

## Research Article

# Fast Mesh Simplification Method for Three-Dimensional Geometric Models with Feature-Preserving Efficiency

Yongzhi Wang , Jianwen Zheng, and Hui Wang

*School of Architectural and Surveying & Mapping Engineering, Jiangxi University of Science and Technology, Ganzhou, China*

Correspondence should be addressed to Yongzhi Wang; [gisstarww@qq.com](mailto:gisstarww@qq.com)

Received 12 September 2018; Revised 13 December 2018; Accepted 31 December 2018; Published 3 February 2019

Academic Editor: Manuel E. Acacio Sanchez

Copyright © 2019 Yongzhi Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To avoid excessive details, thus omitting less important content, of three-dimensional (3D) geometric models, this study proposes a fast mesh simplification method based on an energy-operator for 3D geometric models with salient feature-preserving efficiency. The energy-operator can evaluate the smoothness and complexity of the regional mesh in 3D models. Accordingly, it can be directly used to simultaneously reduce the candidate triangle and its three neighboring triangles. The proposed method can dramatically collapse the excessive details in relatively smooth areas and preserve more important salient features during the simplification process. It can also maintain a trade-off between time efficiency and salient feature-preserving accuracy. The effectiveness and efficiency of the new method are demonstrated by comparing it with OpenMesh, which is considered the most popular mesh operation software and is capable of achieving accurate mesh simplification models. The new mesh simplification method based on the energy-operator can provide accurate and concise models for interactive 3D rendering, calculating, simulating, and analyzing.

## 1. Introduction

Along with the development of high-resolution data acquisition techniques, such as three-dimensional (3D) laser scanning, 3D geometric models have become increasingly more complex [1–4]. However, such complex 3D geometric models are not always needed. Due to the limitations in graphics hardware, these models are all difficult to interactively render, calculate, simulate, and analyze [2, 3, 5–8]. In addition to improving the capacity of graphics hardware, it is essential to develop a highly efficient mesh simplification method to avoid excessive details of 3D geometric models [7, 9–15].

Mesh simplification is widely used in many industrial areas, such as engineering applications, 3D printing, and cultural heritage protection [3, 16, 17]. For engineering applications, in order to improve the efficiency in dynamically simulating complex CAD models, an efficient mesh simplification method is required. Meanwhile, to meet the simulation requirements of engineering precision, simplified models should preserve main features of

original models. For 3D printing, in order to print models faster, mesh simplification can be used to generate sparse wireframe models from the complex solid models. For the cultural heritage protection, the volume of cultural heritage models is extremely huge and poses a serious challenge for visualization. Mesh simplification is an important method to reduce the triangle (or other geometrical cells) numbers. Thus, the highly efficient mesh simplification method is becoming an essential issue for many industrial areas.

By studying the related work (Section 2) adequately, we propose a faster mesh simplification method for 3D geometric models with salient feature-preserving efficiency. An energy-operator is used to calculate the complexity of triangles, which is described in Section 3.2. The energy-operator accounts for the geometrical characters of the neighboring triangles and collapses three edges of a specific triangle in one contraction process. The new method is considered a trade-off for time efficiency and salient feature preservation. Section 3 introduces the methodology of the new method, which is applied to simplify three popular 3D

geometric models in Case Study (Section 4). In Section 5, we discuss and demonstrate the efficiency and validation of the new simplification method in comparison with OpenMesh.

## 2. Related Work

Various methods have been proposed for mesh simplification, and these can be divided into three main categories: vertex decimation, vertex clustering, and iterative edge contraction [5–8, 18–20]. Vertex decimation methods involve three major steps: selection of candidate vertexes that satisfy a distance or an angle criterion, removal of loop triangles around the candidate vertexes, and triangulation of the resulting holes [9, 18, 20–22]. Although these methods are time-efficient, the candidate vertex selection criterion is based on a local error estimate and therefore cannot guarantee the boundary errors. Moreover, they can only preserve positions, and few salient features are preserved during simplification [7, 19, 22].

Vertex clustering methods use grid structures to divide the original model into clusters and then compute a new vertex to represent each cluster [23–25]. These methods are time-efficient and have a high data reduction rate. However, grid size often results in a geometric error bound; as a result, the simplified mesh has very low quality and cannot preserve more salient features [6, 19].

For the iterative edge contraction methods, the main idea is to collapse edges into new vertexes iteratively [6, 26–28]. The essential difference between these methods lies in how they choose an edge to contract [6]. Therefore, they need to develop error metrics to determine the priority of collapse [7]. Quadric error metric (QEM) is characterized by high computational efficiency and can produce high-quality simplification models [18, 28]. Thus, the QEM-based iterative edge contraction method is considered a state-of-the-art quality-oriented simplification method [11, 29]. Many mesh processing applications and open sources employ the QEM method to improve the simplification result. For example, the most popular open-source software, OpenMesh, uses QEM as an error criterion and achieves an accurate mesh simplification method [7, 30].

However, the QEM-based iterative edge contraction method still has some weaknesses. QEM measurement is only based on the Euclidian distance between a certain vertex and its planes that meet at this vertex; it does not consider the geometric characters of the neighboring triangles. Thus, QEM-based methods may preserve geometric features only to a certain extent, i.e., the simplified mesh shows a sparsity of salient features but shows high density in smoother regions [1, 6]. In addition, QEM-based iterative edge contraction methods can only collapse two triangles that share the same edge in one vertex pair contraction process. This convergence speed is lower than that in the vertex decimation and vertex clustering methods. Thus, maintaining a trade-off between time efficiency and salient feature preservation for the current mesh simplification methods continues to be a challenge [31].

## 3. Methodology

*3.1. Basic Idea and Overall Design.* Researchers must know the geometric features around the candidate vertexes, edges, or triangles during the simplification process. If the neighboring features can be used in the simplification process, then the simplified meshes can maintain relatively dense triangles in the salient feature region and relatively sparse triangles in the smooth region. Thus, more salient features can be preserved efficiently while high-volume unimportant details, such as excessive triangles in the smooth or flat regions, are deleted. Hoppe et al. described an energy-minimization technique for generating an optimal mesh by reducing the number of vertexes from the original mesh [5, 26].

In the energy-minimization technique, the distance from the candidate vertexes to the mesh and the number of vertexes in the mesh is used to define the energy function, named as distance energy. If vertexes from a dense mesh are needed to be removed, we can just maintain the distance energy value. If some vertexes in the original mesh are removed, the edges sharing these vertexes need to be collapsed, and then a simplified model can be achieved. However, the energy-minimization technique treats every vertex equally and ignores the neighborhood complexity. Thus, geometric features preserved in the simplified model cannot maintain reasonable levels of detail depending on the complexity of the original models. The mesh fairing methods, such as the discrete fairing method, can be used to evaluate the uniformity of mesh vertex distribution [32–34].

