Improvement of Mesh Simplification Using Normal Vector Diversity

Masoud Ebadi [1] , Kourosh Kiani [2] and Razieh Rastgoo [3]

***Abstract*-- 3D mesh simplification is an important challenge in various fields. While different simplification methods have been proposed in recent years, the focus has shifted to keeping properties such as ridges and valleys along with mesh simplification. While most of the proposed models have used curvature, some challenges exist, such as the computational complexity and sensitivity to the neighborhood size. The latter can be solved by averaging several neighborhoods. This paper proposes a simple yet fast method with less sensitivity to the neighborhood size. To this end, we use the normal vector and the parameters of a probability distribution of its variations to detect the elevations, depressions (geometrical changes), and curve parts. We combine this method with the Quadric Error Metric (QEM) method to produce a hybrid method for 3D mesh simplification, preserving its elevations and depressions. Evaluation results show that our method has a lower error than the other methods.**

***Index Terms*-- 3D Mesh simplification, Normal vector, Curvature, Improvement, Sensitivity.**

I. INTRODUCTION

Nowadays, 3D models are used in various fields, such as virtual reality (VR) [1], heterogeneous materials [2], and Infrastructural Work [3]. Different high complexity methods are used to provide 3D models in each field. For example, in some computer games, many complicated mathematical calculations are necessary to have an accurate collision detection of some objects. These calculations can be able to perform on less detailed models. Therefore, a suitable simplification method is required to reduce the model complexity. In general, simplification methods decrease the calculations, memory usage, and volume of the transmitted data while keeping the quality of the results. Different researchers have proposed different simplification methods [4-28]. These methods can be categorized into two general groups: sensitive to feature and insensitive to feature methods. While the sensitive feature methods consider special priorities for different features of the model, the insensitive feature methods act equally on all parts of the model without special attention to different features. The latter only considers an error measurement function to control the operation and keeps the simplified model similar to the reference. These methods maintain special properties, like curvature, by directing and controlling operations during simplification. The basic algorithm of both categories is similar, but the first category models act somewhat intelligently to increase or decrease the level of simplification. One of the solutions to control the level of simplification is the detection of the important regions and saliency in two-dimensional images, which generally state that regions with lots of changes in light intensity are more important for the human visual system. However, the light intensity is substituted by the geometrical data in 3D models. So, regions with lots of geometrical changes are more important from the point of view of the human visual system. This way, finding the correlation of several features with regions selected by humans is used to find the important regions in 3D models [18]. The curvature feature is the most important among other features. This feature is usually used for determining geometrical changes. In line with this, in this paper, we investigate the curvature problem and propose a simplification method for determining the regions of elevation and depression by using the normal vector of the vertex.

The rest of this paper is organized as follows: In section 2, normal diversity expression is introduced. Related work and the proposed method are presented in sections 3 and 4, respectively. Section 5 demonstrates the experimental results of the proposed method. Finally, the conclusions are drawn in section 6.

II. NORMAL DIVERSITY EXPRESSION

We aim to search the regions that are more visually important and assist in reducing the damage to the sensitive mesh regions. There are several methods to detect the considered regions. The most common way among them is based on the curvature [16], in which the amount of geometrical changes in a region is the criterion to measure the region's importance. This procedure comes from 2D image processing, where more light intensity changes may be more important to

1: M. Ebadi is with the Electrical and Computer Engineering Faculty Semnan University, Semnan, Iran
2: K. Kiani is with the Faculty of Electrical and Computer Engineering, Semnan University, Semnan, Iran.
3. R. Rastgoo is with the Electrical and Computer Engineering Faculty Semnan University, Semnan, Iran
Corresponding author: kourosh.kiani@semnan.ac.ir

the eyes. This way, we first study the curvature-based methods and the corresponding concepts.

