

CURVATURE-WEIGHTED SURFACE SIMPLIFICATION ALGORITHM USING VERTEX-BASED GEOMETRIC FEATURES

HAN-SOO CHOI¹, DALHYEON GWON², HEEJAE HAN², AND MYUNGJOO KANG^{2†}

¹DEPARTMENT OF COMPUTATIONAL SCIENCE AND TECHNOLOGY, SEOUL NATIONAL UNIVERSITY, SEOUL, REPUBLIC OF KOREA

Email address: hschoi86@snu.ac.kr

²DEPARTMENT OF MATHEMATICAL SCIENCES, SEOUL NATIONAL UNIVERSITY, SEOUL, REPUBLIC OF KOREA

Email address: pygmali00n85@snu.ac.kr, hhj5575@snu.ac.kr, mkang@snu.ac.kr

ABSTRACT. The quadratic error metric (QEM) algorithm has been frequently used for simplification of triangular surface models that utilize the vertex-pair algorithm. Simplified models obtained using such algorithms present the advantage of smaller storage capacity requirement compared to the original models. However, a number of cases exist where significant features are lost geometrically, and these features can generally be preserved by utilizing the advantages of the curvature-weighted algorithm. Based on the vertex-based geometric features, a method capable of preserving the geometric features better than the previous algorithms is proposed in this work. To validate the effectiveness of the proposed method, a simplification experiment is conducted using several models. The results of the experiment indicate that the geometrically important features are preserved well when a local feature is present and that the error is similar to those of the previous algorithms when no local features are present.

1. INTRODUCTION

In recent years, computer graphics technology based on tridimensional surfaces using triangular meshes have been increasingly employed in a number of fields [1]. In particular, with the advancements in modern data collection methods, generation of high resolution surfaces is possible [2], and the surfaces can be utilized in day-to-day activities by 3D printing them [3]. However, using a tridimensional surface requires large computer storage capacity. Consistently using these data in simulation programs utilizing tridimensional surfaces is time-consuming. Thus, the necessity of simplification has been an important study topic [4, 5, 6, 7, 8, 9]. Based

Received by the editors March 4 2020; Accepted March 10 2020; Published online March 25 2020.

2000 *Mathematics Subject Classification.* 93B05.

Key words and phrases. Surface Simplification, Quadratic Error Metrics, Curvature-Weighted, Vertex-based Geometric Features.

[†] Corresponding author.

Myungjoo Kang was supported by the National Research Foundation of Korea (2015R1A5A1009350, 2017R1A2A1A17069644).

on the results of simplification, a method of calculating the degree of error from using the simplified surfaces compared with the original surfaces is presented in [10].

Examples of some commonly used simplification algorithms are edge contraction, triangle contraction, vertex decimation, and vertex-pair contraction [4, 11, 12, 13]. All of these algorithms aim to reduce the number of vertices and faces comprising the surface. Furthermore, these algorithms also aim to preserving the important features on a surface.

In 1997, Garland *et al* proposed a simplification algorithm using the vertex-pair contraction algorithm [4]. Adequate simplification was achieved, but the important features of the surface were found to have disappeared. To address this problem, in 2006, B.-S. Jong *et al* proposed a method to preserve the local features using the torsion detection algorithm [5]. In 2002, S.-J. Kim *et al* [6] proposed a simplification algorithm using the discrete curvature norm, and in 2010, L. Li *et al* applied the curvature norm in quadratic error metrics (QEM) using the curvature-weighted method [7]. Further, in 2015, L. Yao *et al* proposed a discrete curvature weighted QEM method [9]. It is important that, when surface simplification is conducted, the local features are preserved well. However, the curvature-weighted algorithm of L. Li *et al* quickly contracts the vertex-pair, which has a smaller absolute sum of two principal curvatures regardless of the geometric features in a vertex. Thus, this algorithm has still the disadvantage of easily losing the local features on a surface.

Therefore, this study suggests conducting the simplification after checking the geometric features on the corresponding vertex. This study categorizes the geometric features of a given vertex on a surface into three: corner, ridge, and smooth [14]. In addition, simplification is conducted in the smooth area rapidly by varying the cost pair according to the category. On the contrary, the simplifications of the ridge and corner areas were carried out more slowly. By using this method, more specific features were preserved well while simplification was being conducted.

This paper is organized as follows: Section 2 briefly explains the existing QEM algorithm and curvature-weighted method using the vertex-pair contraction. Section 3 proposes the simplification method using the vertex-based geometric feature. Section 4 evaluates the suggested method through an experiment and confirms the error with an original surface using the evaluation proposed by Cignoni *et al* [10]. Finally, Section 5 draws the conclusion.

