The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-3/W2-2022
Urban Geoinformatics 2022, 1–4 November 2022, Beijing, China

A NOVEL QUADRATIC ERROR METRIC MESH SIMPLIFICATION ALGORITHM FOR 3D BUILDING MODELS BASED ON 'LOCAL-VERTEX' TEXTURE FEATURES

Tianming Zhao[1,2], Jie Jiang[1,2], Xian Guo[1,2]*

1 School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture(BUCEA), Beijing, 100044, China; 2108570020097@stu.bucea.edu.cn(Z.TM.), jiangjie@bucea.edu.cn(J.J.), guoxian@bucea.edu.cn (G.X.).
2 Key Laboratory of Urban Spatial Information, Ministry of Natural Resources of the People's Republic of China, Beijing University of Civil Engineering and Architecture, Beijing 102616, China
* Correspondence: guoxian@bucea.edu.cn

KEY WORDS: Texture features, Mesh simplification, 3D building model, Vertex curvature, Quadric Error Metric, Edge collapse.

ABSTRACT:

3D building model is an important part of 3D GIS. However, with the rapid development of data acquisition technology, the data volume of the 3D building model has increased dramatically. Levels of detail (LOD) technology determines the resource allocation for object rendering based on the position (Screen Size) and importance of the nodes of the model in the display environment, reducing the model's volume and thus obtaining efficient rendering computations. To ensure the smooth rendering and efficient loading of 3D building models, it is essential to simplify the 3D building models to generate LOD. By taking the texture discontinuity and topological complexity into account, this paper proposes a quadratic error metric mesh simplification algorithm based on "local-vertex" texture features for 3D building models simplification. Using the texture features of the model texture map and the vertex curvature at local vertices, we increase the edge collapse cost of the model in the rich texture areas. After each simplification operation, we use centre of gravity coordinate method to optimize the texture coordinates to preserve the model's detailed features and topological relationships. A series of experimental comparisons with other state-of-the-art methods verify the effectiveness of the proposed method for simplifying 3D building models. The method in this paper helps to maintain the detailed features and topological relationships of 3D building models while reducing the volume of model and better generating LOD for application in 3D GIS.

1. INTRODUCTION

3D building models are an essential part of the virtual geographic environment. In recent years, 3D building models have been widely used in digital cities, 3D navigation, and game production, etc. However, with the rapid development in data acquisition technology, the volume of 3D building models has increased dramatically. Although the level of computer hardware has been greatly improved recently, the level of computer hardware at present still cannot meet the requirements of large-scale 3D scene implementations, resulting in poor user experience. The main solution to this problem is to use the levels of detail (LOD) technique during the model rendering process, thus improving the rendering efficiency of the whole scene.

The simplification operation of a model is the key to generating a suitable LOD. Model simplification using specific methods to turn the original model into a rough approximation, reducing the volume t of the model while preserving the essential geometric and visual features of the model as much as possible. For mesh models, researchers have proposed many model simplification methods, such as adaptive subdivision method (DeHaemer Jr, 1991), sampling method (Turk, 1992), vertex clustering method (Rossignac & Borrel, 1993) and geometric element deletion method. The adaptive subdivision method is suitable for multi-resolution surface editing, but the simplification of the model is usually limited in order to guarantee the topology. The sampling method has a complex simplification mechanism, and its processed mesh results tend to be evenly distributed, which is suitable for models with smooth surfaces. The vertex clustering method is relatively efficient, but it cannot retain the topology of the model well in the model simplification. The geometric element deletion method calculates an error metric value for each geometric element on the mesh model and sorts the values into a data structure so that the element with the smallest removed error metric value is deleted at each iteration of the traversal. After that, the other elements affected by the deleted element are updated and the computation continues iteratively until the requirements are met. This kind of methods mainly includes vertex deletion (Schroeder, Zarge, & Lorensen, 1992), edge collapse (Hoppe, DeRose, Duchamp, McDonald, & Stuetzle, 1993), and face collapse. However, there are limitations to these methods. The vertex deletion method cannot simplify non-streaming vertices and can only be applied to stream vertices. The face collapse method has relatively less memory occupation, but the results are relatively coarse. This paper is mainly inspired by the edge collapse, whose time complexity and space complexity are relatively low, and the simplification performance are better.

