Journal of Engineering and Computer Innovations Vol. 2(6), pp. 98-105, June 2011
Available online at http://www.academicjournals.org/JECI
ISSN 2141-6508 ©2011 Academic Journals

Full Length Research Paper

A feature preserved mesh simplification algorithm

Jiacai Wang[1], Lirong Wang[2]*, Jinzhu Li[3] and Ichiro Hagiwara[1]

[1]Department of Mechanical Science and Engineering, Tokyo Institute of Technology, Japan.
[2]Institute of Advanced Integration Technology, Shenzhen Institute of Advanced Technology, Chinese Academy of Science, China.
[3]Insigma China, Beijing, China.

Accepted 12 April, 2011

Large-volume mesh model faces challenge in rendering, storing, and transmission due to large size of polygon data. Mesh simplification is one of solutions to reduce the data size. This paper presents a mesh simplification method based on feature extraction with curvature estimation to triangle mesh. The simplified topology preserves good geometrical features in the area with distinct features, that is, coarse simplified mesh in the flat region and fine simplified mesh around the areas of crease and corner. Sequence of mesh simplification is controlled on the basis of geometrical feature sensitivity, which results in reasonable simplification topology with less data size. This algorithm can decrease the size of the file by largely simplifying flat areas and preserving the geometric feature as well.

Key word: Mesh simplification, feature extraction, curvature estimation.

INTRODUCTION

Nowadays, product design based on reverse engineering is popularly due to the fast development of three-dimension (3D) measurement technology like 3D laser or CMM (coordinate-measuring machine) scanning systems. Measured representation of a 3D model is usually described in a million triangle mesh data. On the other hand, in traditional computer-aided design (CAD) modeling systems, high-level geometric primitive surfaces initially defined by versatile modeling operations like extrusion, constructive solid geometry and freeform deformations are usually be tessellated into lowest common form of polygonal mesh for display rendering. Complex triangle meshes face challenge in real-time rendering performance, storage capacities and transmission. In order to effectively operate complex mesh, typically triangular meshes, mesh simplification has emerged to dispatch meshes from a complex level to a simplified level (Li et al., 2005).

In recent years, automatic simplification to highly detailed mesh model has received increasing attention in the research field such as real-time rending of large-scale terrain, surface reconstruction in reverse engineering and rapid prototyping. Several kinds of simplification algorithms such as vertex or triangle decimation to remove vertices or triangle and re-triangulating the surrounding mesh, vertex clustering to condense vertices in certain cell into one vertex, vertex pair contraction and iterative edge contraction have been studied since the 1990s. Iterative edge collapse is a kind of simplification approach that tries to preserves volume geometric properties by collapse control. Progress mesh (PM) representation is a kind of iterative edge collapse based mesh simplification method developed by Hoppe (1996), in which an energy function to describe the complexity and fidelity of mesh is used to track simplification quality. This method requires many vertex distance evaluations and it reduces the computational speed. Other edge contraction based simplification, such as quadric error metrics (QEM) based simplification algorithm proposed by Garland (1997), use QEM to choose the edge to be simplified and the new vertex after contraction. Recent years some new mesh simplifications are developed. introduce a kind of simplification of surface mesh using Hausdorff envelope, in which two tolerance areas defined by an approximation of Hausdorff distance in percentage of the minimal bounding box size of the initial surface mesh are used to control mesh simplification and

*Corresponding author. E-mail: wanglr97@hotmail.com Tel: +86-755-86392143.

optimization in order to preserve the geometry of the surface. A mesh optimization algorithm based on neural network of Growing Neural Gas mode is presented to obtain a simplified set of vertices representing the best approximation of 3D object in (Noguera et al., 2009). Tang et al. (2007) develope a mesh simplification algorithm based on surface moments and volume moments defined to the original mesh to simplify mesh in an edge collapse scheme. Boubekeur and Alexa (2009) introduced a fast simplification algorithm based on stochastic vertex sampling. The stochastic sampling depends on local feature estimator of normal information, and mesh simplification starts from selecting vertices and re-indexing triangles to be clustered. Nate and Jihad (2010) proposed a novel top-down mesh simplification approach, in which simplification is carried out by a series of carving operation of iteratively removing tetrahedral to topological complex mesh. González et al. (2009) present a user-assisted mesh simplification method applied to triangle mesh converted from CAD models. This mesh simplification can be conducted to the whole models with a single level of detail, and some simplified parts can be modified or refined furtherly in different levels of detail according to user's demanding.