The curvature is one of the geometrical properties representing a deviation from a flat surface in a specific direction. This is a local feature and varies for different points of the object. From a mathematical point of view, for a 2D curve F(t) (that is, in parametric form), the curvature of the point K(t) is equal to the ratio of angle changes of the normal vector of the curve (between the considered point and a point in its vicinity) to over the curve length between two points (Fig. 1). It can be calculated as (1):

$$F(t) = g(t) * \vec{i} + h(t) * \vec{j} \quad (1)$$

$$T(t) = \frac{F'(t)}{|F(t)|}$$

$$K(t) = \frac{T'(t)}{|F'(t)|}$$

Where $g(t)$ and $h(t)$ are the parametric functions of i and j. F'(t) is the parametric derivation, T(t) is the tangent vector on the curve, and T'(t) is the derivation of the curve. For 3D surfaces, the surface is cut in one direction, and the curvature is calculated for the resulting curve. The results are obtained in different numbers in different directions. The smallest and biggest numbers are called min and max curvature, respectively.

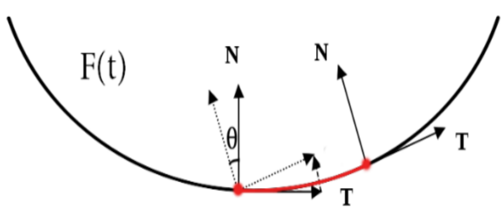

Fig. 1. The curve of F(t) and angle changes of a normal vector.

Two common methods are used to calculate curvature in the mesh where space is discrete: The first group includes methods that initially estimate the surface using a continuous function, then using the second derivation to compute the curvature. The second group estimates the second derivation and curvature directly without the continuous function estimation and using the geometrical data of the mesh such as face normal vector or face angle. One of the advantages of the second group is high performance. However, they also have larger errors. Curvature calculation needs a neighborhood for each point. The curvature determines the deflection amount of the surface at each point, but we are searching for elevation and depression in an area. Averaging in a ring is usually used to achieve this result. The size of the ring is important. A smaller ring gives more details, while more local and larger rings result in more global information. Determining the appropriate ring size also affects the correct determination of saliency. So, in addition to the neighborhood of the curvature calculation, the averaging also needs the neighborhood rings.

III. Related work

One well-known proposed work in mesh saliency is [16] where the mean curvature is calculated for each point using a neighborhood (N(v,σ)). Then, a weighted mean for each vertex is calculated in another neighborhood (G(ϑ,σ)) to reduce the effect of the size of N(v,σ). The weights are calculated based on a Gaussian neighborhood centered on the considered vertex. After that, two neighborhoods (G(ϑ,σ)) with a little difference in size are selected and the curvature is subtracted to calculate the saliency. The variance or width of this Gaussian neighborhood affects saliency results. The larger neighborhood results in more continuous regions as shown in Fig. 2. Finally, to solve this problem, saliency is calculated in different sizes and after normalizing the results, they are added to each other as shown in Fig. 2. Therefore, determining the elevation and depression regions using curvature is both costly and time-consuming and also very sensitive to the size of neighborhoods at different steps. Changes in the neighborhood size can cause serious differences in the results. To tackle these challenges and consider the curvature behavior, we propose a method using a normal vector of vertices to reduce the challenges and improve the results. The proposed method attempts to determine regions with lots of geometrical changes at a low cost and without complex calculations. These regions should be continuous as possible and the results must be approximately consistent, with minor parameter changes that can be used in different meshes. Vertex normal vectors can provide the necessary solution.

The normal vector is one of the vertex properties in the mesh. Its most important application is facing angle identification for lighting in computer graphics. As shown in Fig. 3, the normal vector has several directions in the regions with curvature. This property helps to use a normal vector instead of curvature to identify the change areas. This will tackle the challenge of the computational complexity of the curvature. However, the challenge of sensitivity to the neighborhood size will remain.

