PERGAMON

Computers & Graphics 26 (2002) 657–663

COMPUTERS
&GRAPHICS

www.elsevier.com/locate/cag

Surface simplification using a discrete curvature norm

Sun-Jeong Kim[a], Chang-Hun Kim[a,*], David Levin[b]

[a] Department of Computer Science, Korea University, Seoul 136-701, South Korea
[b] Department of Applied Mathematics, Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract

This paper proposes a mesh simplification algorithm using a *discrete curvature norm*. Most of the simplification algorithms are using a distance metric to date. The distance metric is very efficient to measure geometric error, but it is difficult to distinguish important shape features such as a high-curvature region even though it has a small distance metric. We suggest a discrete curvature norm to measure geometric error for such features. During simplification the new vertex resulted from an edge collapse takes a position using a butterfly subdivision mask to minimize geometric error. This paper shows that simplification results have smaller geometric errors than previous works, when a discrete curvature norm and a distance metric are together applied to its criterion. © 2002 Published by Elsevier Science Ltd.

Keywords: Surface simplification; Multiresolution modeling; Edge collapse; Level of detail; Discrete curvatures

1. Introduction

In real-time or interactive applications, 3D polygonal models with millions of polygons are burdening some even with fast graphics hardware. Therefore mesh simplification has been the subject of a great deal of research. Simplification is the act of transforming a 3D polygonal model into a simpler version. A good simplification algorithm reduces the number of polygons while trying to retain the good approximation of the original shape and appearance. Polygonal models reconstructed from range-scanning systems may also be represented in several levels of details (LOD) in addition to being compressed and simplified. LOD can be built easily by consecutive simplifications with different error metrics.

In spite of extensive use of triangle meshes in geometric modeling and computer graphics, there is no agreement on the most appropriate way to estimate simple geometric attributes such as curvatures on discrete surfaces. Many surface-oriented applications need an approximation of the first- and second-order differential properties. Unfortunately, since meshes are piecewise linear surfaces, the concept of continuous curvatures is not common. For example, most simplification algorithms use the geometric distance [1,2] as their simplification criteria, which is another way to approximate local curvature [3]. But the discrete curvature can be computed by many schemes [4–6], and is useful to enhance the shape description of a triangular surface. Therefore, a discrete curvature is also one of the good criteria of simplification to preserve the shape of an original model.

Some simplification methods have used approximated curvatures as their criteria. In retiling [7], curvature is approximated to the radius of the largest sphere that is placed on the more curved side of the surface, but it is just the extension of 2D case into 3D. In data reduction scheme [8], it is proposed how to determine the principal curvatures and their associated directions by a least-squares parabolic fitting of the adjacent vertices, though the difficult task of selecting an appropriate tangent plane was left to the user. In static polyhedron simplification [9], the Gaussian curvature is used and error zone is defined with a sphere as error bound at each vertex. Like above, although the discrete curvature is useful for describing characteristics of polygonal model, it is rarely formalized as an error metric.

*Corresponding author. Tel.: +82-2-3290-3199; fax: +82-2-953-0771.

E-mail addresses: sunjeongkim@korea.ac.kr (S.-J. Kim), chkim@korea.ac.kr (C.-H. Kim), levin@math.tau.ac.il (D. Levin).

URL: http://cgvr.korea.ac.kr/

PII: S0097-8493(02)00121-8

An error metric is the difference between two polygonal models. Small error metric between two models means to be very similar to each other. Like a simplification criterion, the geometric distance has been used for defining an error metric between an original and a simplified model [1,10,11]. Some error metrics combine it with other attributes—color, normal, and texture coordinates—but they are too complex to be represented altogether [12,13]. Recently, other notions of an error metric, other than a geometric distance, are introduced. For example, image-driven simplification [14] defines an image metric, which is a metric based on pixel-wise differences between two images, and simplifies a mesh using image metrics between images from some views. In this paper we define a discrete curvature norm on meshes. By [5], we can approximate curvatures on a discrete surface reasonably, and be relieved from the difficulty of computation. A discrete curvature norm means the difference between discrete curvatures of an original and those of a simplified model. It can complement a distance metric for a high-curvature region with a small distance metric. Also the butterfly subdivision mask is used to minimize the distortion of discrete curvatures after collapsing an edge. When a discrete curvature norm and a distance metric are together applied to simplification criterion, the simplified results are closer to an original than those of previous works because they have smaller geometric error.