In this paper, feature extraction based mesh simplification is investigated. During simplification, edge-weight based on edge-collapse simplification is carried out, which is more efficient than energy minimization based on optimization mesh algorithm in PM introduced by Hoppe (1996). An algorithm to control simplification sequence is defined by an edge-weight using average curvature. As a result, less simplification is carried out on the surface part near to the sharp geometrical feature like corner; and the surface part without distinguished geometrical feature like flat surface will be largely simplified with large-scale mesh. This way, the geometrical feature can be preserved by ordering control of edge-collapse.

GEOMETRICAL FEATURE EXTRACTION

The estimating intrinsic geometric feature of polygonal mesh is an important stage in laser range scanning, scientific computing, computer vision, medical imaging and so on. There are two classified groups to extract geometric feature from polygonal mesh, volumetric-based method to utilize global characteristics of 3D object and boundary-based method to describe an abject based on distinct local properties of its boundary and their relationship. In boundary-based methods, curvature range image data. Comparison of Gaussian and mean estimation is important to features recognition, segmentation and registration to polygonal mesh. Great effort had been invested in computing curvature from realcurvature estimation methods on triangular mesh are investigated by Magid et al. (2007). A curvature estimation scheme for triangle meshes using biquadratic Bezier patche demonstrates to be good at dealing with irregular shape presented in Razdan and Bae (2005). In the computer vision literature, one can find some approaches for depth images that can handle features correctly and extract the feature lines. Guy and Medioni (1997) extract surfaces, feature lines and feature junctions from noisy point clouds, and space around the point cloud is discretized into volume grid. Gumhold et al. (2001) adopted the algorithm used by Guy and Medioni (1997) but avoided the discretization into a volume grid, in order to allow efficient handling of non-uniformly sampled point clouds. Kodani et al. (2003, 2004) investigated a method to select only points located on feature lines after classifying all points on feature lines like edges and tips appearing on smooth surface. The average curvature estimation presented in Kodani et al. (2003, 2004) was adopted by Wang et al. (2008) to extract feature point. In this research, the extraction of feature point is carried out on the basis of average curvature estimation.

In cloud point of a mesh, for each data point p_i, a set N_i of neighbor data points are gathered from its neighboring graph. The number of neighbor points depends on the point density and noise level of the dataset. The following two quantities are extracted from the set of neighboring points:

$$c_i = \frac{1}{|N_i|}\sum_{q=N_i} q \quad (1)$$

$$C_i = \sum_{q\in N_i}(q-c_i)(q-c_i)^T \quad (2)$$

where c_i is center location of the neighbor point set of q, and C_i is the correlation matrix.

The eigenvectors $\{e_0, e_1, e_2\}$ of the correlation matrix C_i and corresponding eigenvalues $\{\lambda_0, \lambda_1, \lambda_2\}$with $\lambda_0 \le \lambda_1 \le \lambda_2$ can be calculated by using Equation 2.

The average curvature can be estimated from the least square fitted plane given by(e_0, c_i). Figure 1 shows a two dimensional projection of the data point p with its tangent plane degenerated to the horizontal line. The distance of a point to its tangent plane is:

$$d = \left|e_0^t(p-c_i)\right| \quad (3)$$

Curvature estimation from fitted plane and average neighbor distance μ is shown in Figure 1. The curvature