Using energy-minimization and mesh fairing methods for reference, we present an energy-operator that can account for the surrounding geometric information to judge the complexity of the candidates. In addition, we can collapse triangles (named as candidate triangles) in which the energy value is lower than the threshold. Triangles sharing one of the edges of the candidate triangles will be collapsed simultaneously. Thus, what can be constructed is a new mesh simplification method capable of maintaining a trade-off between time efficiency and salient feature-preserving accuracy.

The overall design of the new method involves three major steps (Figure 1). First, define and calculate the energy-operator. Second, decimate triangles and calculate new vertexes based on the energy-operator. Finally, reconstruct the topology.

### 3.2. Definition and Calculation of the Energy-Operator.

We present the concept of an energy-operator to preserve more salient regional features of the simplified 3D geometric models and to introduce it in the proposed simplification method. Through the statistical analysis of the geometric information (e.g., distance) of the first- and second-order neighbors of 3D model vertexes, the energy-operator can evaluate the smoothness and complexity of the regional mesh in the 3D models. Then, we can dramatically collapse the relatively smooth region and preserve more salient and

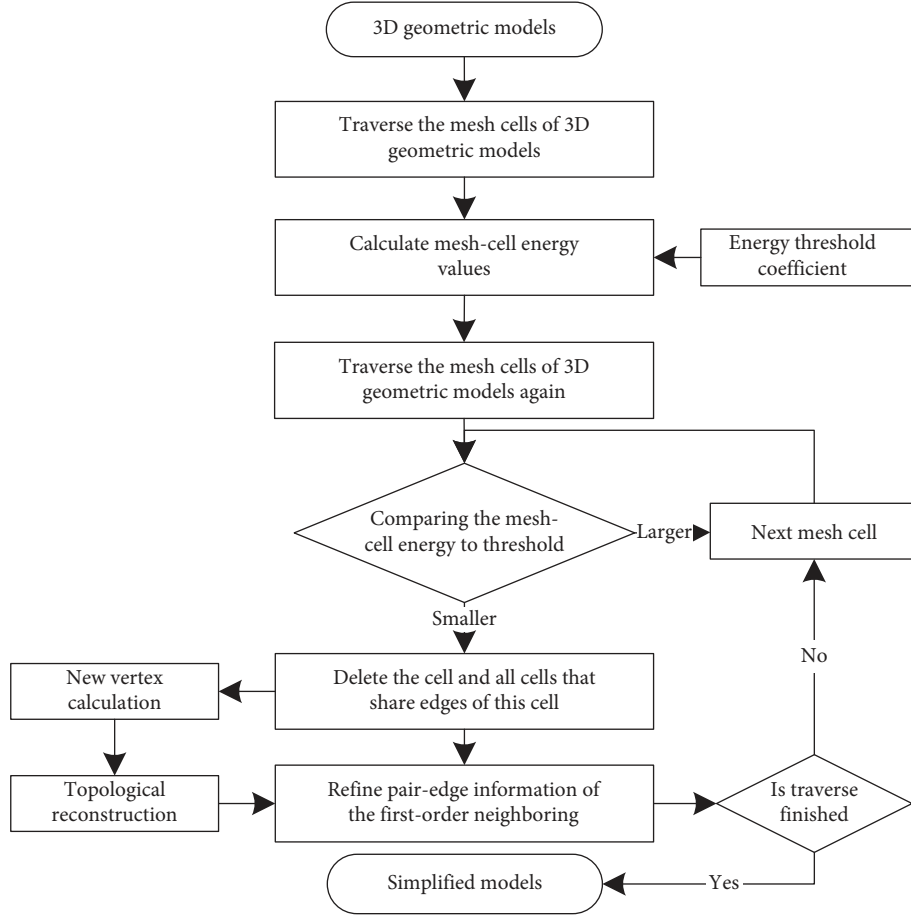


FIGURE 1: Flow chart of the new mesh simplification method.

complex features of the mesh during the simplification process.

The two types of energy-operator are energy-operators of the mesh vertex and those of the mesh cell. The mesh-vertex energy-operator can be calculated using equation (1), while the mesh-cell energy-operator is considered the mean mesh-vertex energy value.

$$Ef(v) = \left| \frac{\sum_{j=1}^{P_v} (v - v_j) \times \bar{L}_j}{\sum_{j=1}^{P_v} \bar{L}_j} \right|, \quad (1)$$

$$\bar{L}_j = \frac{\sum_{j=1}^{P_{v_j}} L_j}{P_{v_j}}, \quad (2)$$

where,  $v$  indicates the candidate vertex,  $P_v$  is the number of edges that shares the same vertex  $v$ , and  $P_{v_j}$  is the in-degree of vertex,  $V_j$ , which is among the first-order neighbors of vertex  $v$ .  $\bar{L}_j$  is the mean length of all ingoing edges of the vertex  $V_j$ .  $\bar{L}_j$  can be calculated using equation (2).

The calculation process of energy-operator values in a geometric model is described as following. First, in traversing the mesh cells, we need to calculate and record the mesh-cell energy value. This record is a required preparatory step in determining the energy threshold. Second, we need to construct a mapping set between mesh cells and their energy

values, which can be organized in key-value pairs in a necessary data structure for follow-up simplification. Third, to avoid duplication in energy value calculation of vertexes often shared by different cells in 3D meshes, we need to construct another mapping set to record the relationship between mesh vertexes and their energy. For convenience in energy calculation of the mesh vertexes and cells, we introduce the HalfEdge data structure into the simplification process [30, 35].

**3.3. Triangle Decimation and New Vertex Calculation.** As described in Section 3.2, the energy-operator can evaluate the smoothness and complexity of the regional mesh in the 3D models. Hence, this study utilizes the energy-operator to realize the triangle (triangles are used to represent the mesh cells because they are popular mesh cells in current 3D models) decimation operation, which can preserve more salient features. The process of triangle decimation is described as follows: traverse the triangle set and calculate every vertex energy value and triangle energy value. If the triangle energy value is smaller than the energy threshold, then this triangle is in a relatively smooth region and can be collapsed.

Calculating the location of the new vertexes after triangle decimation is also an important step in the entire

simplification method. When a candidate triangle is decimated, it will become a new mesh vertex. According to the principle of triangle decimation, the new vertex can be calculated simply by equation (3), where coefficient  $W$  is equal to  $1/3$ , and “ $P_2$ ,” “ $P_3$ ,” and “ $P_5$ ” are vertexes of the candidate triangle to be collapsed, named as “Cell,” and “ $P$ ” is the new vertex after the “Cell” is decimated, as shown in Figure 2.

$$P = W(P_2 + P_3 + P_5). \quad (3)$$

Calculating the coordinates of the new mesh vertex according to the surrounding structure information of the decimated triangle is important to preserve the salient features of the original 3D models. The butterfly subdivision scheme fully considers the geometric and topologic connections of the regional mesh when calculating new mesh vertexes [36], thus enabling the simplified mesh to preserve more salient regional features and be smoother. By considering the principle of triangle decimation and butterfly subdivision scheme, we present a new mesh vertex calculation (equation (4)). Where “ $P_1$ ,” “ $P_4$ ,” and “ $P_6$ ” are the vertexes of neighborhood triangles that share one edge of the “Cell,” as shown in Figure 2. And the sum of the coefficients  $W_1$  and  $W_2$  is equal to  $1/3$  too, which obeys the primary principle of triangle decimation. We suggest that  $W_1$  and  $W_2$  are set as  $4/9$  and  $-1/9$ , respectively, to get better calculation results. When the new mesh vertex calculation is used, all triangles that share the edges of the candidate triangles are also decimated in the triangle decimation process.