The curvature computes the amount of deflection that is strong or weak according to the speed of the surface change. It seems this causes curvature changes by changing the neighborhood because there are different changes in the mesh. However, we do not care about the intensity of changes for simplification. As Fig. 3 shows, the normal vectors have different directions where there are geometrical changes. As with other natural phenomena, these changes clearly have a distribution probability. We don't know this distribution, but a general distribution can be supposed. Most natural phenomena follow a normal or Gaussian distribution. Thus, we assumed that the distribution is Gaussian. Usually, there are different changes in the mesh surface, so the pattern of direction change is not the same everywhere. We are going to estimate the mean and variance of direction changes statistically. Therefore, we calculate the mean and variance for each vertex in a neighborhood, including some rings, with the following equations:

$$m(v) = \frac{\sum_{i=1}^{N} n_i}{N} \quad (2)$$

$$s(v) = \frac{\sum_{i=1}^{N} (n_i - m(v))^2}{N} \quad (3)$$

where m(v) is the mean of normal vectors in the neighborhood of v and s(v) is the normal vector variance around the mean vector. N is the number of vertices in the neighborhood, and n_i is the normal vector of the i-th vertex. It should be noted that the calculations are done separately in each dimension of 3D space. We aim to identify the direction of changes. As shown in Fig. 3, where the direction changes, there is diversity, and the intensity of the diversity represents the number of changes. Therefore, s(v) is directly proportional to the direction of changes. Each dimension of s(v) shows the diversity in that dimension on the mesh. So, we can use the magnitude of s(v) as the criterion to detect the number of geometrical changes. Using the normal vector and the probability distribution of the normal vector direction changes (because of less sensitivity to the surface changes) to find the regions with geometrical changes that lead to more continuous output without using the complex method as [16]. Fig. 4 shows the differences between the proposed model and the model of [16]. It should be noticed that normal vector information of vertices usually is stored with mesh files, so it is not required to calculate it. If there was no mesh data, calculating a normal vector is not somewhat difficult, and because of the elevation and depression, the finding procedure is based on the probability distribution. Changing the neighborhood size doesn't change the results until the pattern of the direction change does not vary.

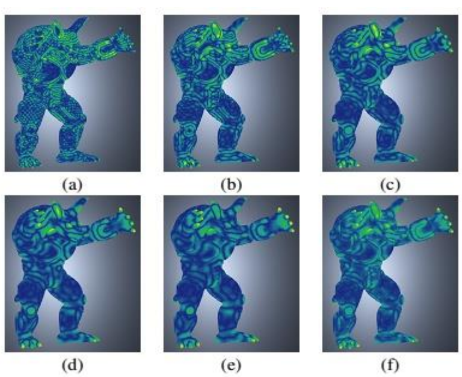

Fig. 3. Result of saliency detection [16] Results have been changed by scale changes.

IV. THE PROPOSED METHOD

The basic purpose of this work is mesh simplification. As mentioned before, we aim to decrease the changes in the important areas during mesh simplification. Since human eyes are more sensitive to areas with changes, we tend to change these areas less than the others. The way to identify changing areas has been explained in section II. . In the previous steps, the obtained weight for each vertex represents the geometrical changes around the vertex. Now, we are going to use this weight for simplification.

Furthermore, an error measurement or cost function in most

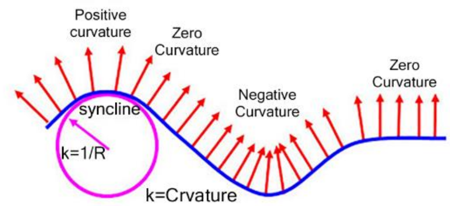

Fig. 2. The position of the normal vector at various locations of curve.

simplification algorithms controls the element selection for elimination and keeps the criterion small. Using the calculated weight in the error measurement of such algorithms allows more attention to the important areas. We combined our work with the method introduced in [4] First, we explain this method and combine our weight with it. The method in [4] is based on edge contraction. More concretely, in this method, vertex v is replaced by an edge (v1, v2) (Fig. 5). This algorithm follows the edges that lead to the minimum changes in the mesh. To this end, the cost is calculated for each contraction.