The selection of the error metric in the edge collapse method directly affects the performance of the model simplification, so it is crucial to determine a reasonable error metric. Garland proposed a quadratic error metric (QEM) (Garland & Heckbert, 1997), which is a representative technique for limiting the local curvature and volume changes of the model in simplification. It uses the distance from the new vertex to the first-order neighbourhood face as the error metric, resulting in simple operation and high computational efficiency. However, the QEM algorithm cannot fully retain the detailed features of the model and does not take the appearance attributes such as texture and colour into account(Garland & Heckbert, 1998). Some experts and scholars have optimized the algorithm based on the QEM algorithm. Garland extended the QEM algorithm by adding matrix dimensions, which added more appearance properties but could not deal effectively with locally flat regions(Garland & Heckbert, 1998). C. Michaud(Michaud, Mellado, & Paulin, 2017) extended the QEM algorithm evaluation to quadratic surfaces represented by mesh geometry and fitted planes. Nathaniel (Williams, Luebke, Cohen, Kelley, & Schubert, 2003) takes texture bias and dynamic illumination effects into account and prioritizes edge collapse based on a perceptual model. Carlos

(González et al., 2013) defined the error metric of edge collapse as the final change in the structural appearance of the texture model. Jun Huang (Huang, Wang, & Wang, 2020) introduced the concept of edge curvature approximation measure to include edge curvature in the error metric.

The above mentioned simplification methods can effectively reduce the complexity of the model, but the 3D building model has a complex topology(Q. Li, Sun, Yang, & Jiang, 2013), and the texture coordinates corresponding to the model vertices are usually not unique, leading to discontinuities in the model texture. Simply applying general simplification algorithms to simplified 3D building models may violate certain geometric and topological constraints, which may lead to the occurrence of texture distortions. For the above characteristics of 3D building models, researchers related to 3D building model simplification have been conducted in literature. Wang(Y. Wang, Zhang, Mathiopoulos, & Deng, 2015) proposed a simplified method based on graph cut to optimize the process of building clustering Biao (B. Wang et al., 2021) used the topological dependence among 3D building model components as a constraint for model simplification, but did not consider the texture information of the model. Li (M. Li & Nan, 2021) designed a mesh filtering and simplification method for 3D building mesh models. Chen(Chen, Li, & Li, 2015) proposed a vertex clustering algorithm with texture correlation error degree, which can preserve the texture details of the model well, but it is difficult to maintain the local connectivity of the geometry. However, the current 3D building model simplification algorithm cannot be well applied to complex 3D building models, so this paper proposes a new simplification method for complex 3D building models.

In order to maximum retention appearance of the original 3D building model in the simplification of the 3D building model, this study proposes a quadratic error metric mesh simplification algorithm based on the "local-vertex" texture feature based on the QEM algorithm. Firstly, the Grey-Level Co-occurrence Matrix (GLCM) is used to pre-process the texture map of the model to get the corresponding texture feature map, and then the mesh information is inverse mapped to the texture feature map to get the related texture information. After that, the normal vector of the face is calculated according to the vertex coordinates of the triangular face piece, and the vertex normal vector is obtained by weighting the face normal vector with the area of the face where it is located, and then the vertex curvature can be solved. The texture information is weighted with the vertex curvature to optimize the vertex matrix in the QEM algorithm to increase the collapse cost of the rich texture region of the model and to preserve the detailed features of the model. This study uses a new error metric that takes geometric and texture errors into account in the simplification process and is able to better preserve feature details of the model after simplification compared to traditional methods.

This paper is organized as follows: Section 2 provides a detailed description of the algorithm proposed in this paper, Section 3 analyses the experimental results obtained by the algorithm and summarizes the shortcomings, and Section 4 concludes and outlooks the study in this paper.

2. METHOD

The algorithm flow is shown in Figure 1. Firstly, as the pre-processing part of the data the texture map of the 3D building model is subjected to texture feature calculation, and the mesh information is inverse mapped to the texture feature image to obtain the corresponding texture information values. After that, the curvature of the vertices is calculated, and the texture information values and the vertex curvature are used to optimize the error matrix of the model vertices to calculate the new collapse cost. The edge with the smallest collapse cost is selected for the edge collapse operation, the information of the elements that receive the influence of the edge is updated. Determine whether the simplification rate reaches the threshold, if not then iterate through the edge collapse operation, otherwise end the algorithm.