2. Problem statement

2.1. Notation

A triangular mesh M consists of a set of vertices $V = \{v_i\}_i \subset \mathbf{R}^3$, which are connected by a set of edges $E = \{e_j = (v_{j_1}, v_{j_2})\}_j$ and a set of faces $F = \{f_k = (v_{k_1}, v_{k_2}, v_{k_3})\}_k$. Let $v \in V$ be a vertex of a mesh M and let $v_1, \ldots, v_n$ be the ordered neighboring vertices of v (cf. Fig. 1). We define the edges $\vec{e}_i = v_i - v$ and the angle between two successive edges $\alpha_i = \angle(\vec{e}_i, \vec{e}_{i+1})$. The triangle between $\vec{e}_i$ and $\vec{e}_{i+1}$ is named $f_i = (v, v_i, v_{i+1})$, the corresponding face normal $\vec{n}_i = (\vec{e}_i \times \vec{e}_{i+1})/\|\vec{e}_i \times \vec{e}_{i+1}\|$. The *dihedral angle* at an edge $\vec{e}_i$ is the angle between the normals of the adjacent triangles, $\beta_i = \angle(\vec{n}_{i-1}, \vec{n}_i)$.

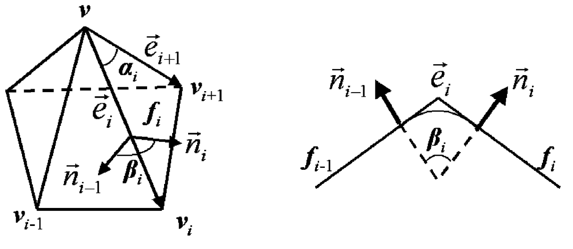


Fig. 1. A vertex v and the related variables for this local configuration (left). The blending cylinder along $\vec{e}_i$ between triangles f_{i-1} and f_i, seen from the side (right).

2.2. Simplification problem

Our goal is to simplify a polygonal model using consecutive edge collapses based on a discrete curvature norm. An edge collapse $(v_1, v_2) \to v$ (cf. Fig. 2) allows us to control the new vertex position for retaining the geometry of the original model. Therefore, the quality of our simplification result depends on how to assign the position of a new vertex v. If a new vertex takes a position to minimize geometric error between an original and a simplified model using an edge collapse, this simplified model will have good quality. Many previous works used a distance metric for finding such position. But with only a distance metric it is difficult to measure the variance of appearance such as simplifying high-curvature region with a small distance metric. A new metric is needed to measure geometric error for such region during surface simplification.

3. Discrete curvatures

3.1. Computation of discrete curvatures

From a theoretical point of view triangle meshes do not have any curvature at all, since all faces are flat and the curvature is not properly defined along edges and at vertices because the surface is not C^2 differentiable there. But thinking of a triangle mesh as a piecewise linear approximation of an unknown smooth surface, one can try to estimate the curvatures of that unknown surface using only the information that is given by the triangle mesh itself.

We are particularly interested in computing the sum of absolute *principal curvatures* $|\kappa_1|$ and $|\kappa_2|$ at the vertices of a mesh, since the cost to be minimized in simplification process is based on it. But, first let us derive the *Gaussian curvature K* and *mean curvature H*, and then get the sum of principal curvatures $|\kappa_1| + |\kappa_2|$ using them.

Gaussian curvature of a vertex is related to angles and faces that are connected to that vertex, and mean curvature is related to dihedral angles and edge lengths.