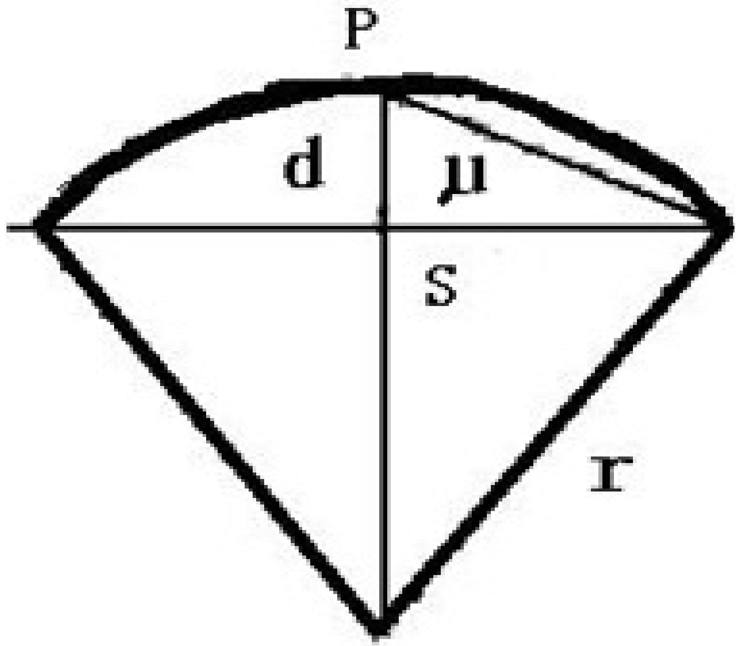


Figure 1. Curvature estimation.

radius, as shown in Figure 1, intersects the tangent plane at approximately the average neighbor distance μ . From $s^2 = \mu^2 - d^2$ and $s^2 = r^2 - (r-d)^2$, the curvature $k = 1/r$ computes to $2d/\mu^2$ and is a good criterion for detecting feature point of crease and corner points. Curvature estimation k_i for each data point p_i with distance d_i from the fitted plan can be defined as:

$$k_i = \frac{2d}{\mu^2} \quad (4)$$

By computing the maximum curvature estimation $k_{\max}$ of all data points, the normalized curvature is defined as:

$$w_{ki_i} = \frac{k_i(p)}{k_{\max}} \quad (5)$$

where w_{ki} is in the range of [0,1]. When w_{ki} near to 1, the possibility of that point to be considered as a crease on corner point is highly; and when w_{ki} near to 0, the neighboring region around that point is approaching to a flat plane. Here the corner points are regarded as the feature points. The extraction of feature points depends on the point density and data noise.

MESH SIMPLIFICATION

Geometrically, triangle mesh consists of a set of vertices. A set of triangles can be used to describe a piecewise linear surface by connecting subset of the vertices together in an order. In this research, edge collapse transformations are carried out on the initial mesh M^n, and a coarser approximation of M^0 is obtained after N operations of simplification, as described as $M^n \xrightarrow{Ecol_{n-1}} \cdots \xrightarrow{Ecol_1} M^1 \xrightarrow{Ecol_0} M^0$. Edge collapse transformation of $Ecol(V_t, V_s)$ to collapse the edge $V_t V_s$, as shown in Figure 2, unifies 2 adjacent vertices V_t and V_s into a new vertex V'_s. The vertex V_t and the two adjacent triangles of f_l and f_r vanish. Then the neighborhood faces around V'_s are rearranged, and the geometrical information of $(V_t, V_s, V_f, V_r, V'_s)$ is recorded as a sequence of vertex splitting record. The new point can located on the one position on edge $V_t V_s$ defined by user.

Sequence of edge collapse transformation, validity check of collapse transformation and curvature control are important influences on quality of the simplified approximating meshes in order to keep typical topology features of the original model and to achieve acceptable approximations as required. Edge collapse may potentially introduce undesirable inconsistency, degeneracy into the mesh, like holes or degenerated lines or vertex.