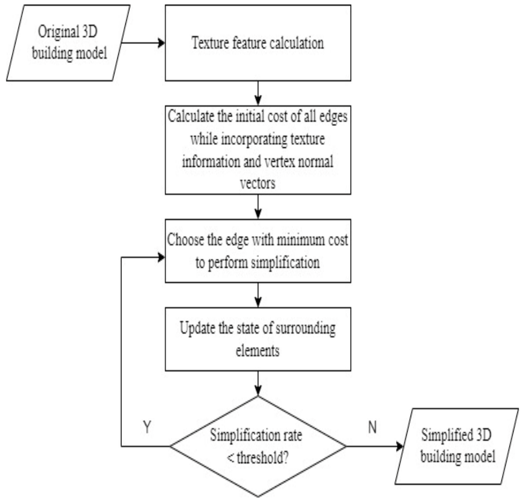


Figure 1. Flowchart of the method

2.1 Description of the QEM algorithm

The following is a detailed description of the QEM algorithm. Firstly, define a 4×4 initial quadratic error matrix Q for all vertices $\bar{v} = [x\ y\ z\ 1]^T$ in the mesh model where the error of vertex $\bar{v}$ is a quadratic term of the form $\Delta(\bar{v}) = \bar{v}^T Q \bar{v}$. For the vertices v_j、v_i、v_k of a face in the model, assume that the edge formed by any two of them is (v_i, v_j). The new vertex v_{new} is produced by collapse the edge (v_i, v_j) and define the quadratic error matrix $Q_{new} = Q_i + Q_j$ for the new vertex v_{new}.

The error matrix Q is related to the sum of squares of the first-order neighborhood face distances from the new vertex to the original vertex which resulting from the edge collapse. Assuming that the plane equation is $ax + by + cz + d = 0$, where $a^2 + b^2 + c^2 = 0$, and d is a constant. The square of the distance $d^2(v_{new})$ is calculated according to the formula of the distance from the point to the plane as:

$$d^2(v_{new}) = {v_{new}}^T (pp^T) v_{new} \tag{1}$$

where $p = [a\ b\ c\ d]^T$
v_{new} is the position of the new vertex

Denoting pp^T as K_p, it can be represented as a 4×4 symmetric matrix which can be expressed as follows:

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

Then the quadratic error matrix $Q(v)$of the vertex v can be expressed as:

$$Q(v) = \sum_{p \in \mathrm{plane}(v)} K_p \tag{3}$$

https://doi.org/10.5194/isprs-archives-XLVIII-3-W2-2022-109-2022 | © Author(s) 2022. CC BY 4.0 License.

where $plane\ (v)$ = the set of all triangular faces containing vertex v.

In summary, for a collapsed edge (v_j, v_i), the collapse cost of this edge into a new vertex v_{new} is denoted by $cost(v_i, v_j)$:

$$\begin{aligned} cost(v_i, v_j) &= {v_{new}}^T Q_{new} v_{new} \\ &= {v_{new}}^T (Q_i + Q_j) v_{\text{new}} \\ &= {v_{new}}^T \left(\sum_{p \in \text{ lane } (v_i)} K_p + \sum_{p \in \text{ plane } (v_j)} K_p \right) v_{new} \end{aligned} \quad (4)$$

$cost(v_i, v_j)$ can also be expanded as follows:

$$\begin{aligned} cost(v_i, v_j) &= {v_{new}}^T Q_{\text{new}}\, v_{\text{new}} \\ &= q_{11}x^2 + 2q_{12}xy + 2q_{13}xz + 2q_{14}x + q_{22}y^2 \\ &\quad + 2q_{23}yz + 2q_{24}y + 33_{33}z^2 + 2q_{34}z + q_{44} \end{aligned} \quad (5)$$

where $v_{\text{new}} = [x\ y\ z\ 1]^T$

The collapse cost depends on the position of the new vertex v_{new}, and there exists an optimal position makes $cost(v_i, v_j)$ gets a local minimum. Taking the partial derivatives of x, y and z in equation (5) and making them zero is equivalent to solving for:

$$v_{new} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (6)$$

If the coefficient matrix is invertible, the position of the new vertex v_{new} can be found by equation (6). Otherwise, by comparing the collapse cost at the midpoint and two endpoints of the edge (v_i, v_j), the point with the smallest collapse cost is selected as the position of the new vertex v_{new}.

