Discrete Differential Error Metric for Surface Simplification

Sun-Jeong Kim, Soo-Kyun Kim, and Chang-Hun Kim
Department of Computer Science and Engineering
Korea University
1, 5-ka, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea
{sunjeongkim, nicesk, chkim}@korea.ac.kr

Abstract

In this paper we propose a new discrete differential error metric for surface simplification. Many surface simplification algorithms have been developed in order to produce rapidly high quality approximations of polygonal models, and the quadric error metric based on the distance error is the most popular and successful error metric so far. Even though such distance based error metrics give visually pleasing results with a reasonably fast speed, it is hard to measure an accurate geometric error on a highly curved and thin region since the error measured by the distance metric on such a region is usually small and causes a loss of visually important features.

To overcome such a drawback, we define a new error metric based on the theory of local differential geometry in such a way that the first and the second order discrete differentials approximated locally on a discrete polygonal surface are integrated into the usual distance error metric. The benefits of our error metric are preservation of sharp feature regions after a drastic simplification, small geometric errors, and fast computation comparable to the existing methods.

1. Introduction

In realtime or interactive applications, 3D polygonal models with millions of polygons are burdensome even on fast graphics hardware. Therefore polygonal surface simplification has been one of the main subjects of a great deal of research. Simplification is the act of transforming a 3D polygonal model into a simpler version. It must reduce the number of polygons while trying to retain the good approximation of the original shape and appearance. Simplification algorithms can be divided into some categories according to the strategies employed: vertex decimation [16], vertex clustering [14], edge contraction [5, 8, 9, 11], and face constriction [19].

In spite of extensive use of 3D polygonal models in geometric modeling and computer graphics, there is no agreement on the most appropriate way to estimate simple geometric attributes such as curvatures on polygonal surfaces. Many surface-oriented applications need an approximation of the first and the second order differential properties. Unfortunately, since 3D polygonal models are piecewise linear surfaces, the concept of continuous curvatures is not common. Most simplification algorithms use a geometric distance metric [2, 5, 10, 16] as their simplification criteria. But it is hard to measure an accurate geometric error for the high curvature and thin region that has a small distance error metric. Recently the discrete curvature can be computed by many schemes [3, 4, 13], and is useful to enhance the shape description of polygonal surfaces. Therefore, discrete differential metrics are also good criteria of simplification to preserve the appearance of an original model.

Some simplification methods have approximated discrete differential metrics, especially curvatures as their criteria. In retiling [17], 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 [7], it is proposed how to determine the principal curvatures and their associated directions by a least-squares paraboloid fitting of the adjacent vertices, though the difficult task of selecting an appropriate tangent plane was left to the user. In static polyhedron simplification [18], the Gaussian curvature is also used and error zone is defined with a sphere as error bound at each vertex. Like above, although discrete curvature is useful for describing characteristics of polygonal model, it is rarely formalized into an error metric.

The error metric is a measure of difference between two polygonal models. Small error between two models means to be very similar to each other. An error metric is usually defined as the geometric distance between an original and a simplified model [2, 5, 10, 15]. Some error metrics combine other attributes – color, normal, and texture coordinator – but these methods are too complex to be represented altogether [6, 9]. Recently an error metric is defined in different ways instead of the geometric distance. In memoryless simplification [11] an error metric is based on geometric properties of the mesh such as volumes and areas. Image-driven simplification algorithm [12] 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 several views.

In this paper, we propose a new discrete differential error metric for surface simplification. Our simplification algorithm uses iterative edge collapses, which can control the position of new vertex in order to retain the geometry of the original model, and do not need a re-triangulation. Since a distance error metric and the first and the second order discrete differentials are integrated into our new error metric, our new discrete differential error metric works better than a previous quadric error metric in finding a position of new vertex that minimizes a geometric error. As well, our simplification is as fast as the previous method because additional computation is done only in the pre-processing stage.

Contributions

- We define a new discrete differential error metric on surfaces. Since polygonal surfaces may be considered as piecewise linear approximation of unknown smooth surfaces, their tangent planes and discrete curvatures can be estimated.

