

Multi-Feature Metric-Guided Mesh Simplification

Hailing Wang, Fu Qiao and Bo Zhou

Abstract This paper proposes a multi-feature metric-guided mesh simplification algorithm, to preserve the crucial visual feature information for simplified model, to solve some mesh simplification algorithms generate poor quality at low levels of detail. Our algorithm is an extended quadric error metrics based on geometric and visual characteristics. The proposed method does not increase the computing time associated with original data and model shape analysis. Moreover, it can preserve the shape features of the models in the processing of simplification. The results of our algorithm have been compared execution time and visual quality with other mesh simplification algorithms, and the results show that our algorithm improved visual shape features for simplified model and got a good quality without additional computing time.

Keywords Mesh simplification · Geometric feature metric · Visual feature metric · Half-edge collapse

1 Introduction

Mesh simplification is a very important research in many domains, such as virtual reality, computer graphics, medical simulation, and scientific visualization. It reduces the number of mesh and allows fast rendering of large-scale models,

H. Wang (✉) · F. Qiao · B. Zhou
College of Computer and Information Engineering, Heilongjiang University
of Science and Technology, Harbin 150027, China
e-mail: whl020519@163.com

F. Qiao
e-mail: 463667708@qq.com

B. Zhou
e-mail: 807571875@163.com

which approximate the original ones, to make the compromise between realism and rendering speed. Mesh simplification is a fundamental method for the large amount of triangle surface models.

Many impressive mesh simplification algorithms have been developed for mesh simplification. All the algorithms can approximately address the different aspects of large amount of triangles and get different objectives based on vertex merging [1], edge collapse [2], and triangle decimation [3], and the key issue of them is simplification metric [4, 5], such as distance metric [1, 6], area metric [7, 8], volume metric [9, 10], normal vector [11], and discrete curvature [12], but these metrics always a single one, not retaining the important feature information after simplification. In recent years, multi-feature metric has received increasing attention as an alternative feature-preserving metric for mesh simplification [13]. To determine the performance of simplification algorithm and the quality of simplification result is following fundamental criteria: visual quality, running time, and memory requirement. Different prior algorithms have different advantage and disadvantage in terms of the three factors. Some algorithms can simply estimate error metric by computing the distance of the vertex to the average plane, or the normal angle metric of vertex, so fast to simplify the original model in running times, but produce poor approximated models [14]; some methods pursue better quality simplification results to improve precise error control and generate good quality approximated models, but perform poorly in execution times [15]; many of the algorithms are fast and generate good quality simplification result, but need larger memory overhead [16]. According to our analysis, most of the above algorithms mainly aim to reduce the quantity of polygons, get low level of detail simplification model while maintaining the shape characteristics of the original model with fast running time.

In this paper, we present a novel approach to preserve shape feature of the model. Our algorithm combines geometric error and visual characteristic based on QSlim, which is driven by half-edge collapse. The error metrics include quadric error metrics, weighted by normal information, and the visual feature of torsion metric, weighted by shape of triangle, to retain the shape and detail characteristics. To verify the conclusion, we make comparison with the several published algorithms on visual image result and error detection; experimental results show our algorithm can effectively get high-quality simplified models and reduce error in the simplification process.

2 Multi-Feature Metric Algorithm

2.1 Geometrical Feature Metric

Quadric error metric [6] associates an error to a vertex according to its distance to a quadratic surface. Quadric error metric is an area-weighted metric, that is, the simplification algorithm based on quadric error metric easily preserves large

triangles and eliminates small triangles. However, the features of the model generally require many small triangles to represent them. Therefore, if one vertex pair with some large triangles is present on a flat surface and another with some small triangles is present on the feature parts of the model, the QSlim is likely to choose the vertex pair on the feature parts to contract and preserve the vertex pair on the flat surface.