2. CURVATURE-WEIGHTED SURFACE SIMPLIFICATION

Quadratic Error Metrics: The algorithm was suggested by Garland *et al* in 1997 [4]. As shown in Fig. 1, on a given surface, this algorithm uses the vertex-pair contraction. A simplification of the algorithm is as follows:

Let us assume that the planes of the adjacent triangles from the vertex v are $planes(v)$. In the one plane's equation of triangular mesh $ax + by + cz + d = 0$ within $planes(v)$, the corresponding vertex v is defined as $v = [x, y, z, 1]^T$ and $p = [a, b, c, d]^T$. Here, $a^2 + b^2 + c^2 = 1$. Using the coefficients of the plane equation, the following matrix K_p is defined:

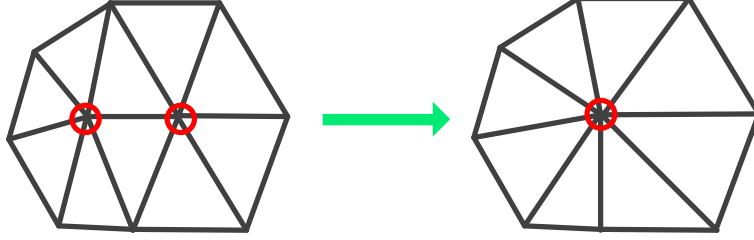


FIGURE 1. Vertex-pair contraction

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

Based on the defined K_p , $Q(v)$ is defined as $Q(v) = \sum_{p \in \text{planes}(v)} K_p$; that is, by adding K_p of all the triangles adjacent to the vertex v , the quadratic error metric of the vertex v is calculated.

In order to conduct vertex-pair contraction, adequate \bar{v} which is newly contracted from two vertices should be calculated. \bar{v} is set as follows, from the sum of Q_1, Q_2 that are extracted from v_1, v_2 . If the matrix for the calculation of \bar{v} does not have an inverse, $\bar{v} = (v_1 + v_2)/2$ can be used.

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix} \bar{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, [q_{i,j}] = Q_1 + Q_2$$

Finally, after calculating $\Delta(\bar{v}) = \bar{v}^T(Q_1 + Q_2)\bar{v}$, $\Delta(\bar{v})$ can be chosen and contracted, which has the smallest value out of the entire vertex-pairs. This process can be repeated for any required number of times.

A brief summary of the QEM algorithm has been given below.

- (1) Calculate Q matrix at all vertices on the given surface.
- (2) Identify the valid pair (v_1, v_2) from the respective vertex; that is, consider the cases when (v_1, v_2) is edge or $\|v_1 - v_2\| < t$. t is the threshold parameter.
- (3) Calculate \bar{v} from the respective vertex-pair (v_1, v_2) and find cost $\Delta(\bar{v})$.
- (4) Conduct contraction by selecting the pair $\Delta(\bar{v})$ which has the smallest value at the corresponding heap of cost $\Delta(\bar{v})$.
- (5) Repeat steps 2, 3, and 4 for the required number of iterations.

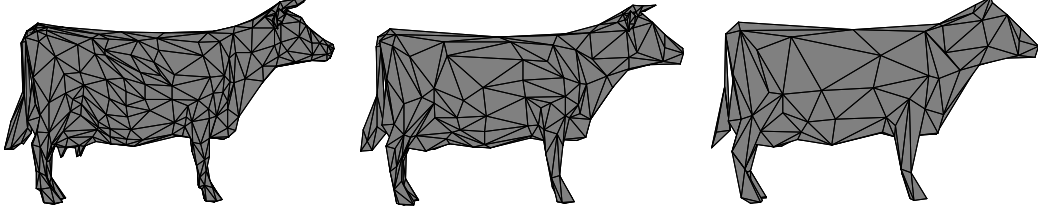


FIGURE 2. Model - cow : Local features disappear so easily. The number of face is 5804, 500, 300 respectively.

The results indicated that the simplification of the QEM algorithm, in general, was achieved adequately. However, when this algorithm was used, a number of local features were found to have disappeared because of the repetitive contraction. From Fig. 2, it can be seen that the region of horn of the surface of the model has disappeared.

In order to address this issue, S.-J. Kim *et al* proposed a simplification algorithm using the discrete curvature norm (2002)[6]; Long Li *et al* applied this method to the QEM algorithm (2010)[7].

Triangular meshes are composed of vertices, edges, and faces. First, the vertices are connected by edges, and the faces are organized as $F = \{f_k = (v_{k_1}, v_{k_2}, v_{k_3})\}_k$. As shown in Fig. 3, let us assume that the adjacent vertices of any vertex v are v_1, v_2, \dots, v_n . An edge is defined as $e_i = v_i - v$, and the angle of arc formed by the edge is defined as $\phi_i = \angle(e_i, e_{i+1})$. Further, the three vertices are composed of two edges e_i and e_{i+1} , which are set as the surface $f_i = (v, v_i, v_{i+1})$. Thus, the normal vector of the corresponding face is $n_i = (e_i \times e_{i+1}) / \|e_i \times e_{i+1}\|$. The dihedral angle formed by the normal vector of two adjacent triangles is defined as $\beta_i = \angle(n_{i-1}, n_i)$.