- Using our new discrete differential error metric, surface simplification algorithm can produce rapidly higher quality approximations of polygonal models than previous works because the new discrete differential error metric includes a distance error metric and the first and the second discrete differentials, and additional computation is done only during pre-processing.

2. Problem Statement

The goal of surface simplification is to generate an approximation of the original rapidly. An error metric can control the quality of simplified models. Previous work [5] has shown that a quadric error metric allows fast and accurate geometric simplification of meshes, but this metric has a weak point for a highly curved and thin region because its error is computed using plane equations. We want to solve this drawback by adopting a theory of local differential geometry into our error metric.

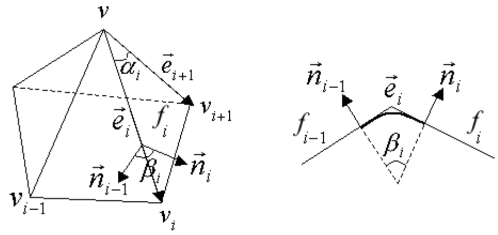


Figure 1. A vertex v and the related variables for this local configuration (left). The blending cylinder along $\vec{e}$ between faces f_{i-1} and f_i, seen from the side (right).

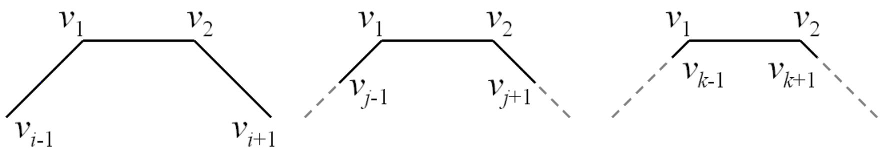
2.1. Notation

A polygonal surface 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 $F = \{f_k = \Delta(v_{k_1}, v_{k_2}, v_{k_3})\}_k$. Let $v \in V$ be a vertex of a polygonal surface M and let $v_1, \ldots, v_n$ be the ordered neighboring vertices of v (cf. Fig 1). Each vertex v has a geometric position $\mathbf{p} \in \mathbf{R^3}$. 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 triangular face between $\vec{e}_i$ and $\vec{e}_{i+1}$ is named $f_i = \Delta(v, v_i, v_{i+1})$, 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 faces, $\beta_i = \angle(\vec{n}_{i-1}, \vec{n}_i)$. For each vertex v, the vertex normal $\vec{n}_v$ is defined by $\vec{n}_v = \sum_{f_i \ni v} \vec{n}_i$, and the tangent plane P_v is orthogonal to $\vec{n}_v$. Note that in these definitions we identify the index 0 with n and the index $n+1$ with 1.

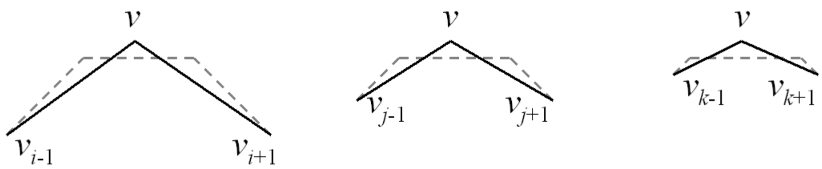
2.2. Problem

Our simplification algorithm uses iterative edge collapses since an edge collapse does not need a re-triangulation. Besides, it can control the position of a new vertex in order to retain the geometry of the original. The key point to produce high quality approximation is how to assign its position that minimizes the geometric error.

Many previous simplification algorithms use a distance error metric in order to find the optimal position of a new vertex after an edge collapse, and a point-to-plane distance that is used for computing a distance error is the one of its weak points. In Fig. 2(a), there are three cases where the same edge (v_1, v_2) is on different polygonal shapes. Their shapes are definitely different but the plane equations of the neighbor faces of this edge are same. If this edge is collapsed using the distance error metric only, each of the three cases will result in Fig. 2(b) respectively. Because the plane equations of the three cases are same, the positions of new



(a) Three polygon shapes before an edge collapse $(v_1, v_2) \to v$, where all plane equations are same.