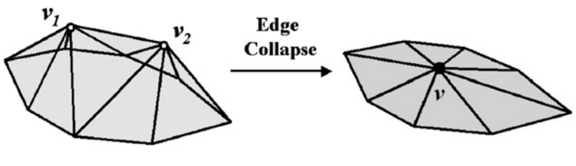


Fig. 2. Edge collapse $(v_1, v_2) \to v$.

We can define the *integral Gaussian curvature* $\bar{K} = \bar{K}_v$ and the *integral mean curvature* $\bar{H} = \bar{H}_v$ with respect to the area $S = S_v$ attributed to v by

$$\bar{K} = \int_S K = 2\pi - \sum_{i=1}^{n} \alpha_i \qquad \text{and}$$

$$\bar{H} = \int_S H = \frac{1}{4} \sum_{i=1}^{n} \|\vec{e}_i\| \beta_i,$$

where α_i is the angle between two successive edges, $\|\vec{e}_i\|$ is the length of the edge $\vec{e}_i$, and β_i is the dihedral angle of the edge $\vec{e}_i$ (cf. Fig. 1).

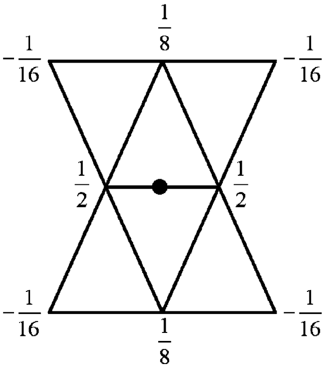


Fig. 3. The buttefly subdivision mask.

To derive the curvatures at the vertex v from these integral values we assume the curvatures to be uniformly distributed around the vertex and simply normalize by the area

$$K = \frac{\bar{K}}{S} = \frac{2\pi - \sum_{i=1}^{n} \alpha_i}{\frac{1}{3}A} \qquad \text{and}$$

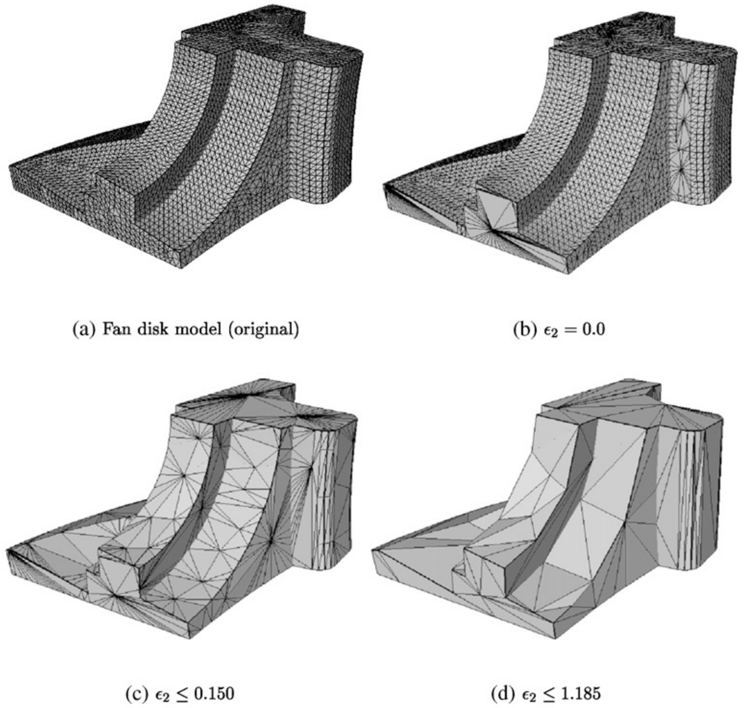
$$H = \frac{\bar{H}}{S} = \frac{\frac{1}{4}\sum_{i=1}^{n} \|\vec{e}_i\| \beta_i}{\frac{1}{3}A},$$

where A is the sum of the areas of adjacent faces around a vertex v.