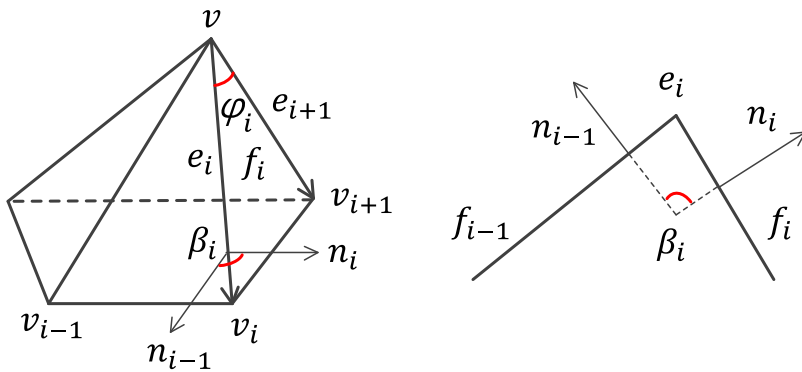


FIGURE 3. Local configuration related to vertex v

Under such conditions, the discrete Gaussian curvature K at v is defined as follows [6]:

$$K = \left(2\pi - \sum_{i=1}^n \phi_i \right) / \left(\frac{A}{3} \right)$$

A is the sum of the area of the adjacent triangles with vertex v . The discrete mean curvature H at v is defined as follows:

$$H = \left(\frac{1}{4} \sum_{i=1}^n \|e_i\| \beta_i \right) / \left(\frac{A}{3} \right)$$

Therefore, the principal curvatures on a corresponding vertex v are obtained as follows: If $H^2 - K < 0$, $H^2 - K = 0$.

$$\kappa_1 = H + \sqrt{H^2 - K}, \kappa_2 = H - \sqrt{H^2 - K}$$

The method in [7] preserved the edge having the large sum of absolute curvature by multiplying the value resulting from the calculation of the cost pair based on the following definition of the absolute curvature of the corresponding vertex v :

$$\kappa_{abs} = |\kappa_1| + |\kappa_2|$$

Figure 4 shows the results based on the QEM and curvature-weighted methods. It can be seen from Fig. 4 that the region of horn of the cow rapidly disappears when the QEM [4] algorithm is used, whereas this region is preserved well when the curvature-weighted [7] algorithm is used.

Therefore, it is clear that the curvature-weighted algorithm preserves the local features better than the QEM algorithm. However, when this algorithm is used, the geometric features comprising a small number of triangle meshes disappear easily. The following part of this section explains the simplification method which utilizes the vertex-based geometric features.

3. VERTEX-BASED GEOMETRIC FEATURES

In the previous method, when the sum of absolute curvature in the crease area was smaller than the sum of the flat area on a curved surface, it contracted rapidly. In other words, despite being a geometrically significant feature, the corresponding feature disappears easily when the sum of absolute curvature is small. In Fig. 5, the two vertices marked by red circles on the left surface are smooth areas and the two vertices on the right surface are crease areas. κ_{abs} values of the two vertices are larger than 0 on the vertices on the left. On the contrary, κ_{abs} value of one vertex in the smooth area is almost 0 and between κ_1 and κ_2 of the other vertex is larger than 0, in the case of the vertices on the right. Thus, a case where the sum of the absolute curvature on the vertices on the right is smaller than that on the left occurs. In this

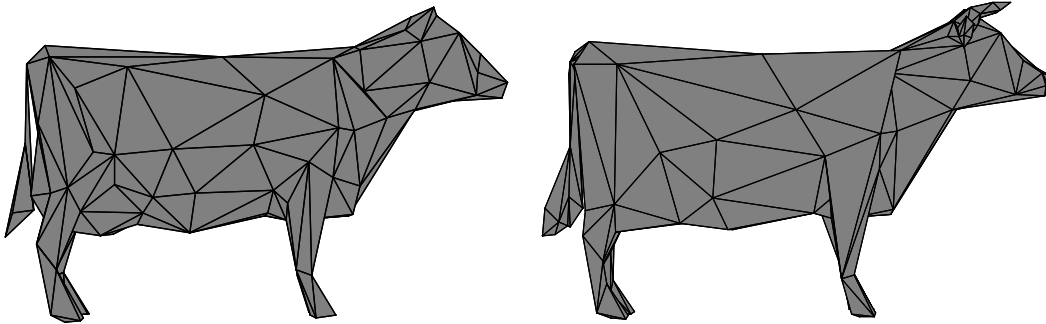


FIGURE 4. Comparison between QEM and curvature-weighted method respectively. The number of faces is 300 equally.