(b) After an edge collapse $(v_1, v_2) \to v$, where all new vertex v's positions and distance error metrics are same.

Figure 2. Seen from the side, although their shapes are obviously different, they have the same distance error metric because of the same plane equations.

vertex v in three cases will be also same. For that reason, all of the three cases will have the same values of a distance error metric. That means, even though their shapes are obviously different, they are simplified altogether within the same error metric regardless of their different shapes. This is mainly caused by using only a distance error metric for surface simplification. Therefore our new error metric is invented in order to assign different error metrics for such three cases, since they have obviously different shapes.

Based on local differential geometry, the vertex v_1 and v_2 in each case of Fig. 2(a) have different differentials, especially discrete curvatures, from the vertex v_1 and v_2 in other case. Therefore this paper proposes new concept of a discrete differential error metric for polygonal surface simplification, which is the unification of a distance, a tangential, and a discrete curvature error metrics.

3. Discrete Differential Error Metric

Discrete Differential Error Metric (DDEM) is newly defined since it is hard to measure an accurate geometric error for a highly curved and thin region using only a distance error metric. Because that region gathers planes close, the small value of a distance error metric can be approximated even though the region is conspicuous. Defined by combining a distance error metric with a tangential and a discrete curvature error metrics, DDEM can measure more accurate geometric error. Therefore surface simplification by using our new error metric can maintain the high curvature features, which are usually removed by using only a distance error metric.

DDEM is defined by unifying a distance error metric and the first and the second order discrete differentials, which become a tangential and a discrete curvature error metrics:

$$DDEM(v) = f(v) + f'(v) + f''(v)$$

where $f(v) = Q^v(\mathbf{p})$ is a distance error function, $f'(v) = T^v(\mathbf{p})$ is a tangential error function, and $f''(v) = C^v(\mathbf{p})$ is a discrete curvature error function. After an edge collapse, the new vertex is assigned to the position that minimizes $DDEM(v)$, and the edge which will be collapsed next is chosen as the one from the top of the priority queue that has the lowest value of $DDEM(v)$.

3.1. Distance error metric

DDEM uses a quadric form[5] as its distance error metric. The quadric error metric is based on weighted sums of squared distances. It defines on each face f of the original mesh a quadric $Q^f(\mathbf{p})$ equal to squared distance of a point $\mathbf{p} \in \mathbf{R}^3$ to the plane containing the face f. $Q^f(\mathbf{p})$ is defined as the distance in $\mathbf{R}^3$ from $\mathbf{p}$ to plane of face (v_1, v_2, v_3). Each vertex v of the original mesh is assigned the sum of quadrics on its adjacent faces weighted by face area is assigned:

$$Q^v(\mathbf{p}) = \sum_{f \ni v} area(f) \cdot Q^f(\mathbf{p}).$$

Let us now derive $Q^f(\mathbf{p})$ for a given face $f = (v_1, v_2, v_3)$. The signed distance of $\mathbf{p}$ to the plane containing f is $\vec{n}^T\mathbf{p} + d$, where the face normal $\vec{n} = (\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)/\|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\|$ and the scalar $d = -\vec{n}^T\mathbf{p}_1$. As an aside, a different formulation is to obtain these parameters by solving the linear system,

$$\begin{pmatrix} \mathbf{p}_1^T & 1 \\ \mathbf{p}_2^T & 1 \\ \mathbf{p}_3^T & 1 \end{pmatrix} \begin{pmatrix} \vec{n} \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

with the additional constraint that $\|\vec{n}\| = 1$.

Therefore, the squared distance between point $\mathbf{p}$ and plane containing f is

$$Q^f(\mathbf{p}) = (\vec{n}^T\mathbf{p} + d)^2 = \mathbf{p}^T(\vec{n}\vec{n}^T)\mathbf{p} + 2d\vec{n}^T\mathbf{p} + d^2,$$

which can be represented as a quadric functional $\mathbf{p}^T\mathbf{A}\mathbf{p} + 2\mathbf{b}^T\mathbf{p} + c$ where $\mathbf{A}$ is a symmetric 3×3 matrix, $\mathbf{b}$ is a column vector of size 3, and c is a scalar. Thus,

$$Q^f = (\mathbf{A}, \mathbf{b}, c) = (\vec{n}\vec{n}^T, d\vec{n}, d^2)$$

is stored using 6+3+1=10 coefficients. The advantage of this representation is that the quadric is obtained using simple linear combinations of these coefficient vectors.