The v could be anywhere between v1 and v2. Each point has a different cost. The minimum cost for v is selected by [4] The cost is the summation of the squared distances between v and the planes that meet v1 and v2. Here, the plane means the planes created by using the normal vector of each triangle met by the vertex and the coordinate of the vertex. The cost is updated after each contraction. After sorting, an edge with a minimum cost is removed, causing a minimum change in the shape at each step. This action has high computational complexity due to the duplicate calculations of distances. To deal with this challenge, [4] proposes a method, including a mathematically proof, which is led to the squared distance from vertex v to a plane p, as shown in (4).

$$\mathrm{d(v)} = \mathrm{v}\, k_p v^T \quad (4)$$

where kp is a matrix based on the coefficients of the plane equation, p and v are the considered points, and T is the transpose sign. Initially, k_p is calculated for each triangle; then, the summation of k_p corresponds to meeting triangles for each vertex calculated (Q). To calculate the distance from v to meet planes with v1 and v2, the specified Q are used as follows:

$$\mathrm{C(v)} = v(Q_1 + Q_2)v^T \quad (5)$$

$$Q_i = \sum_{p\, \in planes\ of\ v_i} k_p$$

where C(v) is the cost of replacing vertex v to edge (v1, v2) and others, as in (4). In [4] for new vertex v, Q equals Q1+Q2, and recalculation is not required. Now, if we want to apply the considered weight to each vertex, we need to multiply the weight by (4), which needs the changing (4) and (5) to (6) and (7).

$$d(v) = w_v\left(v\, k_p v^T\right) = v\,(w_v\, k_p)v^T \quad (6)$$

$$C(v) = v(wQ_1 + wQ_2)v^T \quad (7)$$
$$wQ_i = w_v \sum\nolimits_{p\, \in planes\ of\ v_i} (k_p)$$

where w_v is our weight, other parameters are the same as (4) and (5). This way, we engaged the procedure's importance in high change areas. To create more differences between the areas, we also add a nonlinear transformation as follows:

$$w_{v=(\omega_v)^{\gamma}} \quad (8)$$

where ω_v is the calculated weight in the last step and γ is an exponent to create nonlinear differences between low and high change areas. Fig. 6 shows an example of output where QEM had done the simplification with and without the calculated weights.

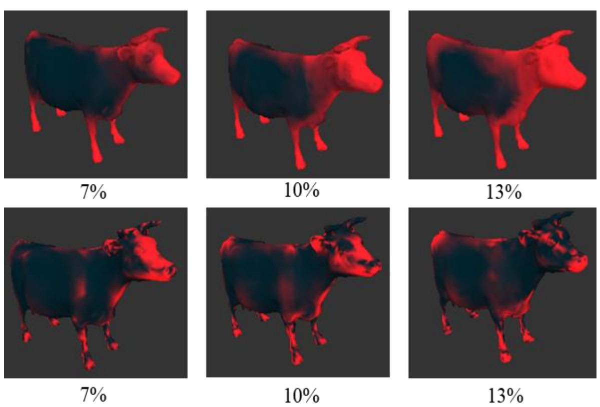

Fig. 4. Top row- Identification of normal diversity for different neighborhoods. Bottom row- mean curvature results for different neighborhoods. The size of the neighborhood is determined based on the bounding box diameter. More red means more geometric.

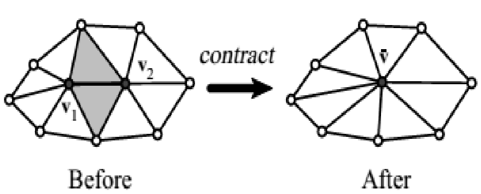

Fig. 5. Edge contraction.

V. EXPERIMENTAL RESULTS

In this section, several experiments are done, and the results are analyzed. First, an experiment is done to find the appropriate value of γ and neighborhood size (d). After determining the appropriate values of these parameters, our results and QEM results are compared. Finally, the outputs of some well-known models are shown and compared. We performed these experiments with C++ and used the official implementation of QEM with a dataset provided by Princeton University for segmentation evaluation, including 380 models in 19 different classes. The average of the mean curvature is used as the comparison criterion.