Normal field of a surface model plays fundamental role in its appearance, and it has been used for the metric of the geometric distortion in some geometry analysis and simplification processing [17]. Normal field variation truly represents the importance of a vertex. In view of this evidence in support of the strength of normal field, we use the normal field variation across one-ring neighborhood of a vertex to define its importance. So we make use of normal information to describe the features of the mesh surface:

$$C_{\text{norm}}(v) = \frac{\sum_{i=1}^k (1 - \mathbf{n} \cdot \mathbf{n}_i)}{k} \quad (1)$$

where \mathbf{n}_0 is the normal vector of vertex, v , $\{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k\}$ is the set of one-ring neighborhood of vertex v , the included angle between vertex v , and $\{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k\}$ is $\mathbf{n} \cdot \mathbf{n}_1, \mathbf{n} \cdot \mathbf{n}_2, \dots, \mathbf{n} \cdot \mathbf{n}_k$, and they are monotonically decreasing functions for weighting the difference of two normal vectors. $C_{\text{norm}}(v)$ is a higher value in high-curvature areas and small triangles. We use normal information weight quadric error metric for geometrical feature metric as $CA(v) = C_{\text{norm}}(v) \cdot Q(v)$.

2.2 Visual Feature Metric

Visual feature is a very important characteristic for model [18–20]. Although several algorithms, curvature metric, and normal metric can describe the visual feature of the mesh model, the curvature metrics are time-consuming for simplification process, and normal field always does not get good quality approximations because of losing volume. According to the Frenet-Serret formulas, they denote that there is some fixed relationship between curvature and torsion [21]. This illustrates that the torsion is a very important visual feature if with reasonable designing form. To retain the important shape characteristics and not destroy crucial features of original model, the torsion of the model should be considered during the simplification process. Our algorithm reforms the edge contraction based on the foregoing reasons with the torsion error metric; the shape feature cost of edge collapse is defined as follows:

$$T(v) = \frac{\sum_{v_i \in \text{planes}(v)} \|\mathbf{B}_i - \mathbf{B}\|}{\sum_{v_i \in \text{planes}(v)} \|v_i - v\|}, \mathbf{B}_v = v \sum_{v_i \in \text{planes}(v)} h_i \mathbf{n}_i, \tag{2}$$

$$h_i = \begin{cases} (\mathbf{n} \cdot \mathbf{n}_i - \varepsilon)^2 & \text{if } \mathbf{n} \cdot \mathbf{n}_i > \varepsilon \\ 0 & \text{if } \mathbf{n} \cdot \mathbf{n}_i \leq \varepsilon \end{cases}$$

The torsion error metric of vertex, edge, or plane is some approximation to curvatures. In fact, for the same torsion error metric with some big and flat triangles and another with some small and high torsion triangles, the simplification algorithm should likely choose the vertex pair of small and flat triangles to contract and to preserve the vertex pair on the large and high torsion one.

One common measure of surface area is edge length, for example, the area of one triangle is $a = \sqrt{s(s - l_1)(s - l_2)(s - l_3)}$, and $s = 1/2(l_1 + l_2 + l_3)$. However, this measure is very complex and takes more computing time. For geometry theory, given a fixed-length edge e , the larger the area of the triangle e , the better the quality is, for example, equiangular triangle is best; the obtuse triangle is poor. This method is same as with angle, e.g., equiangular triangle is best, obtuse triangle, the opposite angle of e is smaller, it is worse. To improve computing efficiency, one such approximation can be defined as length ratio in the triangles of vertex pair as follows:

$$\varsigma_{\text{length}} = \begin{cases} \frac{l_i}{l_{i+1} + l_{i+2} + l_{i+3} + l_{i+4}}, & l_i \text{ is inner edge} \\ \frac{l_i}{l_{i+1} + l_{i+2}}, & l_i \text{ is boundary edge} \end{cases} \tag{3}$$

l_i is one edge of triangles; $\{l_{i+1}, l_{i+2}, l_{i+3}, l_{i+4}\}$ is the other edges in the triangles, which include l_i . So, the visual feature can be obtained as $CB(v) = \varsigma_{\text{length}} \cdot T(v)$

3 Simplification Algorithm Summary

The paper develops an improved method for the surface simplification, combining two parts; one is geometrical feature metric that derives from quadric error metric, weighted by vertex normal information, and the other is visual feature metric that is torsion metric, weighted by edge length ratio of triangles. According to the foregoing analysis, the total error metric function is defined as $Cost_{\text{total}} = CA + CB$.