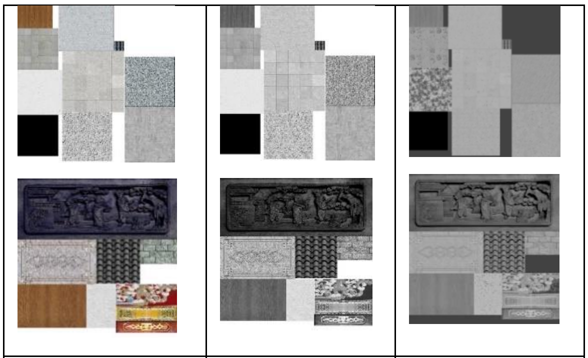
2.2 Image texture feature extraction

Some model vertices in 3D building models may have more than one texture coordinate, which results in a discontinuity in the model texture. The texture image of the model provides rich visual detail information to the model, and the information contained in different areas on the texture image may vary greatly, with complex variations in some parts and simple colours in others. If the texture information is not considered in the simplification process, it will lead to texture distortion.

We first extract texture features from the original texture image to obtain the texture image corresponding to the texture feature map. Then inverse map the 3D mesh to the texture feature map to obtain the relevant texture feature information values $vertex_{texture(i)}$. Thus, the relevant texture factors are introduced into the error metric to achieve the purpose of retaining rich texture regions in the model simplification and producing less texture bias in local areas with more texture information.

In this paper, the process of texture feature extraction is used as the pre-processing stage of the method, and the Grey-Level Co-occurrence Matrix (GLCM) (Mohanaiah, Sathyanarayana, & GuruKumar, 2013)is used to extract texture features.

The quadratic statistical features of the image's GLCM can accurately and qualitatively reflect the texture characteristics of the image. As shown in Figure 2, that is, the texture feature images of the experimental data model 1 and model 2 extracted by the GLCM algorithm. Part (a) is the original texture image corresponding to the model, part (b) is the grayscale map of the original texture image, and part (c) is the texture feature map with Median parameters corresponding to the original texture image.



(a)	(b)	(c)

Figure 2. Calculation of texture features: (a) original texture map (b) grayscale image (c) texture feature Map

The 3D mesh is inverse mapped to the computed texture feature images to obtain the texture information values at the corresponding pixel in each image. In this paper, we calculate the arithmetic mean of the texture information values of multiple texture feature images to obtain the texture feature information values $vertex_{texture}$ at each pixel:

$$vertex_{texture} = \frac{\sum_{i=1}^{k} vertex_{texture(i)}}{n} \quad (7)$$

where $vertex_{texture(i)}$ is the value of the associated texture feature information at each pixel
n is the number of texture feature images used

2.3 Vertex curvature control

The triangular faces that make up a 3D building model are not uniform, and there may be a large number of narrow triangular faces in some areas.

If the edge collapse operation is performed where the model has a large number of narrow triangular faces, it may have an huge impact on the model's appearance. For better edge collapse operation, vertex curvature is included as an error measure in this paper. Because triangular mesh models are segmented linear approximations to smooth or segmented smooth surfaces where vertex curvature refers to the approximate curvature of the vertices in general, the vertex curvature is smaller in flatter regions of the 3D model and more significant in areas with larger model transitions. In the error metric of edge collapse, reducing the error weight of edges with more significant vertex curvature is beneficial to preserve the topology and local detail features of the model. The following is a detailed definition of vertex curvature.

The normal vector n_i of a triangular face is calculated from the vertex coordinates of the face, and n_i can be expressed as:

$$n_i = \frac{(v_j - v_i) \times (v_k - v_j)}{\| (v_j - v_i) \times (v_k - v_j) \|} \quad (8)$$

where v_j、v_i、v_k is the three points that make up the face

The face normal vectors of the first-order neighbourhood of v_i and the area of the face S_i are weighted to yield the vertex normal vector n_{V_i}:

$$n_{v_i} = \frac{\sum_{i=1}^{k} S_i n_i}{\| \sum_{i=1}^{k} S_i n_i \|} \tag{9}$$

After getting the normal vertex vector of vertex v_i, the curvature c_{v_i} of the vertex v_i can be calculated as follows:

$$c_{v_i} = \frac{\sum_k \alpha(n_{v_i}, n_i)}{k} \tag{10}$$

where $\alpha(n_{v_i}, n_i)$ is the angle between the vertex normal vector and the k related triangle face pieces in the neighbourhood.