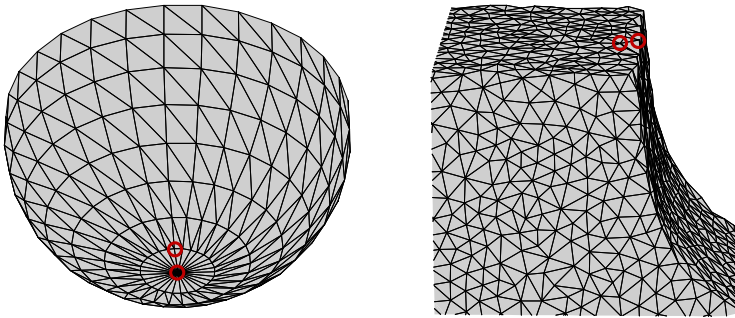


FIGURE 5. Example - Crease areas can be contracted faster than smooth area

case, although the features of the vertices on the right are more important than those of the left, it contracts faster. This is a disadvantage of the existing simplification algorithm [7].

Thus, it is difficult to determine the local geometric structure from the corresponding vertex with the little information available on Gaussian curvature, mean curvature, and principal curvature on the surface comprising the triangle mesh. It is necessary to determine a certain structure near a vertex using the information on the adjacent triangle.

In Fig. 6, the vertex is marked in blue to determine the geometric structure. Let us assume that the angles of respective triangles that form an angle with the vertex are θ_i . Further, because the corresponding triangle can be an obtuse triangle, the weight of angle is set as $\omega_i = \min(\theta_i, \pi - \theta_i)$.

In Fig. 6, the blue vertex is p and the equation of the plane over this vertex which contains the i -th triangle is γ_i . If the offset of this plane's equation is δ , $\delta = -p^T n$. Now, n is the unit normal vector of the plane equation. The signed distance from the plane equation γ to optional

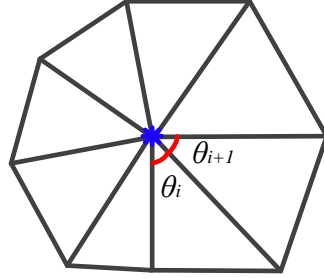


FIGURE 6. Demonstration of i -th vertex and adjacent triangle's angle information

$x \in R^3$ is defined as follows:

$$d(x, \gamma) = (x - p^T)n = x^T n + \delta$$

The calculation of the weighted sum of the squared distance from x with regard to the γ_i set of the plane equations adjacent to the vertex p is as follows:

$$Q(x) = \sum_i \omega_i d^2(x, \gamma_i) = x^T A x + 2b^T x + c$$

Here, $A = \sum_i \omega_i n_i n_i^T$, $b = \sum_i \omega_i \delta_i n_i$, $c = \sum_i \omega_i \delta_i^2$. Thus, because A is a symmetric positive definite matrix, it has three eigenvalues based on the spectral theorem. Let us assume that the eigenvalue of A is λ_i ($\lambda_1 \geq \lambda_2 \geq \lambda_3$). If e_i is the corresponding eigenvector of λ_i , the expression of spectrum theory is as follows:

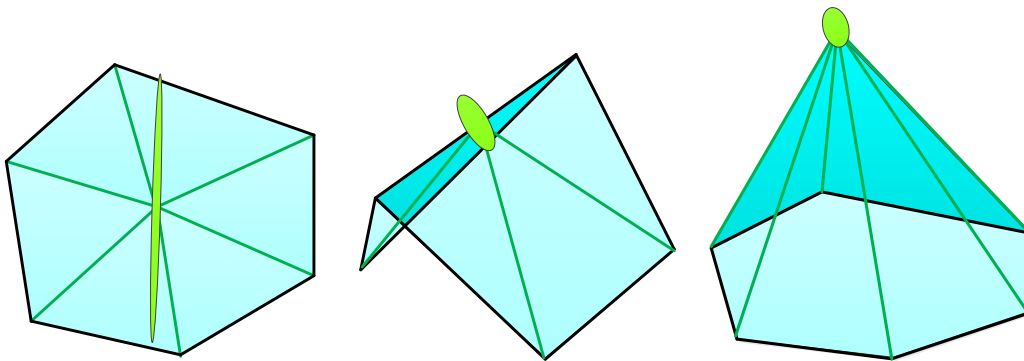


FIGURE 7. Geometrical structure of smooth, ridge and corner, respectively

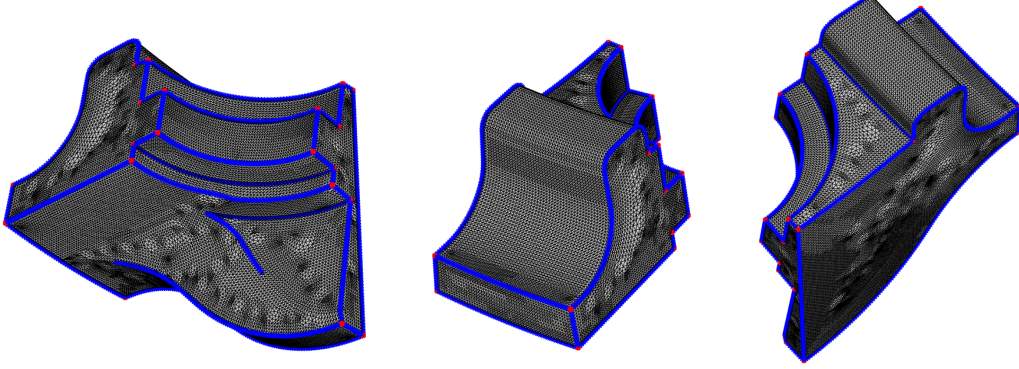


FIGURE 8. Result of geometric feature(Fandisk) - corner(red), ridge(blue) and smooth(none)