$$P = W_1(P_2 + P_3 + P_5) + W_2(P_1 + P_4 + P_6). \quad (4)$$

**3.4. Topological Reconstruction.** After triangle decimation and new vertex calculation, both the original and new vertexes created must be connected spatially according to certain rules to generate the simplified 3D geometric models. This process is called topological reconstruction. In the HalfEdge data structure, the topological reconstruction rules are described as follows. We use “Cell” to represent the candidate triangle and “Cell A,” “Cell B,” and “Cell C” for the triangles sharing one edge of the “Cell.”

**3.4.1. Reconstruction Rule for Regular Manifold Meshes.** When there are no special mesh structures, such as concurrent vertexes and overlapping cells, we need to decimate triangles “Cell A,” “Cell B,” “Cell C,” and “Cell” and delete energy values related to the decimated values. Meanwhile, the pair-edge information of the remaining first-order neighbor triangles of “Cell” should be updated to maintain consistent spatial topology.

According to the physical mesh structures and differences for updating the pair-edge information, this reconstruction rule can be divided into two types. First, there is at least one cell between each of the triangle, “Cell A,” “Cell B,” and “Cell C,” as shown in Figure 3(a). After

triangle decimation and new vertex (“ $P$ ”) calculation (Figure 3(b)), all first-order neighbor edges need to update their pair-edge information. Second, two triangles share one edge, “Cell A,” “Cell B,” and “Cell C,” as shown in Figure 3(c). Given that no other triangle between “Cell A” and “Cell B” that shares the same edge, we need to update the partial edges’ pair-edge information. For example, in Figure 3(c), we can set edges “ $P_1$ ” and “ $P_4$ ” and “ $P_3$ ” and “ $P_4$ ” to be each other’s pair edge, similar to that for edges “ $P_1$ ” and “ $P_4$ ” and “ $P_3$ ” and “ $P_4$ ”. Figure 3(d) shows the resulting mesh after decimation and calculation. For both rule types, the energy value for related triangles in the reconstructed mesh must be updated.

**3.4.2. Reconstruction Rule for Nonmanifold Meshes with Spatial Overlapping.** The complexity of 3D geometric models causes some spatial overlapping triangles in the nonmanifold meshes. Figure 4 shows two main types of spatial overlapping triangles. The first spatial overlapping type is shown as Figure 4(a), where two triangles “Cell 1” and “Cell 2” are spatially overlapped, and there is no triangle around them. For this type, we can decimate these two triangles directly. The second spatial overlapping type is shown in Figures 4(b) and 4(c), where at least one edge of the two spatial overlapping triangles “Cell 1” and “Cell 2” is shared by other triangles. For this type, after deleting the two spatially overlapping triangles, “Cell 1” and “Cell 2,” we also need to update the pair-edge information (e.g., “ $P_1P_2$ ” and “ $P_2P_3$ ”) shared by other triangles.

## 4. Case Study

Three famous 3D geometric models, Stanford Bunny (Figure 5(a)), Skeleton Hand (Figure 6(a)), and Happy Buddha (Figure 7(a)), are used as test cases to demonstrate the validity of the new mesh simplification method presented in this paper. The case study environment is a PC with an Intel(R)/Core(TM)/i7-4700 U/3.40 GHZ CPU and a 16G random access memory. By using the new simplification method, we generate simplified 3D geometric models with different levels of detail information, as shown in Figures 5(b)–5(f) for Stanford Bunny, Figures 6(b)–6(f) for Skeleton Hand, and Figures 7(b)–7(f) for Happy Buddha. The information of triangle numbers and simplification time for these models is shown in Table 1. Through these study cases, we discover that the new simplification method can efficiently preserve the most important salient features of the original models. For example, by comparing the original model (Figure 5(a)) with the simplified model (Figure 5(d)), we find out that the detail salient features are still obvious even though the number of triangles is decreased from 55,903 to 7,795. By contrast, the new method is very time-efficient because it only takes 1.252 s to contract the original model into a simplified model with 48,108 triangles. The same situation happens to the models of Skeleton Hand and Happy Buddha. Thus, the new simplification method can maintain a trade-off between time efficiency and salient feature preservation.

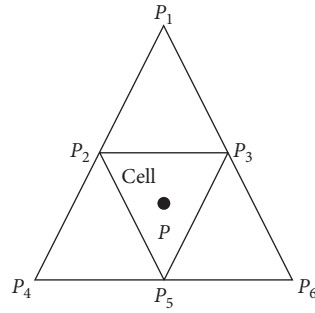


FIGURE 2: New mesh vertex calculation illustration.