At last, a quadric for the distance error metric of a new vertex v after an edge collapse $(v_1, v_2) \rightarrow v$ is assigned the sum of quadrics on its adjacent faces weighted by face area for vertex v_1 and v_2:

$$Q^v = \sum_{f \ni v_1} area(f) \cdot Q^f + \sum_{f \ni v_2} area(f) \cdot Q^f. \quad (1)$$

3.2. Tangential error metric

A tangential error metric is defined by the magnitude of a difference vector between two normal vectors of tangent planes. Assume that an edge (v_1, v_2) will be collapsed to v. Vertex v_1, v_2, and v have the tangent plane P_{v_1}, P_{v_2}, and P_v respectively. As definition, a tangential error metric is the sum of the magnitudes of difference vectors between two normal vectors $\vec{n}_{v_1}$ and $\vec{n}_v$ of tangent planes P_{v_1} and P_v, and two normal vectors $\vec{n}_{v_2}$ and $\vec{n}_v$ of tangent planes P_{v_2} and P_v.

$$T^v(\mathbf{p}) = T^{v_1}(\mathbf{p}) + T^{v_2}(\mathbf{p}) = \|\vec{n}_v - \vec{n}_{v_1}\| + \|\vec{n}_v - \vec{n}_{v_2}\|.$$

In case of $\|\vec{n}_v - \vec{n}_{v_1}\|$, the cosine law is applied to it as

$$\|\vec{n}_v - \vec{n}_{v_1}\|^2 = \|\vec{n}_v\|^2 + \|\vec{n}_{v_1}\|^2 - 2\|\vec{n}_v\|\|\vec{n}_{v_1}\|\cos\theta.$$

Since normal vectors are unit vectors, their magnitudes $\|\vec{n}_v\| = \|\vec{n}_{v_1}\| = 1$, and $\cos\theta$ is found out by $\vec{n}_v^T\vec{n}_{v_1}$. Therefore a tangential error metric for vertex v_1 is

$$\|\vec{n}_v - \vec{n}_{v_1}\|^2 = 1 + 1 - 2\cos\theta = 2(1 - \vec{n}_v^T\vec{n}_{v_1}).$$

At last, a tangential error metric $T^v(\mathbf{p})$ for a vertex v can be arranged by,

$$T^v(\mathbf{p}) = \sqrt{2}\sqrt{1 - \vec{n}_v^T\vec{n}_{v_1}} + \sqrt{2}\sqrt{1 - \vec{n}_v^T\vec{n}_{v_2}}. \quad (2)$$

3.3. Discrete curvature error metric

Let us explain how to derive the computation of discrete curvatures and then define a discrete curvature error metric. This metric is proposed in optimizing a 3D triangular mesh [4]. From a theoretical point of view polygonal surfaces do not have any curvature at all, since all polygonal 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 polygonal surface 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 polygonal surface itself.

We are particularly interested in computing the discrete Gaussian curvature K, the absolute discrete mean curvature $|H|$, and the sum of absolute discrete principal curvatures $|\kappa_1|$ and $|\kappa_2|$ at the vertices of the polygonal surface since our new error metric is based on these values.

Especially in case of principal curvatures, from the relations $K = \kappa_1\kappa_2$ and $M = (\kappa_1 + \kappa_2)/2$, we get $\kappa_{1,2} = H \pm \sqrt{H^2 - K}$. Note that $|\kappa_1| + |\kappa_2|$ is always a real number, even if $|H|^2 = H^2 < K$, which corresponds to complex principal curvature values. Of course, this cannot happen for smooth surfaces, but since we are dealing with discrete curvatures it can occur for some vertices.