$$A = \sum_{i=1}^3 \lambda_i e_i e_i^T = (\lambda_1 - \lambda_2) E_1 + (\lambda_2 - \lambda_3) E_e + \lambda_3 E_1$$

where $E_d = \sum_{i=1}^3 \hat{e}_i \hat{e}_i^T$. Additional necessary variables – α, β, g_i – are defined as $\alpha = \tan^2(\psi/2)$, $\beta = \cot\psi$, and $g_i = |b^T \hat{e}_i| / \min(\epsilon \lambda_1, \lambda_i)$. The parameter ψ is set as $\psi = 20^\circ$ and $\epsilon(10^{-7})$ avoids potential division by zeros. As shown in Fig. 7, [14] identified the geometric structures – corner, ridge, and smooth – of the corresponding vertex by applying the standards mentioned below, using the above-mentioned eigenvalue and values of α and β .

1. If $\operatorname{argmax}_i \{g_i\} = 3$ or $\lambda_3 \geq \beta \max\{\lambda_1 - \lambda_2, \lambda_2 - \lambda_3\}$, the corresponding vertex v is corner.
2. If $\operatorname{argmax}_i \{g_i\} = 2$ or $\lambda_2 \geq \alpha \lambda_1$, the corresponding vertex v is ridge.
3. If the corresponding vertex v does not fall under any of the two previous standards, the corresponding vertex is smooth.

As shown in Fig. 8, the aforementioned theory is validated using the Fandisk model. The Fandisk model is used because it clearly demarcates the corner, ridge, and smooth. The red circles denote corner, the blue stars denote ridge, and the unmarked regions denote smooth. As shown in Fig. 8, adequate results were obtained along with the determination of corner, ridge, and smooth.

The method of varying the weights of cost pair calculation according to the location of corner, ridge, and smooth, suggested by this study, is as follows:

Defined as $\kappa_{abs} = |\kappa_1| + |\kappa_2|$, $\kappa_{abs-max} = \max(|\kappa_1|, |\kappa_2|)$, and $\kappa_{abs-min} = \min(|\kappa_1|, |\kappa_2|)$

1. If the corresponding vertex is a corner,

$$\kappa_{weight-ith} = \kappa_{abs} \times \kappa_{abs-max}^2$$

2. If the corresponding vertex is a ridge,

$$\kappa_{weight-ith} = \kappa_{abs} \times \kappa_{abs-max}$$

3. If the corresponding vertex is smooth,

$$\kappa_{weight-ith} = \kappa_{abs} \times \kappa_{abs-min}$$

When the corresponding vertex is smooth, multiply the smaller value between $|\kappa_1|, |\kappa_2|$ with κ_{abs} . If the corresponding vertex is ridge, the bigger value between $|\kappa_1|, |\kappa_2|$ should be multiplied with the existing κ_{abs} to set a small weight to this feature. When the corresponding vertex is corner, multiply the bigger value's square between $|\kappa_1|, |\kappa_2|$ with the existing κ_{abs} in order to set a big weight. When simplification is to be conducted after setting $\kappa_{weight-ith}$, multiply this weight value when calculating the cost pair. Using this process, the ridge and corner areas are preserved, and the smooth area disappears rapidly.

4. EXPERIMENTAL RESULTS

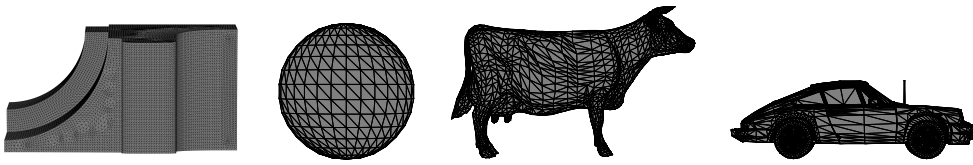


FIGURE 9. Original models for the experiment. Fandisk, sphere, cow, big-porsche respectively

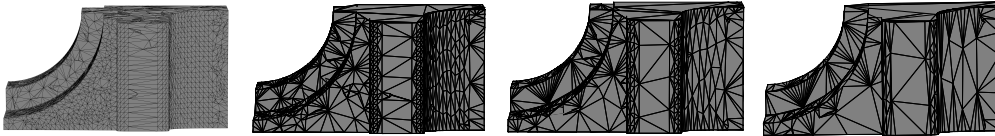


FIGURE 10. Simplification result from proposed method - Fandisk. The number of faces is 10000, 3000, 2000, 1000 respectively

In this section, the validity of the proposed architecture is verified through experiments. The performance of the PC for experiment is conducted on Intel Core i5-6500 @ 3.20GHz, 16.0GB RAM. The corresponding experiment is conducted using Matlab [15]. Fandisk, sphere, cow, and big-porsche are used as experimental models. Figure 9 shows the original data of the models: Fandisk - number of faces is 51,784 and number of vertices is 25,894; sphere - number of faces is 840 and number of vertices is 422; cow - number of faces is 5,804 and number of vertices is 2,903; big porsche - number of faces is 10,474 and number of vertices is 5,247.