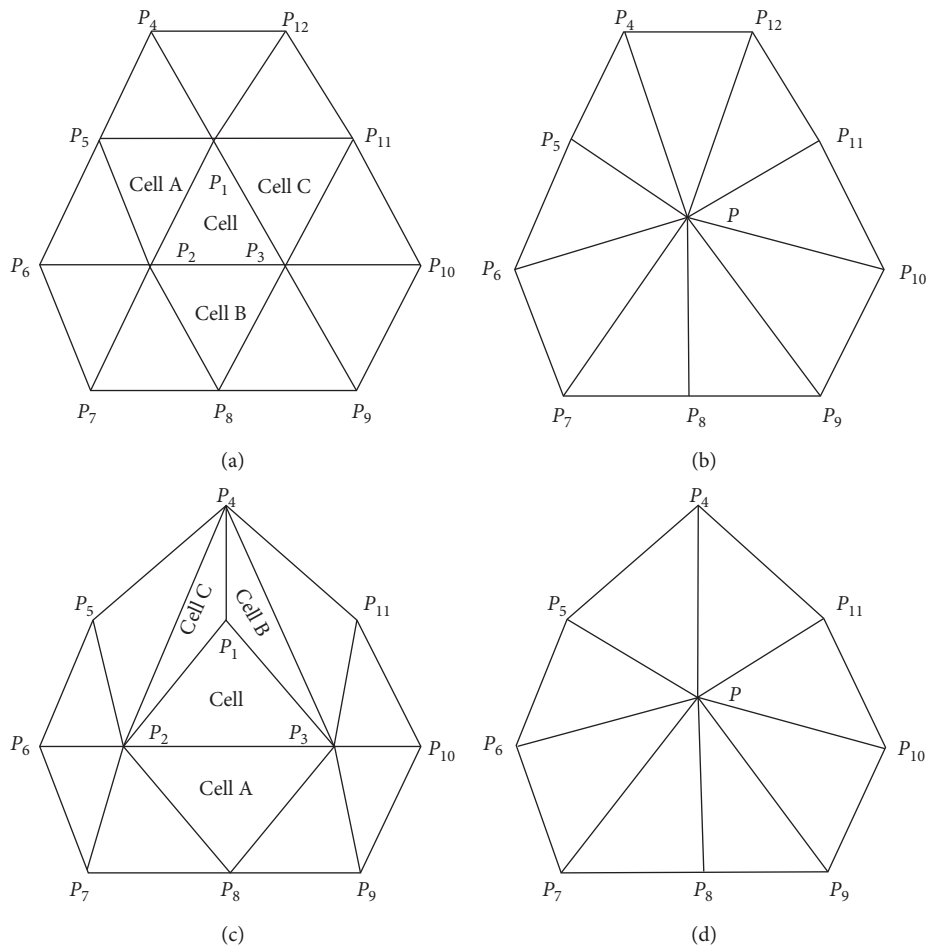


FIGURE 3: Reconstruction rule for regular manifold meshes. (a) The first reconstruction rule type. (b) The pair-edge information of all first-order neighbor edges needs to be updated. (c) The second reconstruction rule type. (d) The partial edges' pair-edge information needs to be updated.

## 5. Discussion

**5.1. Efficiency of the Energy-Operator.** Comparing with the energy-minimization which is used by Hoppe et al. [26], the energy-operator is more efficient in triangle decimation. The energy-operator can evaluate the uniformity of mesh vertex distribution. Instead of only considering the distance between a certain vertex and its planes that meet at this vertex, the geometric characters of the neighboring triangles is also considered. It means that the energy-operator can judge the

complexity of the candidate triangles based on their surrounding geometric characters. Thus, the energy-operator can preserve salient geometric features in various levels of detail according to the complexity of models.

**5.2. Effectiveness of the New Method: A Comparison.** We compare the new simplification method with OpenMesh, one of the most popular computational geometry software, to further demonstrate the effectiveness and efficiency of the



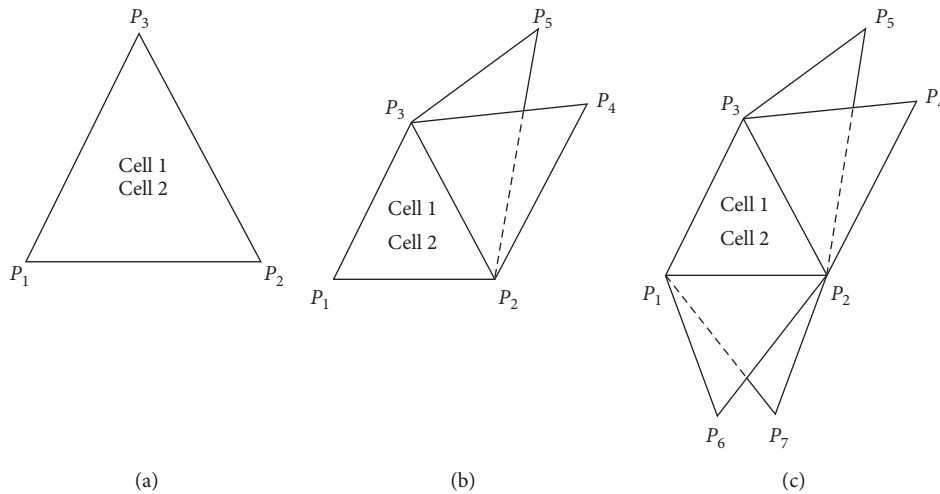


FIGURE 4: Spatial overlapping meshes. (a) The first spatial overlapping type. The second spatial overlapping type in forms (b) and (c).

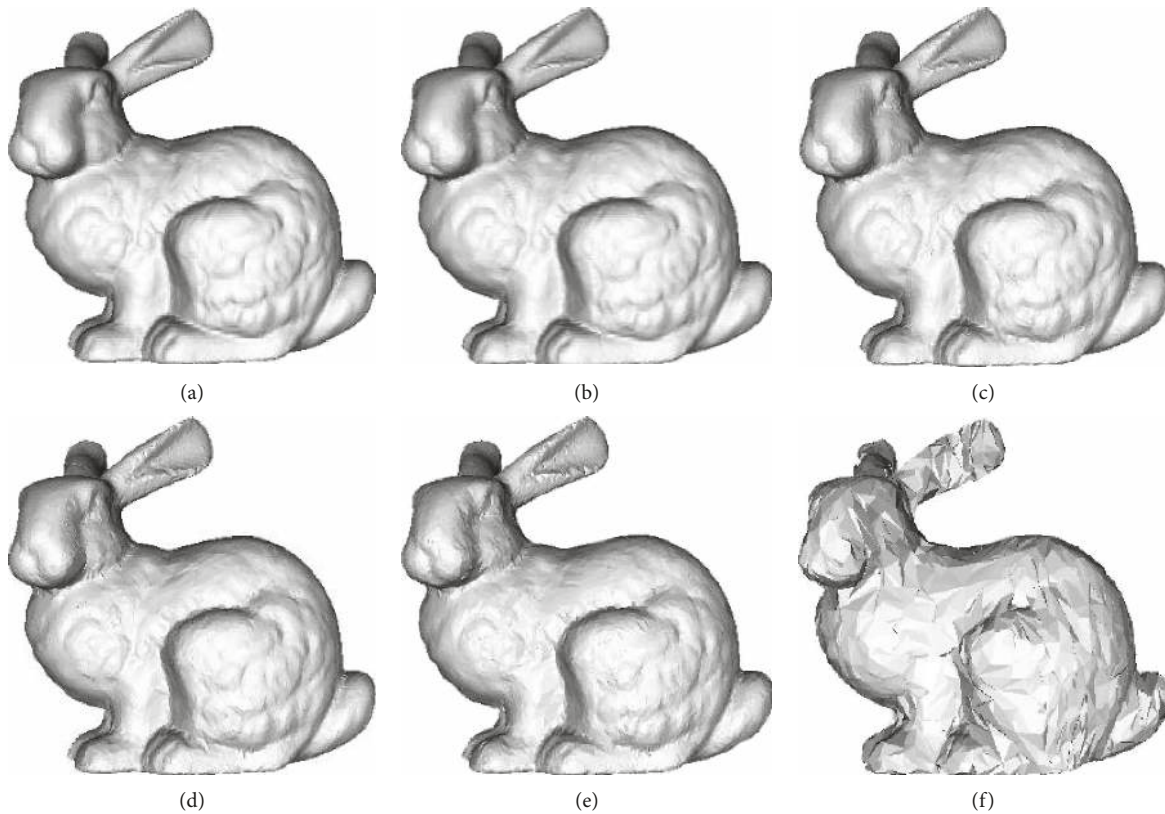


FIGURE 5: Original bunny model and simplified models with different levels of detail information. (a) Original model with 69,451 triangles. Simplified models with (b) 55,903 triangles, (c) 41,611 triangles, (d) 30,799 triangles, (e) 20,735 triangles, and (f) 7,795 triangles.

new simplification method. OpenMesh utilizes the QEM method based on HalfEdge, the basic data structure for the new simplification method, which makes it reasonable to compare the two methods [30].