The integral discrete Gaussian curvature $\bar{K} = \bar{K}_v$ and the integral absolute discrete mean curvature $|\bar{H}| = |\bar{H}_v|$ with respect to the area $S = S_v$ attributed to v is defined by

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

and

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

where α_i denotes the angle at a vertex, $\vec{e}_i$ is an edge of a vertex, and β_i is a dihedral angle (cf. Fig. 1).

To derive the discrete 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:

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

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.

Moreover, from the relations $K = \kappa_1\kappa_2$ and $M = (\kappa_1 + \kappa_2)/2$, we can get the sum of the absolute discrete principal curvatures without knowing H but only $|H|$:

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

Now we can define a discrete curvature error metric, which measures a certain difference of discrete curvatures between an original and a simplified polygonal model. The discrete curvature error metric is the variance value of discrete curvatures of neighbor vertices of a collapsed edge. Let a vertex $v_i, i = 1, \ldots, n$ be neighbor of a vertex v. A discrete curvature error metric $C^v(\mathbf{p})$ for a vertex v is defined by

$$C^v(\mathbf{p}) = \sum_{i=1}^{n} |C_{v_i} - C'_{v_i}|. \quad (3)$$

C_{v_i} can be chosen among the discrete Gaussian and the absolute discrete mean curvature, and the sum of absolute discrete principal curvatures of a vertex v_i, and C'_{v_i} means the discrete curvatures of a vertex v_i after an edge collapse.

4. Simplification Algorithm

Our simplification algorithm is based on iterative edge collapses and DDEM combined with Eq. (1), (2), and (3).

$$DDEM(v) = Q^v(\mathbf{p}) + T^v(\mathbf{p}) + C^v(\mathbf{p}) \quad (4)$$

As noted above, we can choose one discrete curvature error metric among the discrete Gaussian and the discrete absolute mean curvature, and the sum of absolute discrete principal curvatures. Starting with a polygonal model and choosing one discrete curvature error metric, we perform a simplification algorithm. For each edge collapse $(v_1, v_2) \rightarrow v$, simplification algorithm determines $DDEM(v)$ and then assigns the position $\mathbf{p}$ of a new vertex v minimizing $DDEM(v)$. An edge collapse with the lowest DDEM is chosen and actually carried out.

The algorithm can be quickly summarized as follows:

1. Compute the quadrics of DDEM by Eq. (4) for the initial polygonal model.

2. Compute the new vertex position $\mathbf{p}$ for each edge collapse $(v_1, v_2) \rightarrow v$. A discrete differential error metric of the vertex v becomes the cost of an edge collapse.

3. Place all the edge collapses in a heap with the edge of the minimum cost at the top.

4. Iterative remove an edge collapse of least cost from the heap, collapse this edge, and update the costs of all edge collapses involving v_1 and v_2.

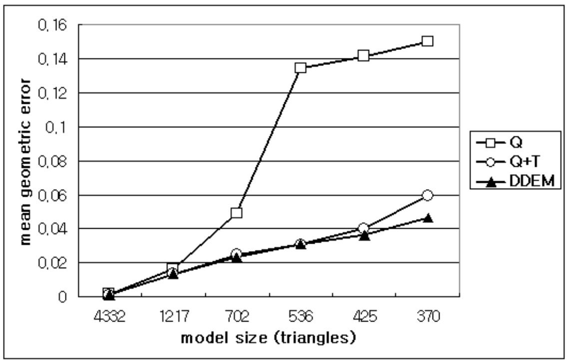
5. Implementation and Results

All models were simplified on a PC with Pentium III 866Mhz processor and 512MB of main memory. We have tested a number of simplified models, and used the Metro [1] in order to measure surface deviation after simplification. Metro accepts two polygonal models – an original and a simplified model – and computes mean geometric errors of the simplified model with respect to the original. This is done by point sampling on the simplified model uniformly, using Phong interpolation to estimate the surface normal at each sample, and intersecting the line defined by the point sample and its normal with the original model. Both the maximum and mean distances between the point samples and their corresponding intersections with the original are recorded.