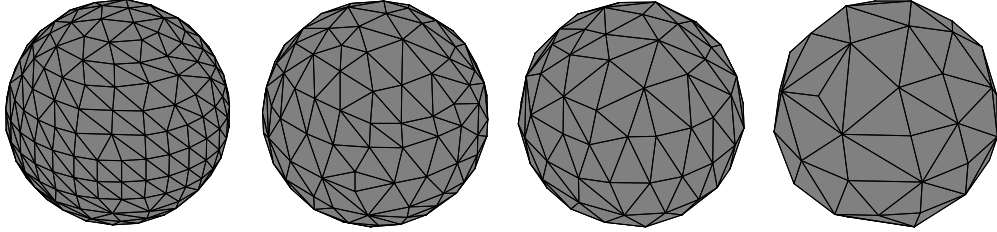


FIGURE 11. Simplification result from proposed method - sphere.
The number of faces is 500, 300, 200, 100 respectively

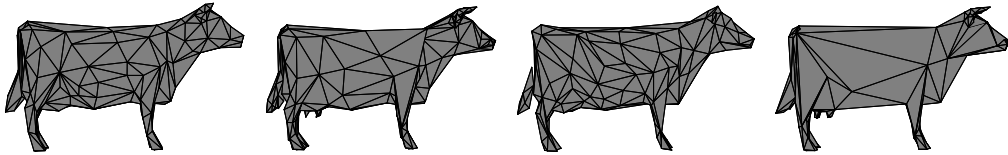


FIGURE 12. Simplification result - cow, The number of faces : 400
QEM, curvature-weighted, discrete curvature, proposed method respectively

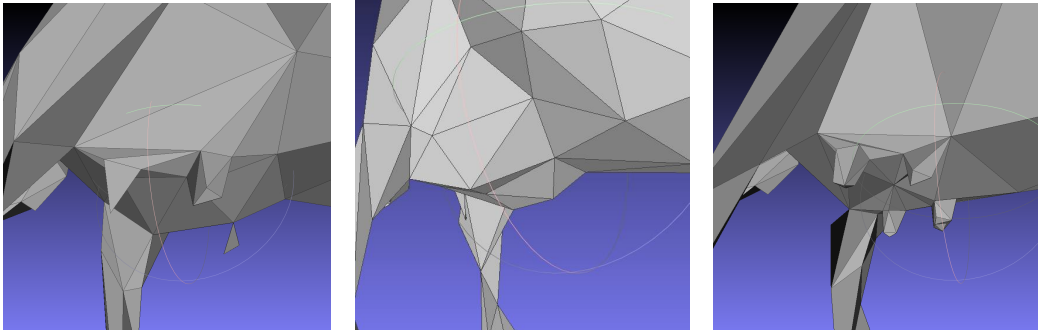


FIGURE 13. Simplification result of the breast area. curvature-weighted, discrete curvature, proposed method respectively

Figure 10 shows the experimental results of Fandisk, and Fig. 11 shows the experimental results of sphere. The results of these models indicate that the simplification was adequate for the model with no local feature.

Figure 12 shows the experimental results of the model cow. When the number of faces comprising the surfaces is 400, the result is similar to that of the curvature-weighted method and

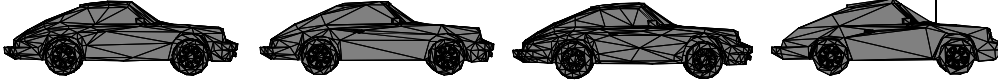


FIGURE 14. Simplification result - big-porsche , The number of faces : 2000
QEM, curvature-weighted, discrete curvature, proposed method respectively

better than the QEM and discrete curvature methods, as seen from Fig. 12. However, as seen from Fig. 13, the method proposed in this paper preserved the local features such as the breast of cow better than the curvature-weighted and discrete curvature methods well, especially in terms of details.

Finally, Fig. 14 shows the experimental results acquired using the model big-porsche. When the QEM, curvature-weighted, and discrete curvature methods were used, the antenna part of the model disappeared when the number of faces was 2,000. However, when the method proposed in this study was used, the antenna part remained.

