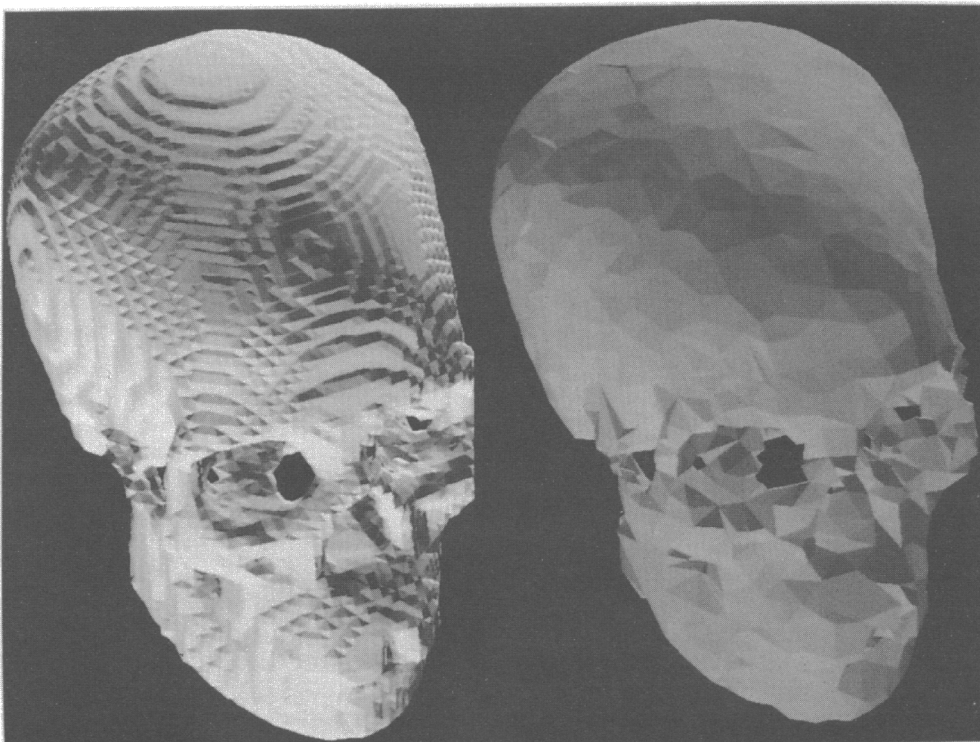


Fig. 12. Triangle reduction of 90% for piecewise triangular approximation of human skull.

The method can be extended to higher-dimensional manifolds as well, e.g., the graph of a trivariate function, whose domain is given as a collection of tetrahedra and whose graph lies in four-dimensional space. A curvature approximation scheme for such higher-dimensional manifolds has been introduced in (Hamann, 1993b). However, the reduction scheme has not been generalized yet to higher-dimensional triangulations.

8. Acknowledgements

The work presented was supported by the Department of Energy under contract DE-FG02-87ER25041 and by the National Science Foundation under contract DDM 8807747 to Arizona State University. I wish to thank all members of the Computer Aided Geometric Design research group in the Computer Science Department at Arizona State University for their helpful suggestions. The reviewers' comments have increased the clarity of this paper significantly.

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.
- Choi, B.K., Shin, H.Y., Yoon, Y.I. and Lee, J.W. (1988), Triangulation of scattered data in 3D space, *Computer-Aided Design* 20, 239–248.
- do Carmo, M.P. (1976), *Differential Geometry of Curves and Surfaces*, Englewood Cliffs, NJ.
- Dyn, N., Levin, D. and Rippa, S. (1988), Data dependent triangulations for piecewise linear Interpolation, *IMA J. Numer. Anal.* 10, 137–154.
- Dyn, N., Levin, D., and Rippa, S. (1990), 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*, Academic Press, San Diego, CA, 3rd edition.
- Hamann, B. (1993a), Curvature approximation for triangulated surfaces, in: Farin, G., Hagen, H., Noltemeier, H. and Knödel, W., eds., *Geometric Modelling*, Springer, Berlin, 139–153 (presented at: “First Dagstuhl Seminar on Geometric Modelling”, July 1–5, 1991, Dagstuhl, Germany).
- Hamann, B. (1993b), Curvature approximation for graphs of trivariate functions, *Computer Aided Geometric Design*, to appear.
- Lawson, C.L. (1977), Software for C^1 surface interpolation, in: Rice, J.R., ed., *Mathematical Software III*, Academic Press, New York, 161–194.
- 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, New York, 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. Anal.* 8, 185–208.
- O'Neill, B. (1969), *Elementary Differential Geometry*, Academic Press, New York, 3rd printing.