The comparison environment is the same as the case study one. We use the dragon model from the Stanford University Computer Graphics Laboratory as the comparison data. The original dragon model contains 47,800

triangles and 22,998 vertices, and its size is 1,626 kb, as shown in Figure 8.

We use the new simplification method and OpenMesh to simplify the dragon model separately and obtain simplified models with different levels of detail, as shown in Figure 9. The first and second columns are the simplified models generated by the new method with different energy thresholds, which will be discussed in Section 5.3. In

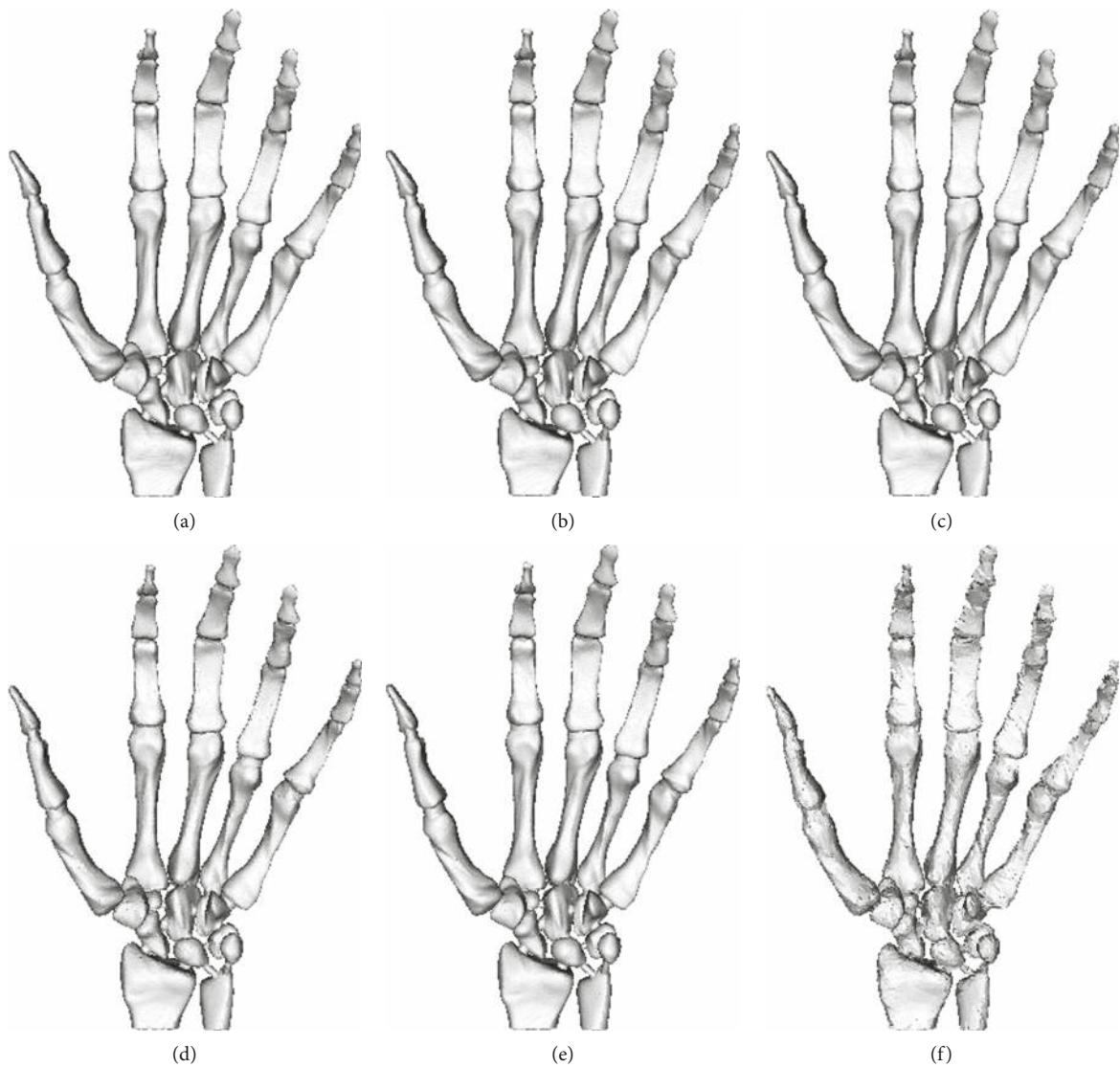


FIGURE 6: Original hand model and simplified models with different levels of detail information. (a) Original model with 654,666 triangles. Simplified models with (b) 547,422 triangles, (c) 456,702 triangles, (d) 355,586 triangles, (e) 186,814 triangles, and (f) 55,302 triangles.

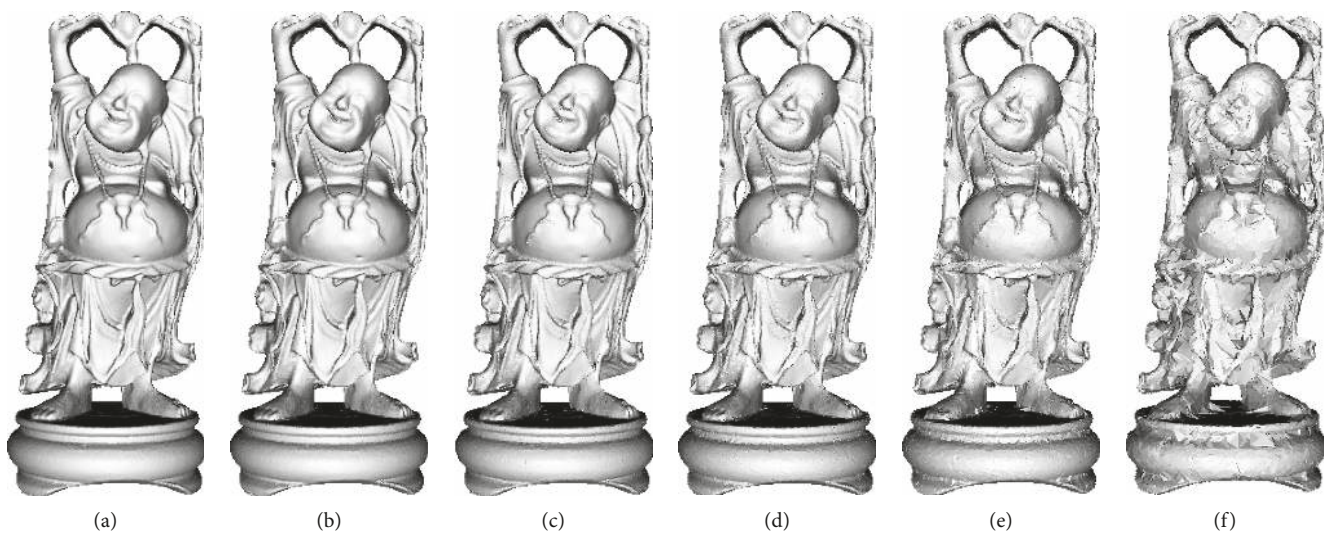


FIGURE 7: Original happy model and simplified models with different levels of detail information. (a) Original model with 1,087,716 triangles. Simplified models with (b) 672,148 triangles, (c) 486,436 triangles, (d) 269,384 triangles, (e) 156,456 triangles, and (f) 98,664 triangles.