Next, the simplification method's validation suggested by this study compares the simplified gap with the error of the original surface through numerical calculation. This method was suggested by Cignoni *et al* in 1998 [10]. The calculation method defined in this study uses max error (max distance) and mean error (mean distance). First, the distance $e(p, S)$ is defined as follows, in the case of given vertex p and curved surface S .

$$e(p, S) = \min_{p' \in S} d(p, p')$$

Here, $d(,)$ is the Euclidean distance between two vertices at R^3 . Based on this definition, max error E and mean error E_m are defined as follows:

$$E(S_1, S_2) = \max_{p \in S_1} e(p, S_2)$$

$$E_m(S_1, S_2) = \frac{1}{|S_1|} \int_{S_1} e(p, S_2) ds$$

Tables 1 to 4 list the results obtained from this calculation. Because the tables contain large amounts of data, they have been placed in the appendix. Here, we attached a graph of Table 1 to 4 for understanding as Fig. 15. Red curve means QEM algorithm, green curve means curvature weighted algorithm, blue curve means discrete curvature algorithm and black curve means proposed method.

The corresponding significant figures were approximated to six places of decimals, similar to the existing studies [5, 7]. The Fandisk and sphere models mostly provided smaller errors in the case of QEM, curvature-weighted, and discrete curvature methods because they have no significant local features. In the case of the mean error of the Fandisk model, when the number of faces is 10,000, the mean error of the proposed method is even smaller than that of the QEM

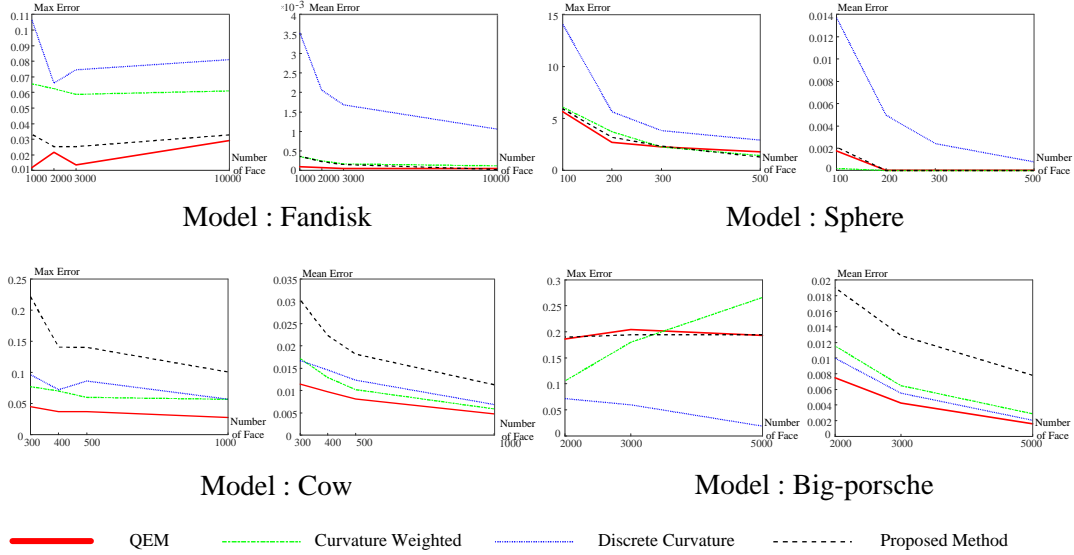


FIGURE 15. Comparison of max error and mean error by each model from experimental results

method. Moreover, in the case of the sphere, when the number of faces is 500, the max error of the proposed method is smaller than that of the QEM method. In the case of mean error, because the number of faces of up to 200 is too small, all the values were measured close to 0 by all the methods, except the discrete curvature method.

However, in the case of cow and big-porsche models, the error of the proposed method is larger than that of the other methods. Because the smooth area of the existing algorithm is preserved well, they have small errors with the surface model but lose local features easily. On the other hand, when the proposed method is used, results with larger errors than the other methods in the smooth area and the results which preserved the local features well are obtained, which have been provided in the tables in the appendix. Therefore, the proposed method presents a slightly larger error than the existing methods.

5. CONCLUSION

This study proposed a vertex-based geometric feature using triangle mesh and surface simplification using the absolute curvature of the mesh vertex. The algorithm was validated by experiments based on various models, and the difference between the simplified surface and the original surface was identified through the mean and max errors proposed by max error E and mean error E_m . The following results can be drawn from the simplification algorithm

proposed in this paper:

1. The corresponding model has similar mean and max distances with the original surface as other existing algorithms, if it does not have local features.
2. The corresponding model preserves the local features well compared with the other existing algorithms even if contraction is conducted many times, if it has local features.

We believe the simplified models can be used in mobile devices because they occupy only a small storage. Future studies will be focused on an algorithm that can enhance the mesh quality such that the shape of the used meshes is similar to that of an equilateral triangle, thereby enabling to conduct simplification.