The proposed algorithm refers to and amends half-edge contraction methods

and QSlim algorithm. Our algorithm is summarized in the five major steps described as follows:

- Step 1: Compute the quadric error metric $Q(v)$, normal information $C_{\text{norm}}(v)$, torsion metric $T(v)$, and edge length ratio ζ_{length} for all the initial vertices in section two.
- Step 2: Select the valid vertex-pair (v_i, v_j) as QSlim.
- Step 3: Calculate the total cost of the valid vertex-pair from $\text{Cost}_{\text{total}}$; construct the priority queue in increasing sequences.
- Step 4: Iteratively remove the lowest-cost edge from the data sequence based on half-edge contraction, and then update all occurrences of related edges and triangles for the simplification one.
- Step 5: Repeat the previous steps until there are no candidate or the error is up to threshold.

4 Experimental Results

The experimental hardware environment is Windows XP with Core E6400, 4G memory, and NVIDIA 9600GT, and the software programming development tool is Visual C++ with DirectX. The experimental models in this work are dog, ox, hand, and raptor. Their detailed information is presented in Table 1.

In the experiments, we also compared our results with those methods by Discrete Curvature Norm (DCN) [12], Visual Feature-Preserving (VFP) [7], QSlim [6], and our algorithm.

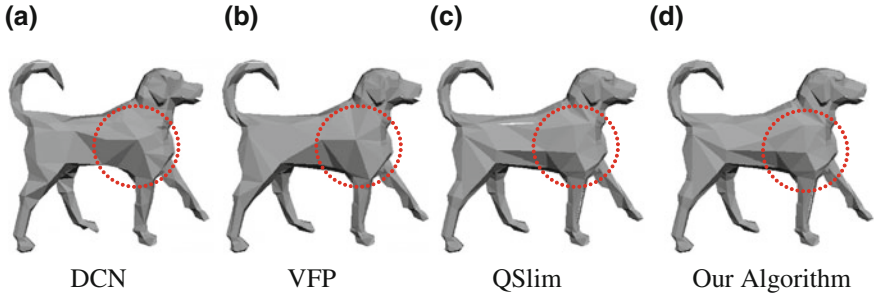
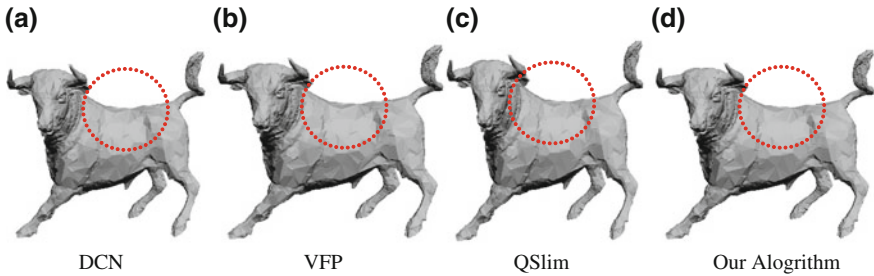
Figure 1 shows the simplification results for the dog model using DCN, VFP, QSlim, and our algorithm. The number of triangles of the simplified model is 913 (30 % triangle optimization). The results show that the DCN degenerates the dog model because its curvature characteristics is weaker to lose the details and shape features, while the VFP, QSlim, and our algorithm get a good quality simplified result.

Figure 2 shows the simplification results for the OX model using DCN, VFP, QSlim, and our algorithm. The number of triangles of the simplified model is 11,409(8 % triangle optimization). The results show that the DCM, QSlim, VFP, and our algorithm all can get a good simplification with some shape feature information because simplification ratio is small; however, QSlim and our algorithm are better than the other two methods.