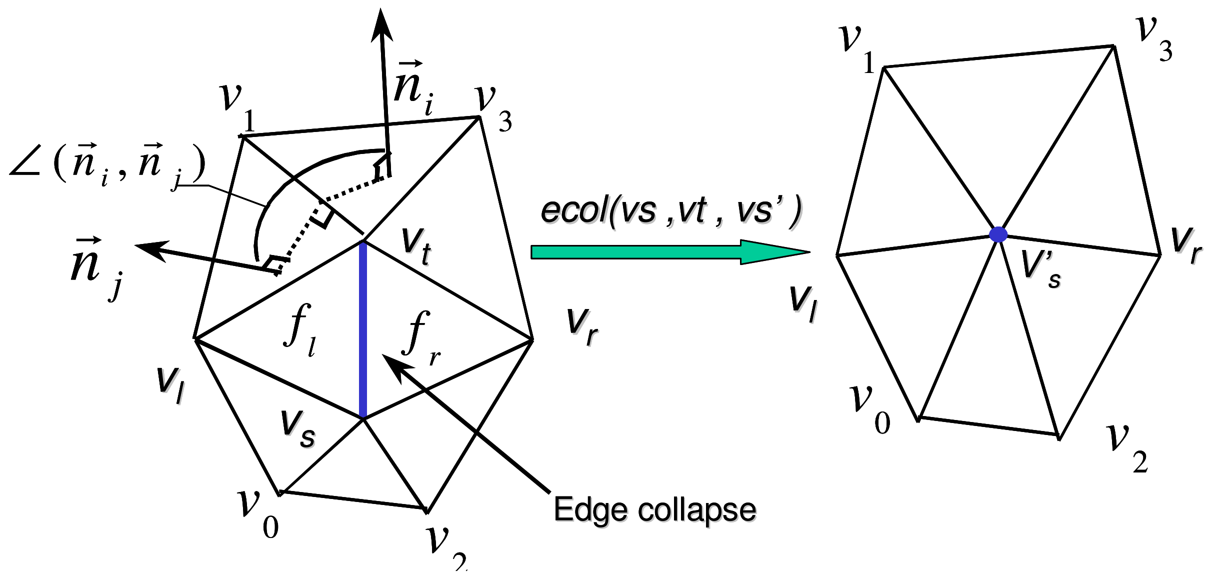


Figure 2. Collapsed edge based mesh simplification.

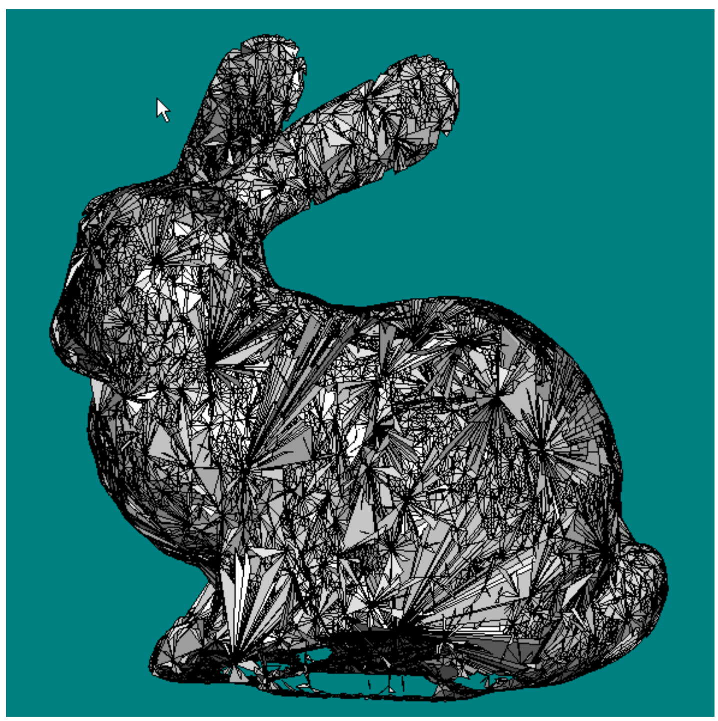


Figure 3. An example with no validity check.

As shown in Figure 3, the simplification result is very bad if there is no any simplification control during simplification (see comparison with Figure 5 (a).