At last the sum of the absolute principal curvatures $|\kappa_1|$ and $|\kappa_2|$ can be computed. From the relations $K = \kappa_1 \kappa_2$ and $H = (\kappa_1 + \kappa_2)/2$, we get $\kappa_{1,2} = H \pm \sqrt{H^2 - K}$. Therefore, the sum of the absolute principal curvatures is

$$|\kappa_1| + |\kappa_2| = \begin{cases} 2|H| & \text{if } K \geqslant 0, \\ 2\sqrt{H^2 - K} & \text{otherwise.} \end{cases}$$

Note that κ_1 and κ_2 are always a real number, even if $H^2 < K$, which corresponds to complex principal curvature values. Of course, this cannot happen for smooth surfaces, but since we are dealing with discrete surfaces it can occur for some vertices.



(a) Fan disk model (original)

(b) $\epsilon_2 = 0.0$

(c) $\epsilon_2 \leq 0.150$

(d) $\epsilon_2 \leq 1.185$

Fig. 4. Example of surface simplification using ε_2, the variance of a discrete curvature L_2-norms. Each about (b) 30%, (c) 92%, and (d) 98% simplification.

3.2. Discrete curvature norm

Now we can define a *discrete curvature norm*, that is the norm of principal curvatures at all vertices. We consider two kinds of discrete curvature norms, namely the L_1-norm and the L_2-norm.

$$\begin{aligned} \|\kappa\|_1 &= \sum_{v \in V} |\kappa_1| + |\kappa_2| \\ &= \sum_{v \in V} \begin{cases} 2|H_v| & \text{if } K_v \geqslant 0, \\ 2\sqrt{H_v^2 - K_v} & \text{otherwise.} \end{cases} \end{aligned} \quad (1)$$

$$\begin{aligned} \|\kappa\|_2^2 &= \sum_{v \in V} |\kappa_1|^2 + |\kappa_2|^2 \\ &= \sum_{v \in V} (4H^2 - 2K). \end{aligned} \quad (2)$$

During simplification, a discrete curvature norm is checked before and after an edge collapse. The difference between them means the variance of appearance and plays a role as an error metric in surface simplification. Note that our greedy simplification algorithm chooses first an edge that has the smallest quantity of difference between discrete curvature norms before and after collapsing it.

4. Simplification algorithm

4.1. Edge collapse with butterfly mask

An edge collapse allows the new vertex position to be controlled for retaining the geometry of the original model. Therefore, the quality of a simplified mesh depends on how to assign the position of a new vertex. To minimize the change of a discrete curvature norm after collapsing an edge, we decide the new vertex position using the butterfly subdivision mask (cf. Fig. 3), which is one kind of subdivision mask.