TABLE 1: Statistics of triangle numbers and simplification time.

Bunny	Triangles	55,903	41,611	30,799	20,735	7,795
	Time (s)	0.271	0.39	0.444	0.45	0.478
Hand	Triangles	547,422	456,702	355,586	186,814	55,302
	Time (s)	85.194	97.792	106.762	107.713	112.165
Happy	Triangles	672,148	486,436	269,384	156,456	98,664
	Time (s)	13.852	21.975	28.811	35.142	35.694

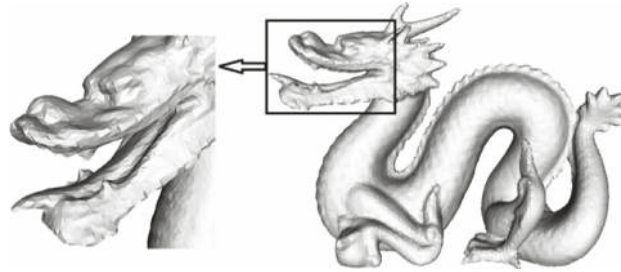


FIGURE 8: Original dragon model with 47,800 triangles (left is the zoom-in view of the head part).

addition, the first column comprises the head part with a zoom-in view of the entire model, which is in the second column. The third (zoom-in view) and fourth (entire view) columns are the simplified models generated by OpenMesh. The time consumption and thresholds for the two methods are shown in Table 2. The comparison of the simplified models and time consumption of these two methods reveal that both the new method and OpenMesh can preserve important salient features effectively. However, the new method is more time-efficient than OpenMesh. Thus, the new mesh simplification method based on the energy-operator can maintain the trade-off between time efficiency and salient feature-preserving accuracy.

Furthermore, the statistical analysis of geometrical information (Table 3) for dragon models simplified by the two methods is performed. Table 3 draws the following conclusions: the new method can contract both triangles and points, so the size of the simplified model reduces quickly; the OpenMesh method can only contract triangles, with vertexes increased for the simplified models, so the size of the simplified model reduces slowly. And it is obvious that the simplified model by OpenMesh is larger than the one (in the same level of detail) simplified by the new method.

**5.3. Value Setting of Energy Threshold.** Energy threshold is an essential basis of the energy-operator and a key factor with respect to the time efficiency and feature-preserving effectiveness of the new simplification method. On the basis of the energy-operator, we recursively calculate each mesh cell's energy value by comparing the energy threshold to determine whether the mesh cell needs to be decimated or preserved. In this study, energy threshold is the value that a special coefficient (energy threshold coefficient, ETC) multiplies by the mean energy value of all cells in the 3D geometric models. The key point in obtaining the

energy threshold is to determine the value of ETC. More cells will be decimated if the ETC value is larger, and vice versa. Thus, choosing suitable ETC values during the simplification process to generate simplified models with different levels of detail information is crucial. Figure 10 shows the simplification results when ETC is set at different values.

**5.4. Potential Limitations of the New Method.** The new simplification method based on the energy-operator can maintain a trade-off between time efficiency and salient feature-preserving accuracy. However, some potential limitations still exist and may occur in topological reconstruction. When 3D geometric models are particularly complex and contain some special spatial topological structures, such as overlapping vertexes, it is difficult to distinguish the exact spatial topological structures by depending only on the first-order neighbors of the candidate cells. This finding requires further research in the future. Another potential limitation is that the new simplification method does not consider feature information such as texture and color in 3D geometric models. In the follow-up research, such feature information should be adapted in the simplification process to improve the ability of salient feature preservation.

## 6. Conclusion

This study proposed a fast mesh simplification method based on energy-operator for 3D geometric models. By using the energy-operator, the smoothness and complexity of the regional mesh in 3D models can be evaluated firstly. We can then dramatically collapse relatively smooth areas of the meshes and preserve more important salient features during the simplification process. Three popular 3D geometric models, Stanford Bunny, Skeleton Hand, and