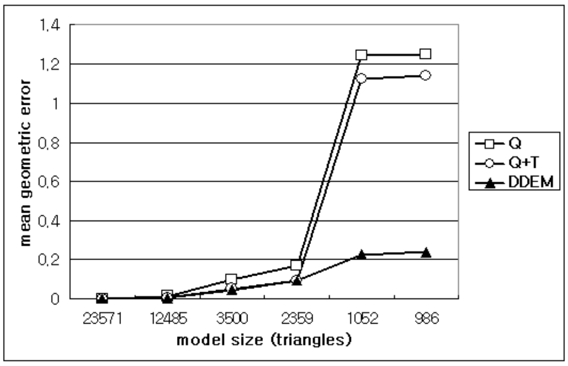
Fig. 3 shows the mean geometric deviations between the original and the simplified models. There is the comparison with three graphs representing geometric errors of simplified models using only a distance error metric, a distance and a tangential error metrics, and a DDEM. It can be seen from these graphs that the simplified results using a DDEM have small mean geometric errors compared with the errors resulted in by other error metrics.

Fig. 4 shows the several simplified steps. Fig. 4(a) is the example simplified using a DDEM, and Fig. 4(b) is simplified using quadric error metrics [5]. When simplified to 700 faces using quadric error metrics, the model lost one propeller. But our simplified result did not lose any propeller until it is simplified to 325 faces.

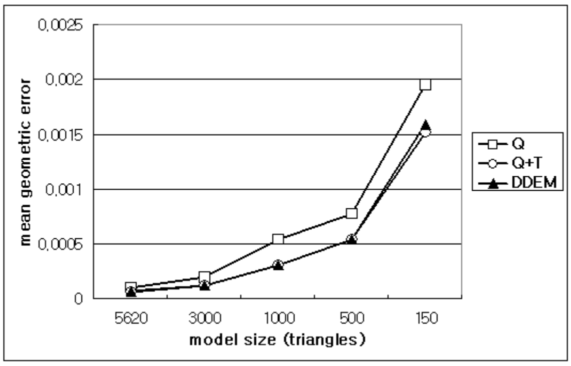
Fig. 5 shows the simplified results using different error metrics. Fig. 5(a) and Fig. 5(e) are the originals, and Fig. 5(b) $\sim$ 5(d) and Fig. 5(f) $\sim$ 5(h) are the simplified results each using only a distance error metric Q^v, a distance and a tangential error metrics $Q^v + T^v$, and a DDEM. Similar to Fig. 4(a), Fig. 5(d) preserved the features of the models such as the propellers and missiles. Compared with Fig. 5(f), 5(h) preserved better the silhouette of the back and the leg, and boundaries of the bottom.



(a) Cessna model



(b) Helicopter model



(c) Stanford bunny model

Figure 3. Mean geometric error measured by Metro [1]. Q**: distance error metric, and** T**: tangential error metric.**

Tab. 1 shows us the comparison with the pre-processing and the run times for simplification algorithm. Even though slightly more time is needed in the pre-processing step for combining several different metrics, it does not significantly impair the total running time.

6. Conclusion and Future Work

In this paper we proposed a new discrete differential error metric based on the theory of local differential geometry in such a way that the first and the second order discrete differentials approximated locally on a discrete polygonal surface are integrated into the usual distance error metric. Our new error metric overcomes the drawback of distance based error metrics such as quadric error metrics, with which it is hard to measure an accurate geometric on highly curved and thin region. The benefits of our error metric are preservation of sharp feature regions after a drastic simplification, small geometric errors, and fast computation comparable to the existing methods.

In future work, it would be desirable to measure the error metrics of attributes for surface simplification and prove theoretical analysis.

References

[1] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring Error on Simplification Surfaces. *Computer Graphics Forum*, 17(2):167-174, 1998.

[2] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright. Simplification Envelopes. *Proceedings of SIGGRAPH 96*, pp. 119-128, August 1996.

Table 1. Comparison of pre-processing and run time. (sec.)