In this research, sequence of edge collapse algorithms are investigated by random and edge weight methods. Edge weight based simplification is to define a weight to each edge, and the edge with the smallest weight will be collapsed firstly. Edge weight is defined as:

$$\omega_{t,s} = l_{t,s} \bar{w}_k \tag{6}$$

where $l_{t,s}$ is the Eulerian distance between vertex v_t and

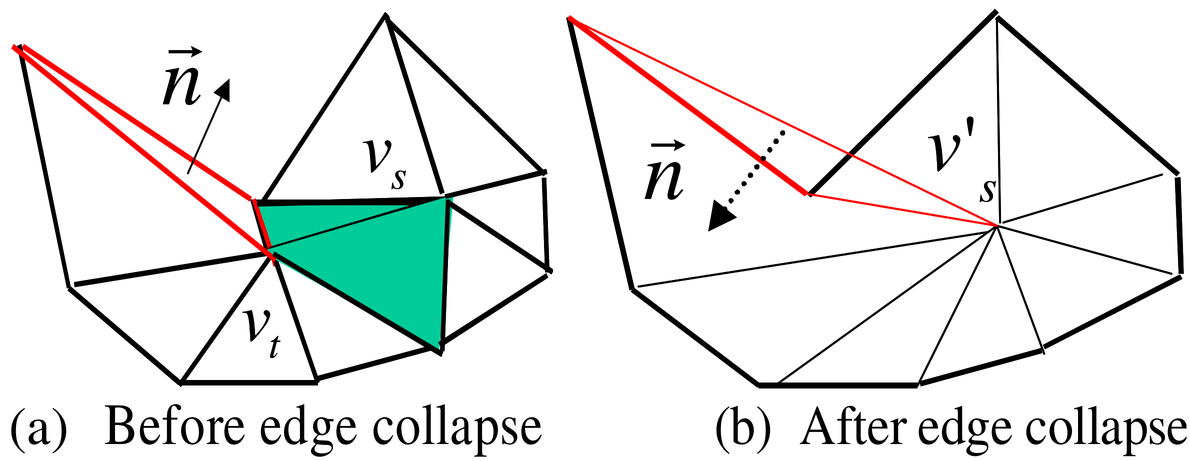


Figure 4. Normal direction inverse.

v_s of the collapse edge. w_k is the average normalized curvature of vertex v_t and v_s, that is:

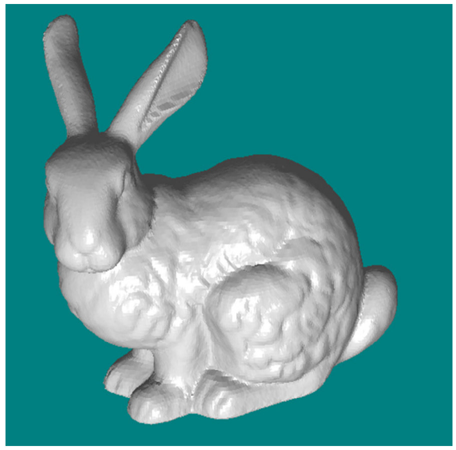
$$\bar{w}_k = (w_{kt} + w_{ks})/2 .$$

Three different validity checks are carried out: normal direction check, sliver triangle check and non-manifold edge control. As show in Figure 4, after edge collapse, the triangle with red side folds over onto itself, and the normal direction is inversed. This will result in a crease on the simplified model, regarded as flipped face. Sliver triangle control is conducted by check criteria of $Q = \frac{4\sqrt{3}\delta}{l_0^2 + l_1^2 + l_2^2}$, where δ is the triangle area after edge collapse, which can be calculated by $\delta = \sqrt{x(x-l_0)(x-l_1)(x-l_2)}$, l_0, l_1 and l_2 are the edge length of the triangle, $x = \frac{l_0 + l_1 + l_2}{2}$. When $Q \to 1$, the triangle approaches to an equiangular triangle; and when $Q \to 0$, the triangle is degenerated into a triangle whose vertices are collinear. Non-manifold edge control, is conducted by check criteria of $v_0 \neq v_1$ and $v_2 \neq v_3$, here v_0, v_1, v_2, v_3 are the vertices of the neighboring faces surrounding f_l and f_r, as shown in Figure 2. In this research, the faces adjacent to discontinuous boundary are prohibited to involve in edge collapse. Moreover, after one edge-collapse, a edge-collapse sequence algorithm is investigated, in which vertex sequence is updated by the added new vertex and edge weight defined in Equation (6). This collapse can be cancelled if the validity check is not passed.