6. APPENDIX

We include tables of the surface simplification results below.

	number of face	10000	3000	2000	1000
QEM	Max error	0.029167	0.013684	0.021605	0.011801
	Mean error	0.000047	0.000050	0.000071	0.000104
curvature weighted	Max error	0.060998	0.058773	0.062470	0.065499
	Mean error	0.000120	0.000170	0.000250	0.000356
discrete curvature	Max error	0.081037	0.074495	0.065953	0.106446
	Mean error	0.001061	0.001683	0.002060	0.003539
proposed method	Max error	0.032843	0.025291	0.025309	0.033128
	Mean error	0.000027	0.000155	0.000228	0.000366

TABLE 1. The comparison of max error and mean error - Fandisk

	number of face	500	300	200	100
QEM	Max error	1.786316	2.256653	2.694252	5.669952
	Mean error	0	0	0	0.001747
curvature weighted	Max error	1.437599	2.266510	3.727547	6.090446
	Mean error	0	0	0	0.000183
discrete curvature	Max error	2.923737	3.839912	5.644363	14.069649
	Mean error	0.000775	0.002412	0.004969	0.013689
proposed method	Max error	1.289047	2.346832	3.196304	5.930771
	Mean error	0	0	0	0.002098

TABLE 2. The comparison of max error and mean error - Sphere

	number of face	1000	500	400	300
QEM	Max error	0.027782	0.037439	0.037439	0.044707
	Mean error	0.004748	0.008156	0.009769	0.011489
curvature weighted	Max error	0.056871	0.059927	0.070074	0.077160
	Mean error	0.005876	0.010211	0.012916	0.017208
discrete curvature	Max error	0.057034	0.086217	0.072263	0.096554
	Mean error	0.006867	0.012343	0.014608	0.016739
proposed method	Max error	0.100730	0.140226	0.140549	0.221994
	Mean error	0.011295	0.018174	0.022327	0.030334

TABLE 3. The comparison of max error and mean error - Cow

	number of face	5000	3000	2000
QEM	Max error	0.192862	0.204640	0.186159
	Mean error	0.001601	0.004256	0.007477
curvature weighted	Max error	0.266027	0.179869	0.105769
	Mean error	0.002882	0.006487	0.011577
discrete curvature	Max error	0.018842	0.059707	0.071663
	Mean error	0.002049	0.005523	0.010006
proposed method	Max error	0.194190	0.194190	0.189809
	Mean error	0.007813	0.012912	0.019028

TABLE 4. The comparison of max error and mean error - big-porsche

REFERENCES

- [1] Mario Botsch, Mark Pauly, Christian Rossl, Stephan Bischoff and Lief Kobbelt. *Geometric Modeling Based on Triangle Meshes*. Siggraph, '06 ACM SIGGRAPH 2006 Courses, Article No.1
- [2] Dobrina Boltcheva and Bruno Levy. *Surface reconstruction by computing restricted Voronoi cells in parallel*. Computer-Aided Design, Volume 90, September 2017, Pages 123-134
- [3] Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F. and Ranzuglia, G. *MeshLab: 3D triangular meshes processing and editing*. Astrophysics Source Code Library, record ascl:1709.003, September 2017
- [4] Michael Garland and Paul S. Heckbert. *Surface Simplification Using Quadric Error Metrics*. In SIGGRAPH 97 Conference Proceedings, pages 209-216, 1997
- [5] Jong B. -S, Teng J. L. and Yang. W. H. *An efficient and low-error mesh simplification method based on torsion detection*. The Visual Computer, 2006, 22(1): 56-67
- [6] S. -J. Kim, C.-H. Kim and D. Levin. *Surface simplification with discrete curvature norm*. In The Third Israel-Korea Binational Conference on geometric Modeling and Computer graphics, Seoul, Korea., October 2001
- [7] Li L, He M. and Wang P. *Mesh Simplification Algorithm Based on Absolute Curvature-weighted Quadric Error Metrics*. Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications, New York, USA: Association for Computing Machinery, 2010: 399-403
- [8] Chang M. and Park S.C. *Range data simplification for reverse engineering*. The International Journal of Advanced Manufacturing Technology, March 2009;41(1) : 86-96
- [9] L. Yao, S. Huang, H. Xu, and P. Li. *Quadratic Error Metric Mesh Simplification Algorithm Based on Discrete Curvature*. Mathematical Problems in Engineering, Vol. 2015

- [10] P. Cignoni, C. Rocchini, and R. Scopigno *Metro: measuring error on simplified surfaces*. Computer Graphics Forum, 1998, 17(2), pp. 167-174
- [11] Hoppe, H. *Progressive meshes*. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, August 1996, pp. 99-108. DOI 10.1145/237170.237216 (1996)
- [12] Gieng, T., Hamann, B., Joy, K., Schussman, G. and Trotts, I. *Constructing hierarchies for triangle meshes*. IEEE Trans Visual Comput Graph 4(2):145-161, DOI 10.1109/2945.694956 (1998)
- [13] Williamn Schroeder, Jontatiau Zarge, and Williatt Lorensen. *triangle meshes*. Computer Graphics (SIGGRAPH 92 Proceedings). 26(2):65-70. July 1992.
- [14] X. Jiao and P. Alexander *Parallel feature-preserving mesh smoothing*. In Proc. Int. Conf. Comput. Sci. Appl., pages 1180-1189, 2005
- [15] G. Peyre *The Numerical Tours of Signal Processing*. Advanced Computational Signal and Image Processing IEEE Computing in Science and Engineering, Vol. 13(4), pp-94,97, 2011