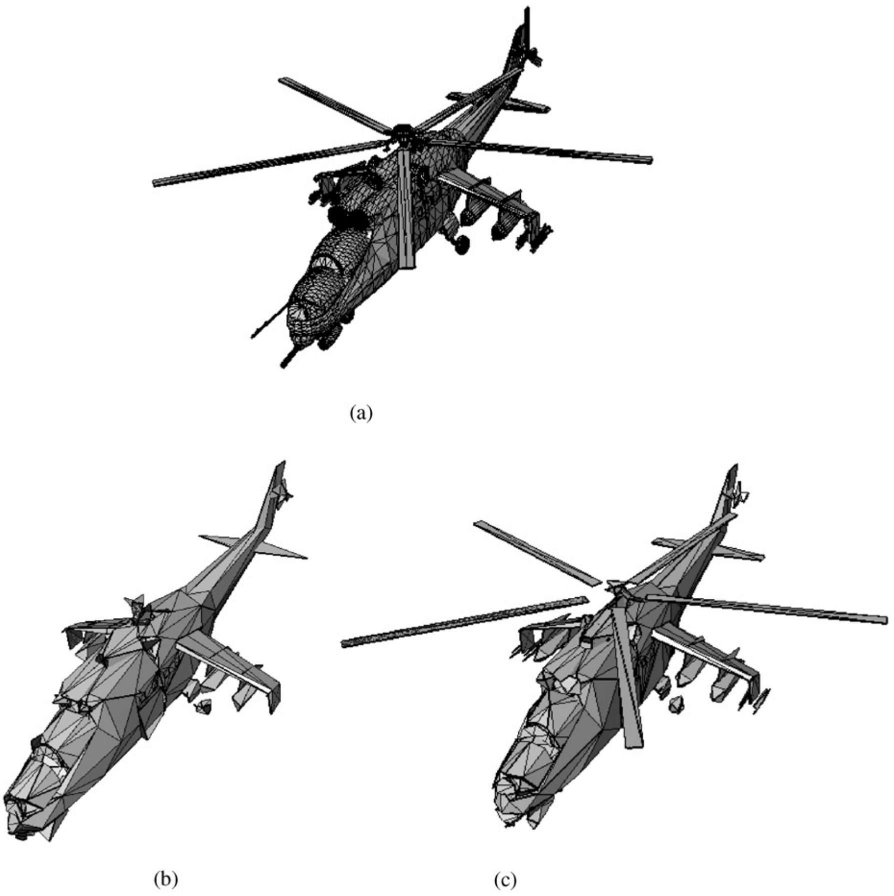


Fig. 5. Simplified results of a helicopter model: (a) an original (number of faces: 34,708), (b) using only a distance metric, and (c) using a distance metric and a discrete curvature norm (number of faces: 986).

4.2. Surface simplification

In our algorithm, the simplification cost is the quantity of the variance of a discrete curvature norm after an edge collapse. During simplification, it is minimized. The butterfly subdivision mask helps the position of a new vertex after an edge collapse to be assigned to keep the geometry of the original. Using a discrete curvature norm and the butterfly rule, the surface simplification algorithm is as follows:

- Our simplification algorithm reads an input original mesh.
- It calculates the discrete curvatures for all vertices and the simplification costs of all edges, which are the quantities of variance of a discrete curvature norm using Eq. (1), or (2) after an edge collapse.
- In ascending order of the simplification cost, the priority queue is constructed for all edges. Therefore there is always the edge with the smallest quantity of variance of a discrete curvature norm on the top of the priority queue.
- If the simplification cost of the edge on the top of the priority queue is smaller than a user-given value, then our algorithm collapses this edge. Otherwise, it ends the simplification process, since all other simplification costs in the priority queue are larger than it.
- After the edge is collapsed, the discrete curvatures of its 1-neighbor vertices and the simplification costs of its 2-neighbor edges are updated. Three edges are removed from the priority queue, and the positions of 2-neighbor edges in the priority queue are updated.

We have tested a new metric on many different polygonal models. Fig. 4 shows the results of surface simplification with respect to various discrete curvature L_2-norms. Many edge collapses were happened within no variance of a discrete curvature norm. In other words, Fig. 4(b) has the same shape as the original Fig. 4(a) although it is reduced by about 30%. Also, the simplification results using discrete curvature L_1 and L_2 norms are similar to each other.

We experimented with applying a discrete curvature norm to surface simplification together with a distance metric [1] (cf. Figs. 5 and 6). Compared with Fig. 5(b), which is a simplified result using only a distance metric, our simplified result Fig. 5(c) kept thin and long propellers. Because the vertices on a propeller have small distance metrics but at the same time large discrete curvature norms, when only a distance metric is used for surface simplification, a propeller is disappeared earlier than when a discrete curvature norm is used together. Therefore, our scheme has the advantage of keeping such sharp and thin features like Fig. 5(c). As in Fig. 6, there is no obvious difference for a model that does not have thin and long features. But a little more triangles lie on the region where high-curvature vertices are Fig. 6(c) shows that the region of a forehead has fewer triangles than Fig. 6(b), but those of eyes, a nose and a lip have a little more triangles.