The simplification process is terminated under condition of simplification ratio, $R_s = \frac{N - N_m}{N} \times 100\ \%$, where N and N_m are the number of triangle mesh in the original model and after edge collapse, respectively. During simplification procedure, the calculated simplification ratio is compared with a given initial ratio defined by user. If it is smaller than the given ratio, the simplification procedure continuous, or the simplification procedure ends. In some cases, if no candidate edge to be collapsed under the initial simplification control parameter is found, the simplification is terminated, even though the simplification ratio did not reach the initial value.

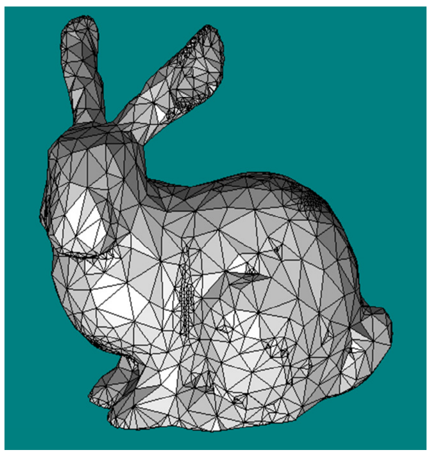
CASE STUDY

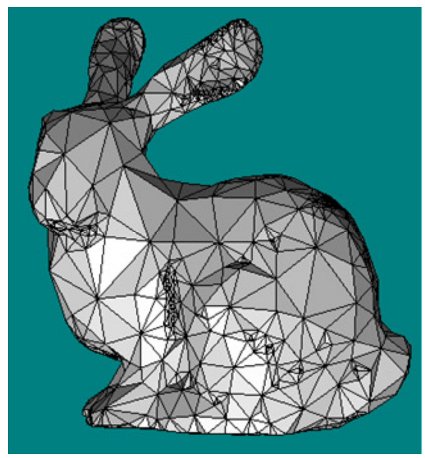
A GUI (graphics user interface) of mesh simplification and refinement has been developed based on C++. In the following example, sliver triangle control parameter of Q is set to 0.5, and the change region of normal direction is within 60°. Figure 5 (b) illustrates the simplified models of Stanford Bunny by using the presented feature sensitivity simplification algorithm. It shows that the mesh on the flat area is largely simplified, such as the area of belly. On the other side, the mesh in the region with a small curvature, such as the areas of ears, is not significantly simplified, and the geometric topology in these areas is preserved. Figure 5 (c) is the simplification results by using the commercial software RAPIDFORM, in which the feature in the area of ears is lost with serious distortion. Comparisons of the simplification results by using the presented method and commercial software RAPIDFORM indicates that the proposed mesh simplified algorithm can keep more



Points: 2904, triangles: 5805, file size: 1204KB

(a) Original model

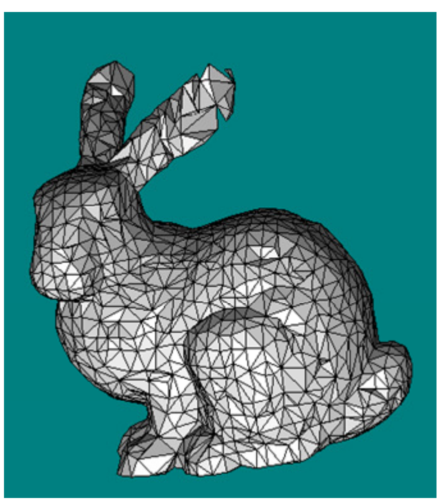


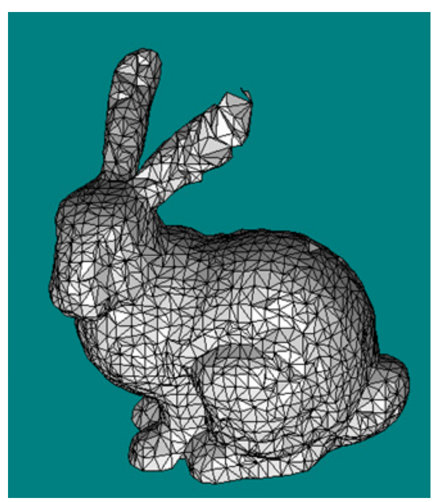


Points: 12033, triangles: 4063, file size: 724KB

Points: 1182, triangles: 2361, file size: 421KB

(b) Simplified mode





Points: 2201, triangles: 4353, file size: 1105KB

Points: 1406, triangles: 2767, file size: 703

(c) Simplified model by commercial software RAPIDFORM

Figure 5. Simplifications of Bunny model.