A. *Determining γ and d*

Both parameters, γ, and d, are directly related to the output; therefore, we changed each in an interval. By each change, the outputs for 100 identical models are obtained. The difference between the average mean curvature before and after simplification is considered the error of combination (γ, d). The neighborhood is considered based on a percentage of the diameter of the bounding box (Fig. 6). We change γ in [1 - 9] and d in [4-14] intervals. The best state is (5, 11), but it is not stable. Thus, to find an appropriate γ and d, each dimension's average is calculated separately according to the values of other dimensions (to understand error changes based on each parameter). The results are shown in Fig. 7. The minimum error occurs at 5 and 11 for γ and d, respectively, consistent with the previous results.

B. *Comparison with QEM*

The results of this work are compared with the results of the QEM. In other words, this is a comparison between the QEM with and without a normal diversity. The γ and d are set to the results of 4.1. For comparison, the level of simplification is changed from 10% to 80% by step 10%. Each method is tested with 100 identical models. The error criterion is the average of mean curvature. Fig. 7 shows the results. It is observed that our method has a lower error.

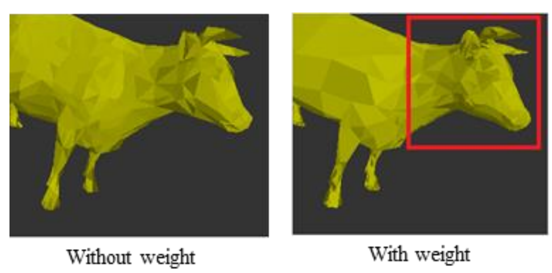

Fig. 6. Differences of output with and without normal diversity weight. Number of initial triangles: 5084 rate of simplification: 70%, retained triangle: 1074. The specific area of the right side has more density.

C. *Output*

In this section, we show our output for several models. The output of normal diversity is shown in Fig. 8, and the simplification output in Fig. 9 shows that the simplification level is not the same everywhere when using the weight of normal diversity. In the flat areas, there is more simplification level, and the number of triangles is less while their size is larger. In contrast, there are more triangles and smaller ones in the areas with more geometrical changes, causing a more accurate initial shape.

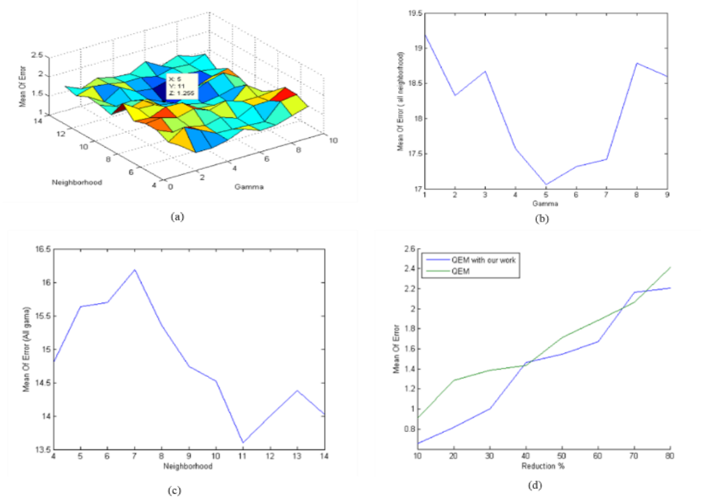

Fig. 7. (a) Error change according to d and γ, (b) The mean of error of all neighborhoods based on gamma, (c) The mean of error of all Gammas based on neighborhood size, (d) The error of the two methods comparison.

Fig. 8. Differences of output with and without normal diversity weight. Number of initial triangles: 5084 rate of simplification: 70%, retained triangle: 1074.The specific area of the right side has more density.