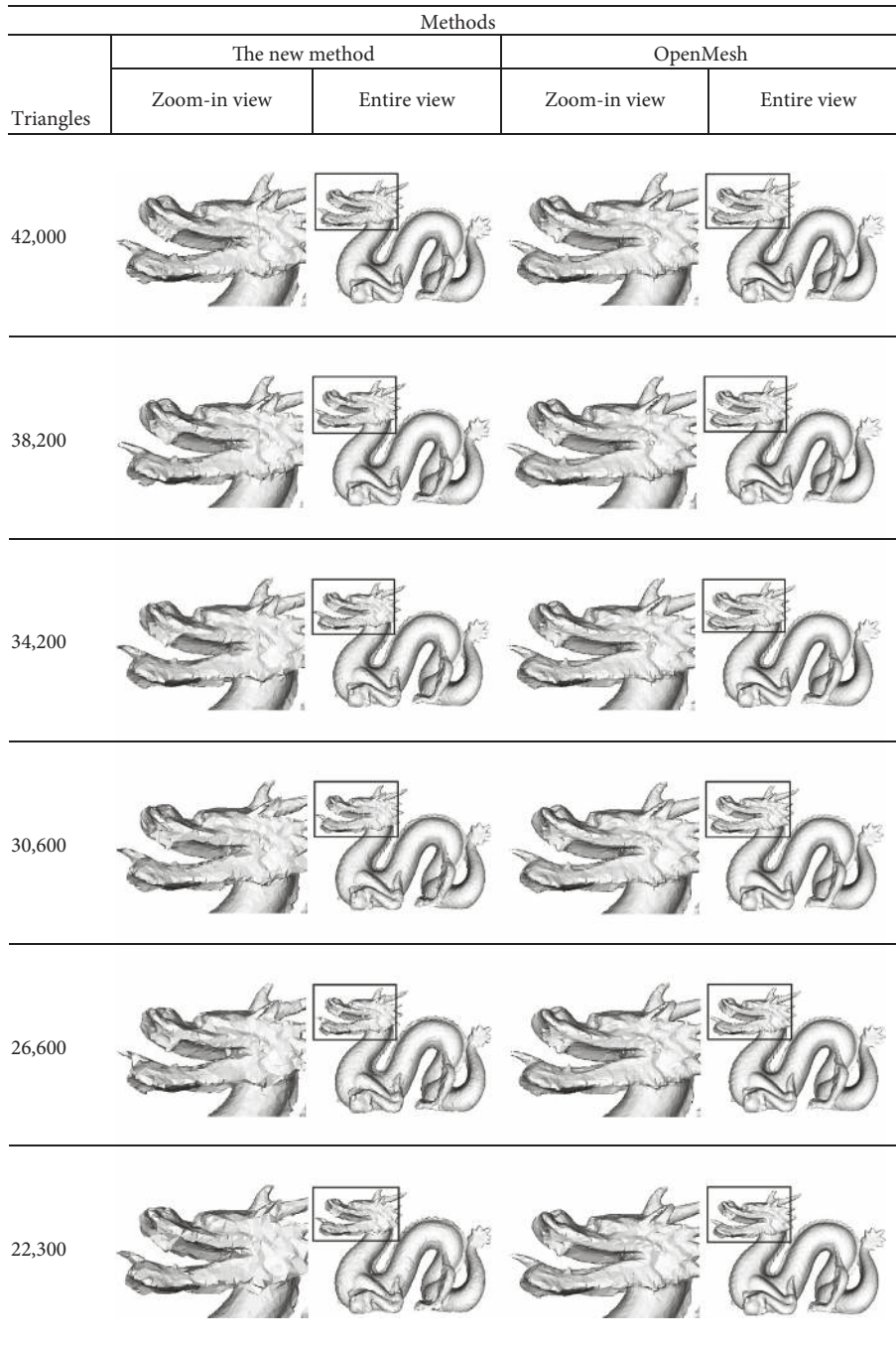


FIGURE 9: Simplified dragon models with different levels of detail by the new method and OpenMesh.

TABLE 2: Statistics of time consumption and thresholds for the new method and OpenMesh.

Simplification method		Triangle numbers					
		42,000	38,200	34,200	30,600	26,600	22,300
The new method	Threshold	0.37	0.514	0.66	0.787	0.95	1.15
	Time (s)	0.102	0.135	0.162	0.184	0.203	0.236
OpenMesh	Threshold	$2E - 14$	$4.5E - 14$	$9E - 14$	$1.6E - 13$	$3E - 13$	$6E - 13$
	Time (s)	3.22	3.224	3.261	3.302	3.354	3.381

TABLE 3: Statistics of geometrical information of dragon models simplified by the two methods.

Triangles remained	The new method		OpenMesh	
	Vertexes remained	Size (kb)	Vertexes remained	Size (kb)
42,000	20,340	1,382	29315	1,825
38,200	18,425	1,249	27371	1,763
34,200	16,455	1,112	25333	1,698
30,600	14,737	991	23502	1,640
26,600	12,777	855	21449	1,575
22,300	10,719	702	19292	1,506

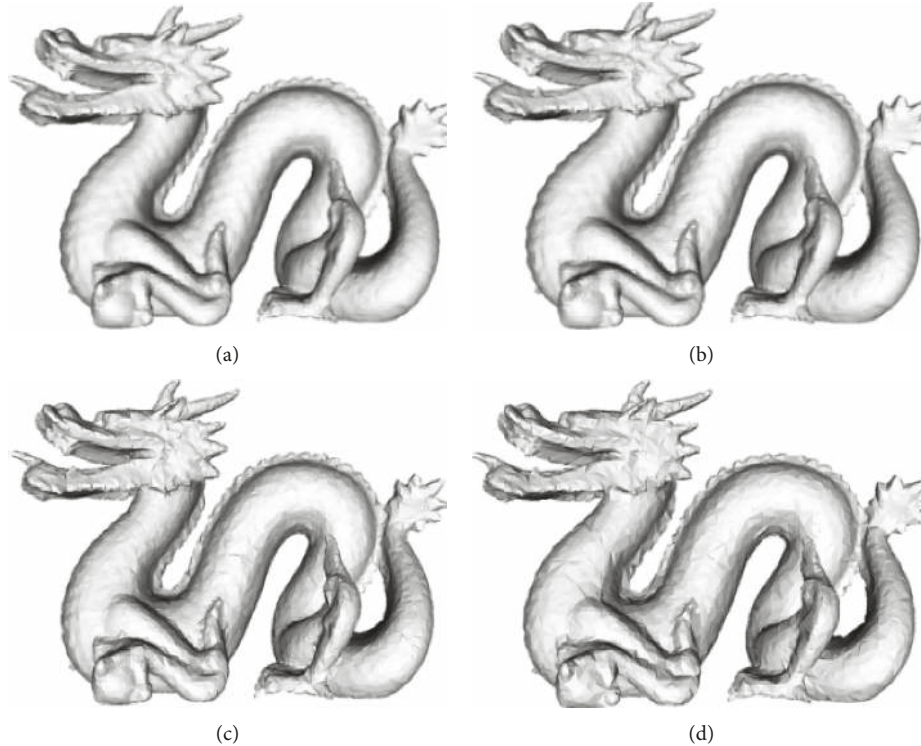


FIGURE 10: Simplified dragon models with different ETC values. Simplified models (a) with 47,790 triangles when ETC is 0.2, (b) with 45,200 triangles when ETC is 0.5, (c) with 33,400 triangles when ETC is 0.8, and (d) with 25,220 triangles when ETC is 1.0.

Happy Buddha, are used as test cases to demonstrate the validity of the new method. The effectiveness and efficiency of the new method are verified by comparing with OpenMesh.

The new method can maintain a trade-off between time efficiency and salient feature-preserving accuracy. Thus, the new method can provide accurate and concise models for many fields, such as interactive 3D rendering and 3D printing and engineering applications. However, when overlapping vertexes exist in 3D geometric models, some potential limitations may occur in topological reconstruction.

In the follow-up research, we will refine the topological reconstruction process, so the new method can deal with special spatial topological structures. And then, we will consider feature information, such as texture and color in 3D geometric models, to improve the ability of salient feature preservation for the new simplification method.

## Data Availability

The data used to support the findings of this study can be found in the Stanford 3D Scanning Repository.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This study was supported by the National Natural Science Foundation of China (No. 41601429), the Natural Science Foundation of Jiangxi Provincial Department of Science and Technology (No. 20171BAB203028), and the Program of Qingjiang Excellent Young Talents, Jiangxi University of Science and Technology. The authors would also like to

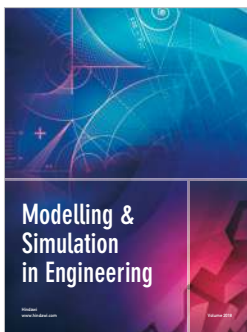
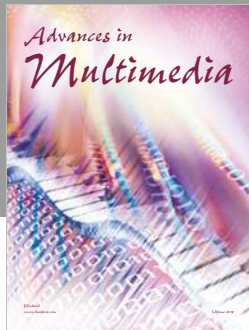
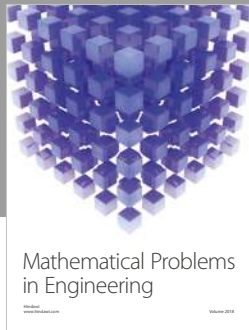
acknowledge the contributions of Jacqueline Wah to the spelling and grammar check for this paper.