Fig. 7 shows the mean geometric deviation between an original and a simplified model measured by Metro [15]. Graphs say that it results in smaller geometric error to use a discrete curvature norm with a distance metric than only a distance metric.

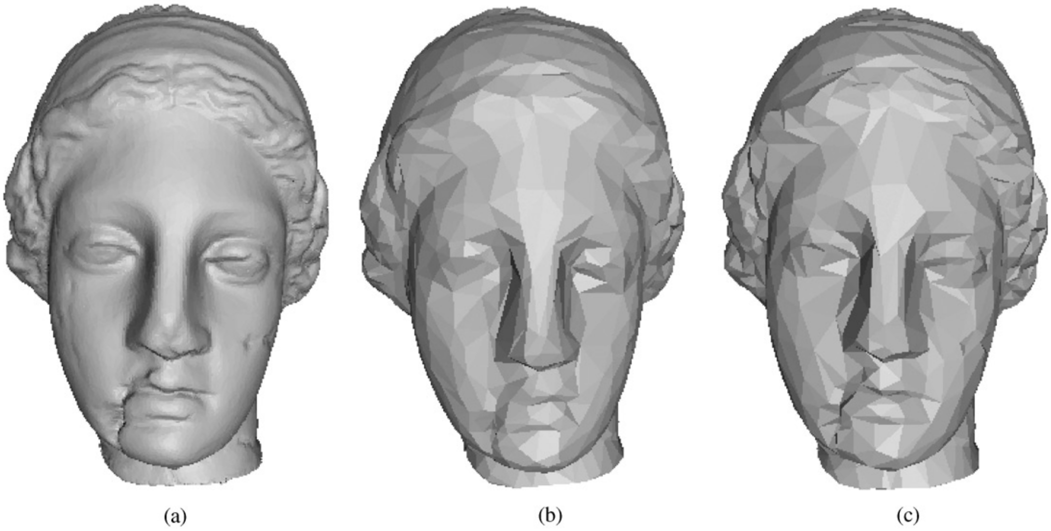
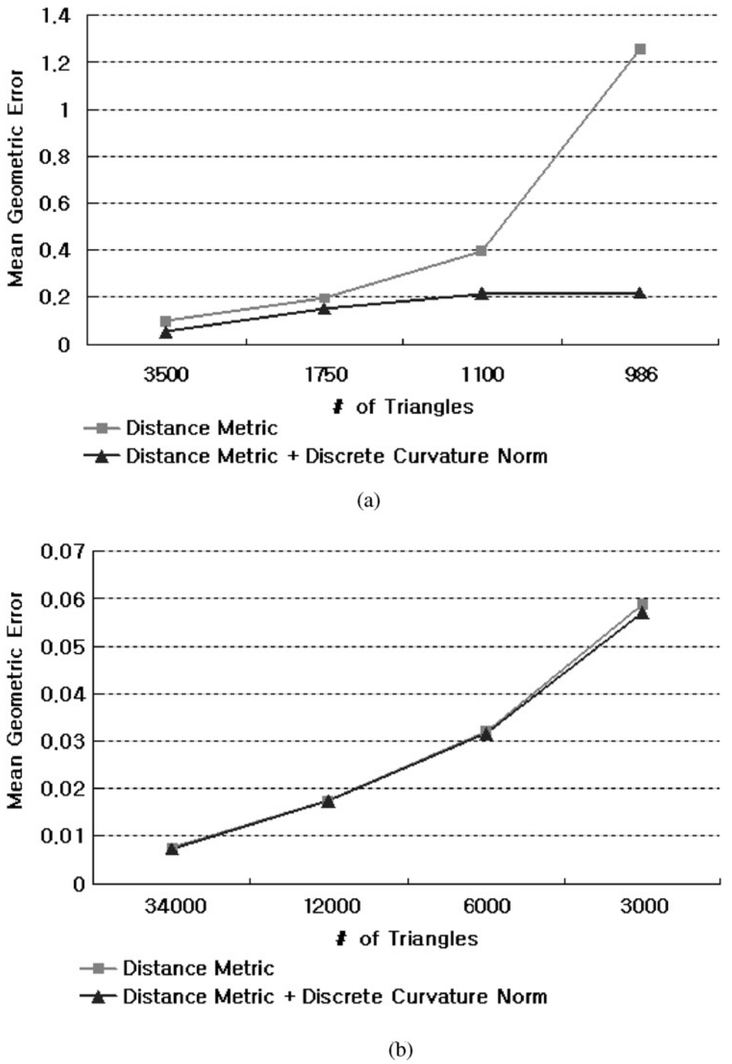