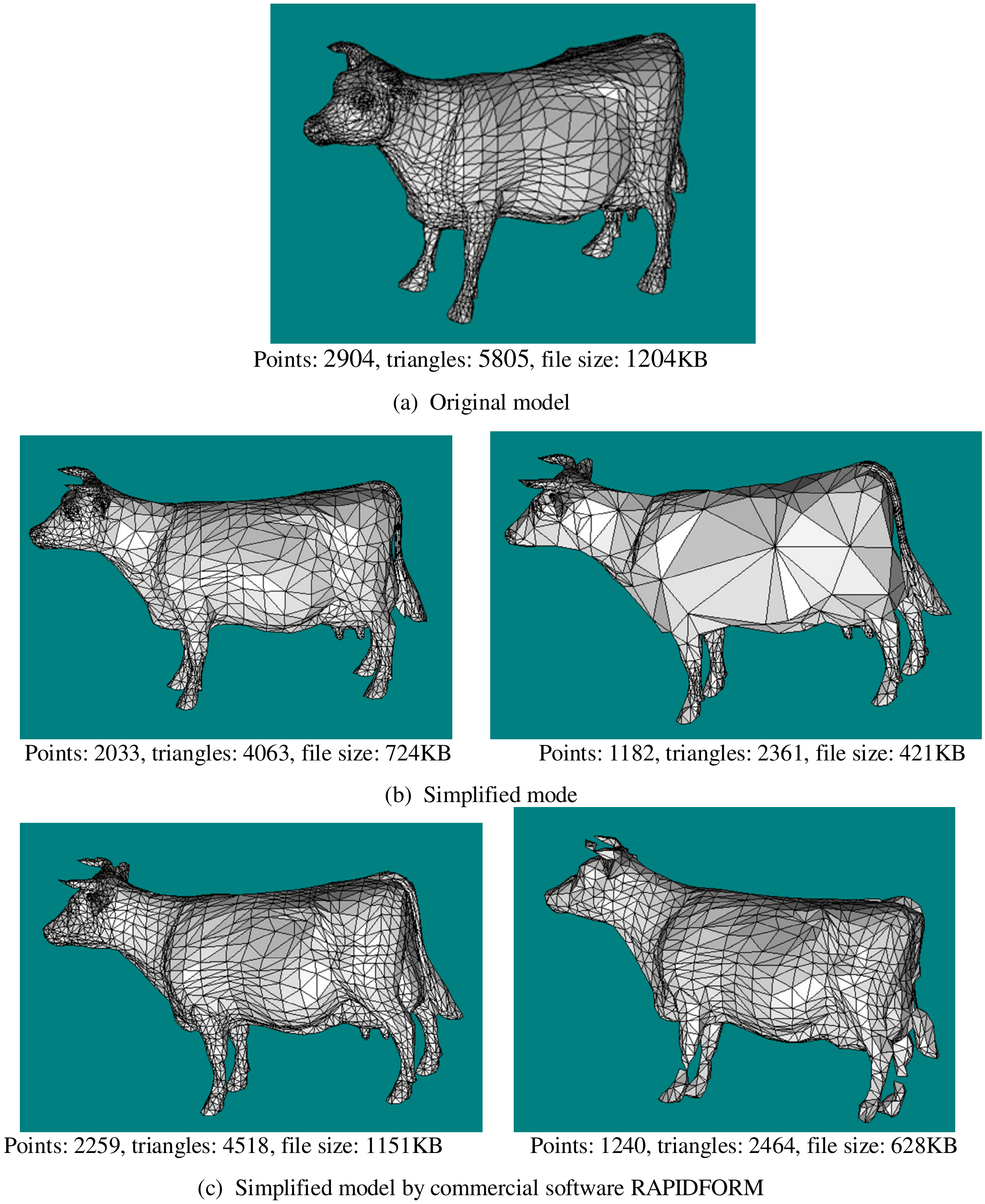


Figure 6. Simplification of a cow model.

accurate topology in the area with small curvature than the uniform mesh algorithm in RAPIDFORM. Figure 6 illustrates the simplified result of another example of a cow model. The areas with distinct feature in horn, eyes, leg and tail demonstrate highly preserved topology than the simplified model by RAPIDFORM. But the body parts are too coarse to reflect the original geometry. More effort will be done to balance the simplifications in the areas with different curvature.