	# of triangles		Simplified by Q^v		$Q^v + T^v$		DDEM	
Model	original	simp.	pre-pro.	run	pre-pro.	run	pre-pro.	run
Cessna	13,546	325	1.893	1.386	2.364	1.392	3.004	1.394
Helicopter	34,708	986	5.849	3.736	6.07	3.740	7.591	3.748
Stanford bunny	69,451	150	10.86	8.51	13.17	9.14	18.87	9.43

[3] M. Desbrun, M. Meyer, P. Schröder, and A. Barr. Discrete Differential-Geometry Operators in nD. *submitted for publication*.

[4] N. Dyn, K. Hormann, S.-J. Kim, and D. Levin. Optimizing 3D Triangulations Using Discrete Curvature Analysis. *Mathematical Methods for Curves and Surfaces: Oslo 2000*, Tom Lyche and Larry L. Shumaker (eds.), pp. 135-146, ISBN 0-8265-1378-6, 2001.

[5] M. Garland and P. Heckbert. Surface Simplification Using Quadric Error Metrics. *Proceedings of SIGGRAPH 97*, pp. 209-216, August 1997.

[6] M. Garland and P. Heckbert. Simplifying Surfaces with Color and Texture Using Quadric Error Metrics. *IEEE Visualization 98*, pp. 263-270, October 1998.

[7] B. Hamann. A Data Reduction Scheme for Triangulated Surfaces. *Computer Aided Geometric Design*, 11(2):197-214, April 1994.

[8] H. Hoppe. Progressive Meshes. *Proceedings of SIGGRAPH 96*, pp.99-108, August 1996.

[9] H. Hoppe. New Quadric Metric for Simplifying Meshes with Appearance Attributes. *IEEE Visualization 99*, pp. 59-66, October 1999.

[10] R. Klein, G. Liebich, and W. Straßer. Mesh Reduction with Error Control. *IEEE Visualization 96*, pp. 311-318, October 1996.

[11] P. Lindstrom and G. Turk, Evaluation of Memoryless Simplification. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):98-115, April-June 1999.

[12] P. Lindstrom and G. Turk. Image-Driven Simplification. *ACM Transaction on Graphics*, 19(3):204-241, July 2000.

[13] D. S. Meek and D. J. Walton. On Surface Normal and Gaussian Curvature Approximations Given Data Sampled From a Smooth Surface. *Computer Aided Geometric Design* 17, pp. 521-543, 2000.

[14] J. Rossignac and P. Borrel, Multi-Resolution 3D Approximations for Rendering Complex Scenes. *Modeling in Computer Graphics*, pp.s 455-465, 1993.

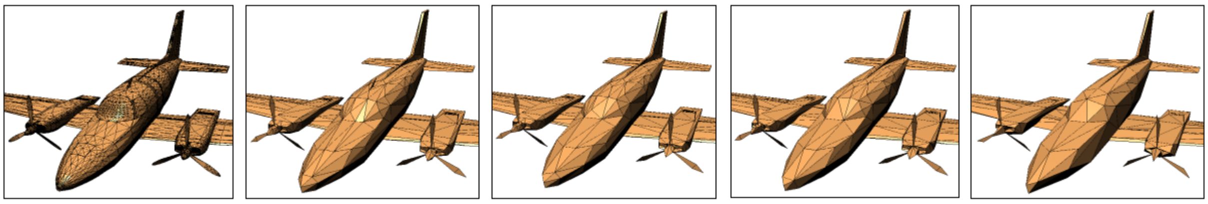
[15] W. J. Schroeder. A Topology Modifying Progressive Decimation Algorithm. *IEEE Visualization 97*, pp. 205-212, November 1997.

[16] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of Triangle Meshes. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):65-70, July 1992.

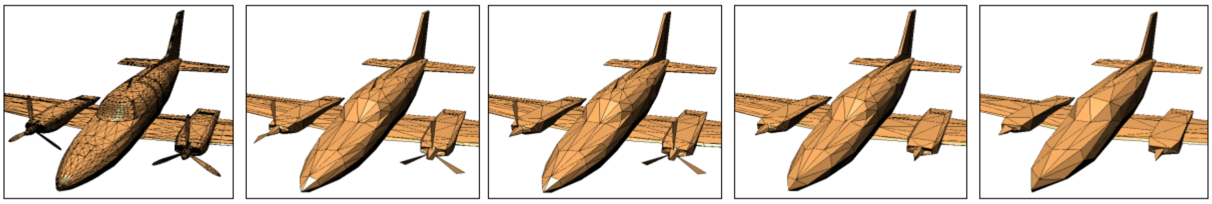
[17] G. Turk. Re-Tiling Polygonal Surfaces. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):55-64, July 1992.