## References

- [1] F. S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 2, pp. 191–205, 2003.
- [2] Y. Yuan, R. Wang, J. Huang, Y. Jia, and H. Bao, "Simplified and tessellated mesh for realtime high quality rendering," *Computers and Graphics*, vol. 54, pp. 135–144, 2016.
- [3] X. Liu, L. Lin, J. Wu, W. Wang, B. Yin, and C. C. L. Wang, "Generating sparse self-supporting wireframe models for 3D printing using mesh simplification," *Graphical Models*, vol. 98, pp. 14–23, 2018.
- [4] Z. Yu, H. S. Wong, H. Peng, and Q. Ma, "ASM: an adaptive simplification method for 3D point-based models," *Computer-Aided Design*, vol. 42, no. 7, pp. 598–612, 2010.
- [5] H. Hoppe, "Progressive meshes," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques-SIGGRAPH '96*, pp. 99–108, ACM, New Orleans, LA, USA, August 1996.
- [6] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques-SIGGRAPH '97*, pp. 209–216, ACM, Los Angeles, CA, USA, August 1997.
- [7] H. K. Choi, H. S. Kim, and K. H. Lee, "An improved mesh simplification method using additional attributes with optimal positioning," *International Journal of Advanced Manufacturing Technology*, vol. 50, no. 1–4, pp. 235–252, 2010.
- [8] A. Chattopadhyay, H. Carr, D. Duke, Z. Geng, and O. Saeki, "Multivariate topology simplification," *Computational Geometry*, vol. 58, pp. 1–24, 2016.
- [9] L. Kobbelt, S. Campagna, and H. Seidel, "A general framework for mesh decimation," in *Proceedings of Graphics Interface '98*, pp. 43–50, Vancouver, British Columbia, Canada, June 1998.
- [10] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," in *ACM SIGGRAPH 2004 Papers on-SIGGRAPH '04*, pp. 905–914, New York, NY, USA, 2004.
- [11] M. Hussain, "Efficient simplification methods for generating high quality LODs of 3D meshes," *Journal of Computer Science and Technology*, vol. 24, no. 3, p. 604, 2009.
- [12] J. Chen, M. Li, and J. Li, "An improved texture-related vertex clustering algorithm for model simplification," *Computers and Geosciences*, vol. 83, pp. 37–45, 2015.
- [13] T. Odaker, D. Kranzlmüller, and J. Volkert, "GPU-accelerated real-time mesh simplification using parallel half edge collapses," in *Proceedings of 10th International Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, vol. 9548, pp. 107–118, Telč, Czech Republic, October 2016.
- [14] B. Yang, F. W. B. Li, X. Wang et al., "Visual saliency guided textured model simplification," *Visual Computer*, vol. 32, no. 11, pp. 1415–1432, 2015.
- [15] S. Kwon, D. Mun, B. C. Kim, and S. Han, "Feature shape complexity: a new criterion for the simplification of feature-based 3D CAD models," *International Journal of Advanced Manufacturing Technology*, vol. 88, no. 5–8, pp. 1831–1843, 2016.
- [16] S. Gao, W. Zhao, H. Lin, F. Yang, and X. Chen, "Feature suppression based CAD mesh model simplification," *Computer-Aided Design*, vol. 42, no. 12, pp. 1178–1188, 2010.
- [17] J. Zhang and G. H. Geng, "The application of level-set mesh simplification algorithm in digital cultural heritage," in *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*, pp. 202–205, IEEE, Chongqing, China, May 2016.
- [18] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 65–70, 1992.
- [19] P. Cignoni, C. Montani, and R. Scopigno, "A comparison of mesh simplification algorithms," *Computers and Graphics*, vol. 22, no. 1, pp. 37–54, 1998.
- [20] H. Lee and H. M. Kyung, "Parallel mesh simplification using embedded tree collapsing," *Visual Computer*, vol. 32, no. 6–8, pp. 967–976, 2016.
- [21] M. Soucy and D. Laurendeau, "Multiresolution surface modeling based on hierarchical triangulation," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 1–14, 1996.
- [22] A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno, "Multiresolution decimation based on global error," *Visual Computer*, vol. 13, no. 5, pp. 228–246, 1997.
- [23] J. R. Rossignac and P. Borrel, "Multi-Resolution 3D Approximations for Rendering Complex Scenes," in *IFIP Series on Computer Graphics*, Springer, Berlin, Germany, 1993.
- [24] K. L. Low and T. S. Tan, "Model Simplification Using Vertex-clustering," in *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, ACM, Providence, RI, USA, April 1997.
- [25] D. Luebke and C. Erikson, "View-dependent simplification of arbitrary polygonal environments," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques-SIGGRAPH '97*, pp. 199–208, ACM Press/Addison-Wesley Publishing Co, Los Angeles, CA, USA, August 1997.
- [26] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Mesh optimization," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques-SIGGRAPH '93*, pp. 19–26, ACM, Anaheim, CA, USA, August 1993.
- [27] R. Ronfard and J. Rossignac, "Full-range approximation of triangulated polyhedra," *Computer Graphics Forum*, vol. 15, no. 3, pp. 67–76, 1996.
- [28] P. S. Heckbert and M. Garland, "Optimal triangulation and quadric-based surface simplification," *Computational Geometry*, vol. 14, no. 1, pp. 49–65, 1999.
- [29] T. Boubekur and M. Alexa, "Mesh simplification by stochastic sampling and topological clustering," *Computers and Graphics*, vol. 33, no. 3, pp. 241–249, 2009.
- [30] M. Botsch, S. Steinberg, S. Bischoff, and L. Kobbelt, "Open-Mesh: a generic and efficient polygon mesh data structure," in *Proceedings of OpenSG Symposium*, Darmstadt, Germany, January 2002.
- [31] C. González, J. Gumbau, M. Chover, F. Ramos, and R. Quirós, "User-assisted simplification method for triangle meshes preserving boundaries," *Computer-Aided Design*, vol. 41, no. 12, pp. 1095–1106, 2009.
- [32] Y. X. Zhou, H. B. Cha, and Q. Y. Shi, "Mesh fairing of butterfly subdivision surfaces," *Journal of Image and Graphics*, vol. 8, no. 3, pp. 299–305, 2003.
- [33] L. Kobbelt, "Discrete fairing," in *Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces*, pp. 101–131, ACM, Cirencester, UK, 1996.

- [34] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques-SIGGRAPH '95*, pp. 351–358, ACM, Los Angeles, CA, USA, August 1995.
- [35] Y. Wang, Y. Sheng, X. Liu, and J. Chen, "Research on the method of geological three-dimensional modeling based on subdivision virtual drills," *Journal of Information and Computational Science*, vol. 11, no. 18, pp. 6713–6724, 2014.
- [36] N. Dyn, D. Levine, and J. A. Gregory, "A butterfly subdivision scheme for surface interpolation with tension control," *ACM Transactions on Graphics*, vol. 9, no. 2, pp. 160–169, 1990.






**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