Besides comparison the visual simplification results, the error assessment between the original model and the simplified one, with a percentage of the number of faces in the approximation, is calculated by maximum and mean geometrical errors of Metro tools under Metro detection. In the experiments, we compare the results with those methods by VFP, QSlim, and our algorithm, in which the running time is close. These data show that there is no significant

Table 1 Information of the original models

Model	Vertices	Triangles
Dog	1,263	1,331
Ox	6,211	12,397

**Fig. 1** Simplified model of dog model with different algorithms **a** DCN, **b** VFP, **c** QSLim, and **d** our algorithm**Fig. 2** Simplified model of OX model with different algorithms **a** DCN, **b** VFP, **c** QSLim, and **d** our algorithm

difference between our algorithm and the other algorithms in terms of the high-resolution models, with a small number of triangles simplified; however, these methods differ significantly when a large number of triangles are simplified. Figure 3 shows the experimental results. For the dog model, the proposed algorithm gets respectively, less 6–28 %, 6–36 % than QSLim, VFP in mean geometric error, and less 2–11 % than QSLim in maximum geometric error. For the OX model, the proposed algorithm gets respectively, less 19–25 %, 16–37 % than QSLim, VFP in mean geometric error, and less 6–20 % than QSLim in maximum geometric error.

We can obtain from the experiments that the results in terms of maximum geometrical error are not as clear as the mean geometrical error; the development

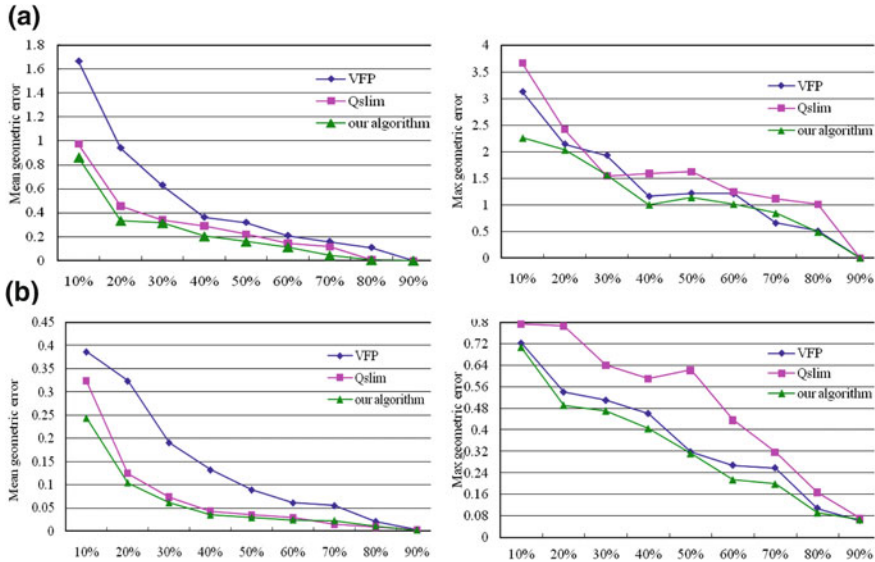


Fig. 3 Error comparison between the simplified model and the original one. **a** Dog model. **b** OX model

tendency and conclusions can be get from these data; our algorithm gives a better simplification quality than Qslim and VFP. The data show the proposed method can achieve a 2–72 % improvement in quality, or error reduction.

5 Conclusions

This paper proposed a novel mesh simplification algorithm with improvements to surface simplification for complex models. The algorithm combines geometrical error metric with quadric-error-metric-weighted normal information and visual feature metric with torsion-metric-weighted edge length ratio. With these metrics, our algorithm preserves the shape feature of its input mesh model in the simplification process. The experiment demonstrated that our algorithm can get efficient effects with the more error parameter for the mesh simplification.