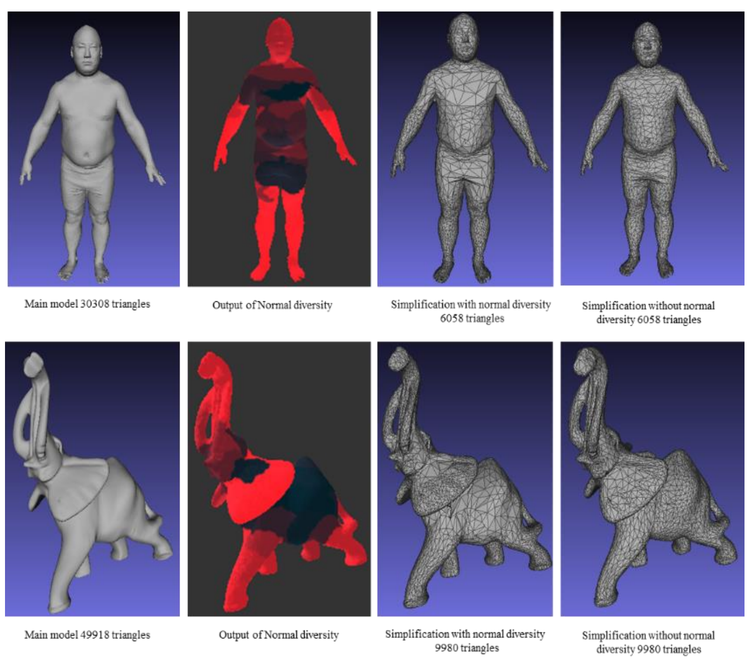

Fig. 9. Simplification output- normal diversity caused more simplification in the flat regions and less in the areas with geometrical changes. Simplification level is 80%.

VI. Conclusion

In this paper, we proposed a method to detect the elevations and depressions in the 3D mesh by using a normal vertex vector based on its probability distribution function, which does not have the problem of using curvature. We used the mean and variance of the normal distribution corresponding to the normal vector direction to detect the geometrical changes. Unlike the curvature, which usually suffers from high computational complexity and high sensitivity to the size of its neighborhoods, our method is faster and less sensitive, and its results are consistent with each other when neighborhood size is changed. We combined our results as a weight for each vertex with QEM. To this end, the calculated weight was multiplied by the initial errors of vertices. The simplification results show that the proposed method maintained the elevations, depressions (geometrical changes), and curve parts better than the general QEM. We used the average mean curvature as the error measure of elevations and depression changes. In future works, we would like to employ deep learning-based models in the field.

VII. References

[1] Yuk Ming Tang, H.L. Ho, 3D Modeling and Computer Graphics in Virtual Reality, Mixed Reality and Three-Dimensional Computer Graphics, 2020, 10.5772/intechopen.91443.

[2] Swantje Bargmann, Benjamin Klusemann, Jürgen Markmann, Jan Eike Schnabel, Konrad Schneider, Celal Soyarslan, Jana Wilmers, Generation of 3D representative volume elements for heterogeneous materials: A review, Progress in Materials Science, Volume 96, July 2018, Pages 322-384.

[3] Mauro Mazzei and Davide Quaroni, Development of a 3D WebGIS Application for the Visualization of Seismic Risk on Infrastructural Work, Int. J. Geo-Inf. 2022, 11(1), 22; https://doi.org/10.3390/ijgi11010022.

[4] Rumeng Lv, Xiaobing Chen, and Bingying Zhang, A simplified algorithm for 3D mesh model considering the influence of edge features, Journal of Physics: Conference Series, 2021.

[5] Guangyou Zhou; Shangda Yuan; Sumei Luo, Mesh Simplification Algorithm Based on the Quadratic Error Metric and Triangle Collapse, IEEE Access, Vol. 8, pp. 196341–196350, 2020.

[6] Schroeder, William J., Jonathan A. Zarge, and William E. Lorensen. "Decimation of triangle meshes." ACM Siggraph Computer Graphics. Vol. 26. No. 2. ACM, 1992.

[7] Renze, Kevin J., and James H. Oliver. "Generalized unstructured decimation [computer graphics]." Computer Graphics and Applications, IEEE 16.6 (1996): 24-32.