Conclusions

This research investigates a geometric topology preserved mesh simplification algorithm for large-scale triangle mesh model. The presented mesh simplification approach can help to reduce the simplified model size at a large-scale keeping geometrical feature under the edge-weight based simplification sequence control. A coarse mesh model can be obtained by mesh simplification, which can reduce disk and memory requirements and speed up visualization and transmission. Moreover, this research can also be applied into 3D streaming technology for web based collaborative product development, real-time rending, rapid prototyping and more. Further improvement will be carried out to control simplifications around the area near geometrical feature, and to improve the smooth topology between flat and feature areas.

ACKNOWLEDGEMENT

This research is supported by Industrial Technology Research Grant Program of 06A11010a, from New Energy and Industrial Technology Development Organization (NEDO) of Japan.

REFERENCES

Boubekeur T, Alexa M (2009). Mesh Simplification by Stochastic Sampling and Topological Clustering. Comput. Graphics, 33: 241-249.

Garland M, Heckbert P (1997). Surface Simplification Using Quadric Error Metrics. Proceeding of SIGGRAPH’97, pp. 209-216.

González C, Gumbau J, Chover M, Ramos F, Quirós R (2009). User-assisted Simplification Method for Triangle Meshes Preserving Boundaries. Comput.-Aided Des., 41: 1095-1106.

Gumhold S, Wang X, Macleod R (2001). Feature Extration from Point Clouds. Proceeding of the 10 International Meshing Roundtable, Sandia national Laboratories, pp. 293-305.

Guy G, Medioni G (1997). Inference of surfaces, 3d curves and junctions from sparse, noisy 3d data. IEEE Trans. Pattern Anal. Mach. Intell., 19(11): 1265-1277.

Hoppe (1996). Progressive Meshes. Proceedings of SIGGRAPH’96, pp. 99-108.

Kodani T, Manabe T, Taniguchi T (2004). Detection of Feature Lines from Point Clouds and Surface Generation of 3D Domain. Proc. Comput. Eng. Confer., 9: 773-776.

Kodani K, Manabe T, Taniguchi T (2003). Surface Generation from Point Cloud on Surface of 3D Domain. Proc. Comput. Eng. Confer., 8: 837-840.

Li DW, Lu WF, Fuh JYH, Wong YS (2005). Collaborative computer-aided design – research and development status. Comput.-Aided Design, 8(1): 17-30.

Magid E, Soldea O, Rivlin E (2007). A Comparison of Gaussian and Mean Curvature Estimation Methods on Triangular Meshes of Range Image Data. Comput. Vision Image Understanding, 107: 139-159.

Nate H, Jihad ES (2010). Carving for topology simplification of polygonal meshes. Comput.-Aided Des., 42: 67-75.

Noguera JV, Tortosa L, Zamora A (2009). Analysis and Efficiency of the GNG3D Algorithm for Mesh Simplification. Appl. Math. Comput., 197: 29-40.

Razdan A, Bae MS (2005). Curvature Estimation Scheme for Triangle Meshes Using Biquadratic Be´zier Patches. Comput.-Aided Design, 37: 1481-1491.

Tang H, Shu HZ, Dillenseger JL, Bao XD and Luo LM (2007). Moment-based Metrics for Mesh Simplification. Comput. Graphics, 31: 710-718.

Wang LR, Xu F, Wang JC, Hagiwara I (2008). Efficient Iterative Closest Point Algorithm for Measured Surface Registration. The 6th International Conference on Vibration Engineering (ICVE’2008), 4-6 June, Dailian, China.