References

1. González, C., Gumbau, J., Chover, M., et al.: User-assisted simplification method for triangle meshes preserving boundaries. *Comput. Aided Des.* **41**(12), 1095–1106 (2009)
2. Van, J., Shi, P., Zhang, D.: Mesh simplification with hierarchical shape analysis and iterative edge contraction. *IEEE Trans. Visual Comput. Graphics* **10**(2), 142–151 (2004)

3. Wang, X., Wu, J., Wu, E.: A novel technique based on triangle decimation for tetrahedral simplification. *Acta Electronica Sinica* **35**(12), 2343–2346 (2007)
4. Qu, L., Meyer, G.W.: Perceptually guided polygon reduction. *IEEE Trans. Visual Comput. Graphics* **14**(5), 1015–1029 (2008)
5. Menzel, N., Guthe, M.: Towards perceptual simplification of models with arbitrary materials. *Comput. Graphics Forum*. **29**(7), 261–270 (2010)
6. Garland, M., Heckbert, P.S.: Simplification using quadric error metrics, Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques 1997. pp. 209–216. ACM Press, New York (1997)
7. Hussain, M., Okada, Y., Nijima, K.: Efficient and feature-preserving triangular mesh decimation. *J. WSCG* **12**(1–3), 167–174 (2004)
8. Park, I., Shirani, S., Capson, D.W.: Mesh simplification using an area-based distortion measure. *J. Math. Model. Algorithms* **5**, 309–329 (2006)
9. Tang, H., Shu, H.Z., Dillenseger, J.L., et al.: Moment-based metrics for mesh simplification. *Comput. Graphics* **31**(5), 710–718 (2007)
10. Yuan-Feng, Z., Cai-Ming, Z., Ping, H.: Feature preserving mesh simplification algorithm based on square volume measure. *Chin. J. Comput.* **32**(2), 203–212 (2009)
11. Schroder, F., Rossbach, P.: Managing the complexity of digital terrain models. *Comput. Graphics* **18**(6), 775–783 (1994)
12. Kim, S.J., Jeong, W.K., and Kim, C.H. LOD generation with discrete curvature error metric. *J. Kiss: Comput. Syst. Theory*, 245–254 (2000)
13. Hoppe, H. New quadric metric for simplifying meshes with appearance attributes. Proceedings of the IEEE Visualization'99, Piscataway, pp. 59–66. IEEE Press, New York (1999)
14. Bao-cai, D.X.H.Y., De-hui, K.O.N.G.: Edge collapse simplification based on weighted quadric error metrics. *J. Beijing Univ. Technol.* **33**(7), 731–736 (2007)
15. Li, L., He, M., Wang, P. Mesh simplification algorithm based on absolute curvature-weighted quadric error metrics. Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications, pp. 399–403. TaiChung (2010)
16. Yang, Jun, Zhu, Shijiao: An image retrieval method by integrating semantics and visual features. *J. Comput. Inf. Syst.* **7**(9), 3219–3225 (2011)
17. Wei, J., Lou, Y.: Feature preserving mesh simplification using feature sensitive metric. *J. Comput. Sci. Technol.* **25**(3), 595–605 (2010)
18. Shi, Z., Luo, H., Niu, X.: Saliency-based structural degradation evaluation of 3D mesh simplification. *IEICE Electron. Express* **8**(3), 161–167 (2011)
19. Thomas, D.M., Yalavarthy, P.K., Karkala, D., et al.: Mesh simplification based on edge collapsing could improve computational efficiency in near infrared optical tomographic imaging. *IEEE J. Sel. Top. Quantum Electron.* **18**(4), 1493–1501 (2012)
20. Lavoue, G., Corsini, M.: A comparison of perceptually-based metrics for objective evaluation of geometry processing. *IEEE Trans. Multimedia* **12**(7), 636–649 (2010)
21. Bin-Shyan, J., Juin-Ling, T., Wen-Hao, Y.: An efficient and low-error mesh simplification method based on torsion detection. *Visual Comput.* **22**, 56–67 (2006)