[8] Hoppe, Hugues. "Progressive meshes." Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM, 1996.

[9] Garland, Michael, and Paul S. Heckbert. "Surface simplification using quadric error metrics." Proceedings of the 24th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1997.

[10] Cao, Weiqun, Hujun Bao, and Qunsheng Peng. "An algorithm for LOD by merging near coplanar faces based on gauss sphere." Journal of Computer Science and Technology 16.5 (2001): 450-457.

[11] DeHaemer Jr, Michael J., and Michael J. Zyda. "Simplification of objects rendered by polygonal approximations." Computers & Graphics 15.2 (1991): 175-184.

[12] Hamann, Bernd. "A data reduction scheme for triangulated surfaces." Computer aided geometric design 11.2 (1994): 197-214.

[13] Schaefer, Scott, and Joe Warren. "Adaptive vertex clustering using octrees." Geometric Design and Computing 2.5 (2003).

[14] Bayik, Tolga, and Mehmet B. Akhan. "3d object database simplification using a vertex clustering algorithm." University of West Bohemia, Plzen, Czech Republic. 1999.

[15] Turk, Greg. "Re-tiling polygonal surfaces." ACM SIGGRAPH Computer Graphics 26.2 (1992): 55-64.

[16] Campomanes-Álvarez, B. Rosario, Sergio Damas, and Oscar Cordón. "Mesh simplification for 3D modeling using evolutionary multi-objective optimization." Evolutionary Computation (CEC), 2012 IEEE Congress on. IEEE, 2012.

[17] Huang, Hui-Ling, and Shinn-Ying Ho. "Mesh optimization for surface approximation using an efficient coarse-to-fine evolutionary algorithm." Pattern Recognition 36.5 (2003): 1065-1081.

[18] Álvarez, Rafael, et al. "A mesh optimization algorithm based on neural networks." Information Sciences 177.23 (2007): 5347-5364.

[19] Hoppe, Hugues. "New quadric metric for simplifiying meshes with appearance attributes." Proceedings of the conference on Visualization'99: celebrating ten years. IEEE Computer Society Press, 1999.

[20] Wei, Jin, and Yu Lou. "Feature preserving mesh simplification using feature sensitive metric." Journal of Computer Science and Technology 25.3 (2010): 595-605.

[21] Lee, Chang Ha, Amitabh Varshney, and David W. Jacobs. "Mesh saliency." ACM Transactions on Graphics (TOG). Vol. 24. No. 3. ACM, 2005.

[22] Wu, Jinliang, et al. "Mesh saliency with global rarity." Graphical Models 75.5 (2013): 255-264.

[23] Chen, Xiaobai, et al. "Schelling points on 3D surface meshes." ACM Transactions on Graphics (TOG) 31.4 (2012): 29

[24] Hongle Li and SeongKi Kim, A Novel Mesh Simplification Method Based on Vertex Removal Using Surface Angle, International Journal of Engineering Research and Technology. Volume 12, Number 8 (2019), pp. 1313-1320.

[25] Guangyou Zhou, Shangda Yuan, and Sumei Luo, Mesh Simplification Algorithm Based on the Quadratic Error Metric and Triangle Collapse, IEEE Access, 2020. 10.1109/ACCESS.2020.3034075.

[26] Rolandos Alexandros Potamias, Stylianos Ploumpis, Stefanos Zafeiriou, Neural Mesh Simplification, CVPR, pp. 18583- 18592, 2022.

[27] Marie-Julie Rakotosaona, Paul Guerrero, Noam Aigerman, Niloy J Mitra, and Maks Ovsjanikov. Learning delaunay surface elements for mesh reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22–31, 2021.

[28] Thibault Lescoat, Hsueh-Ti Derek Liu, Jean-Marc Thiery, Alec Jacobson, Tamy Boubekeur, and Maks Ovsjanikov. Spectral mesh simplification. Computer Graphics Forum, 39(2):315–324, 2020.