[18] P. Veron and J. C. Leon. Static Polyhedron Simplification Using Error Measurements. *Computer-Aided Design*, 29(4):287-298, 1997.

[19] J.-H. Wu, S.-M. Hu, C.-L. Tai, and J.-G. Sun. An Effective Feature-Preserving Mesh Simplification Scheme Based on Face Constriction. *Proceedings of Pacific Graphics 2001*, pp. 12-21, October 2001.

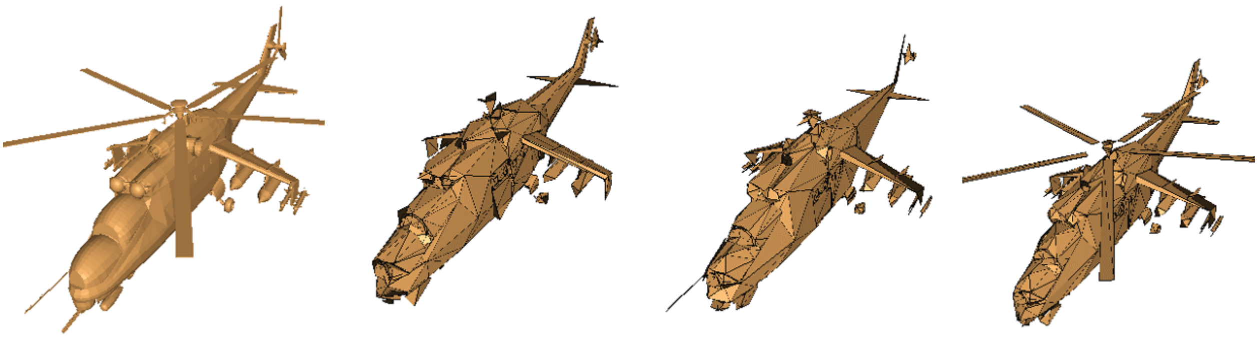


(a) Simplified using a DDEM

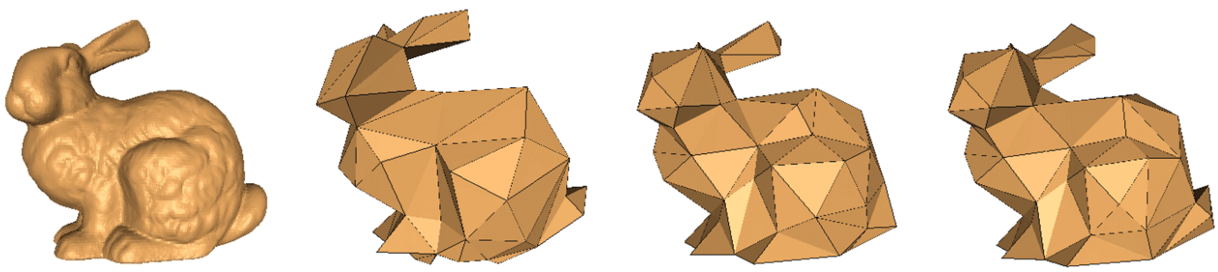


(b) Simplified by the QSlim [5]

Figure 4. Comparison with the proposed algorithm and the QSlim [5]. The leftmost column is the original(13,546 faces), the next is each simplified to 700 faces, 650 faces, 550 faces, and 325 faces.



(a) Original (34,708 faces) (b) Simplified using only Q^v (c) Using $Q^v + T^v$ (d) Using a DDEM



(e) Original (69,451 faces) (f) Simplified using only Q^v (g) Using $Q^v + T^v$ (h) Using a DDEM

Figure 5. Simplified results of the helicopter (986 faces) and the Stanford bunny (150 faces) models.