Fig. 6. Simplified results of an Igea model: (a) an original (number of faces: 268,686), (b) using only a distance metric, and (c) using a distance metric and a discrete curvature norm (number of faces: 3,000).



(a)

(b)

Fig. 7. Mean geometric error measured by Metro [1]: (a) Helicopter model and (b) Igea model.

5. Conclusion

A discrete curvature norm is proposed for surface simplification. It is efficient to measure geometric error for a high-curvature region even though it has a small distance metric. Also after an edge collapse a new vertex takes a position using a butterfly subdivision mask to minimize geometric error. We show that simplification results have smaller geometric error than previous works, when a discrete curvature norm and a distance metric are together applied to its criterion.

In future work, we need to approximate curvatures of vertices appropriately in non-differentiable regions. At some vertices because the square of mean curvature is smaller than the Gaussian curvature, principal curvature becomes a complex number. Also we have to find the optimal position of a new vertex resulting from an edge collapse with respect to discrete curvatures.

References

[1] Garland M, Heckbert P. Surface simplification using quadric error metrics. Proceedings of SIGGRAPH 1997, August 1997. p. 209–16.

[2] Schroeder WJ, Zarge JA, Lorensen WE. Decimation of triangle meshes. Computer Graphics (Proceedings of SIGGRAPH, July 1992) 1992;26(2):65–70.

[3] Heckbert P, Garland M. Optimal triangulation and quadric-based surface simplification. Journal of Computational Geometry: Theory and Applications 1999;14(1–3): 49–65.

[4] Desbrun M, Meyer M, Schröder P, Barr A. Discrete differential-geometry operators in nD, submitted for publication.

[5] Dyn N, Hormann K, Kim S-J, Levin D. Optimizing 3D triangulations using discrete curvature analysis. Mathematical Methods for Curves and Surface 2001;135–46.

[6] Meek DS, Walton DJ. On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. Computer Aided Geometric Design 2000;17:521–43.

[7] Turk G. Re-tiling polygonal surfaces. Computer Graphics (Proceedings of SIGGRAPH, July 1992) 1992;26(2):55–64.

[8] Hamann B. A data reduction scheme for triangulated surfaces. Computer Aided Geometric Design 1994;11(2): 197–214.

[9] Veron P, Leon JC. Static polyhedron simplification using error measurements. Computer-Aided Design 1997;29(4): 287–98.

[10] Cohen J, Varshney A, Manocha D, Turk G, Weber H, Agarwal P, Brooks F, Wright W. Simplification envelopes. Proceedings of SIGGRAPH 1996, August 1996. p. 119–28.

[11] Klein R, Liebich G, Straßer W. Mesh reduction with error control. IEEE Visualization 1996, October 1996. p. 311–8.

[12] Garland M, Heckbert P. Simplifying surfaces with color and texture using quadric error metrics. IEEE Visualization 1998, 1998;263–70.

[13] Hoppe H. New quadric metric for simplifying meshes with appearance attributes. IEEE Visualization 1999, October 1999. p. 59–66.

[14] Lindstrom P, Turk G. Image-driven simplification. ACM Transaction on Graphics 2000;19(3):204–41.

[15] Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplification surfaces. Computer Graphics Forum 1998;17(2):167–74.