2.4 Optimizing collapse cost

The vertex texture information $vertex_{texture}$ and vertex curvature c_{v_i} are weighted to obtain the new error matrix Q_{weight} and collapse cost $cost(v_1, v_2)$ of the vertices.

$$Q_{weight} = \left(1 + vertex_{texture(i)} \times (factor - 1)\right) \times c_{V_i} \times Q \tag{11}$$

where $factor$ is the user-defined weight factor
$vertex_{texture(i)}$ is the value of the associated texture feature information at each pixel
c_{V_i} is the vertex curvature

$$cost(v_i, v_j) = \bar{v}^T (Q_{i\,weight} + Q_{jweight}) \bar{v} \tag{12}$$

where $\bar{v}$ is the vertex coordinate matrix
$Q_{iweight}$, $Q_{jweight}$ are the weighted Q_{weight} matrices of vertices
v_i, v_j are the vertex coordinates

In this paper, the algorithm is optimized by vertex texture information and vertex curvature, to achieve the increased of collapse cost in rich texture regions and achieve the purpose of retaining the detailed features of the model in simplification.

2.5 Texture coordinates adjustment

Since the texture of a model presents a different amount of colour variation in different parts of the model. So, the colour variation can be used to simplify the more uniformly coloured parts of the model to a greater extent. Each vertex may be associated with one or more texture maps, resulting in vertices that may have multiple texture coordinates. To deal with this situation in the model simplification, we make a unique treatment of the data structure of triangles and vertices by storing texture coordinates in triangles, so that vertices are associated with triangles for the purpose to maintain the corresponding texture coordinate relationships between vertices and the associated triangular faces.

After each edge collapse operation, not only the connection relations of the corresponding vertices are updated, but also the texture coordinates of the vertices and the associated faces. In this paper, we use the centre of gravity coordinates of the triangle face and update the texture coordinates of each vertex of the triangle affected by the edge collapse operation by interpolation operation, thus reducing the texture distortion of the model due to the simplification operation.

3. METHOD RESULTS AND DISCUSSIONS

In order to verify the effectiveness of the proposed algorithm, experiments were conducted with 3D building model in triangular mesh format. The hardware configuration of the computer used is Intel(R) Xeon(R) Gold 5118 CPU @ 2.30 GHz 2.29 GHz, 64 GB RAM, and the development language used is Microsoft Visual C++. It should be noted that if the data is in other formats, it is suggested to be converted to triangular mesh format.

As shown in Figure 3, the experimental data in this paper are two 3D building models, model 1 with 374904 triangular faces and 196507 vertices, and model 2 with 625520 triangular faces and 312388 vertices. Model 1 and model 2 are used as the original models for mesh simplification, and the models are simplified using different methods and simplification rates.

In this paper, we use Hausdorff distance (Boulch, de La Gorce, & Marlet, 2014) as the accuracy evaluation index to objectively evaluate the quality of mesh simplification. Given two point-set $A = \{a_1, a_2, a_3 \ldots\}$ and $B = \{b_1, b_2, b_3 \ldots\}$ in Euclidean space, the Hausdorff distance is used to measure the distance between these two point-set. The defining formula is as follows:

$$\mathrm{H(A,B)} = max[\mathrm{h(A,B)}, \mathrm{h(B,A)}]$$

where $h(A,B) = \max_{a \in A} \min_{b \in B} \| a - b \|$
$h(B,A) = \max_{b \in B} \min_{a \in A} \| b - a \|$

(a) Experimental data1

(b) Experimental data2
Figure 3. Original 3D building model

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-3-W2-2022-109-2022 |

Model	Simplification rate	Number of Triangle	Number of Vertices
1	25%	281111	147889
1	50%	187385	98339
1	75%	93735	51144
2	25%	467524	230723
2	50%	311194	153150
2	75%	155223	77804

Table 1. Statistics of model simplification results.

Table 1 shows the details of the model 1 and model 2 simplifications, which indicate the number of triangular faces and vertices of the model at different simplification rates. Figure 4 shows the thumbnail and local enlargement of model 1 obtained by using the QEM algorithm and the algorithm in this paper at different simplification rates. Figure 5 shows the thumbnail and local enlargement of model 2 obtained by using the QEM algorithm and the algorithm in this paper at different simplification rates.

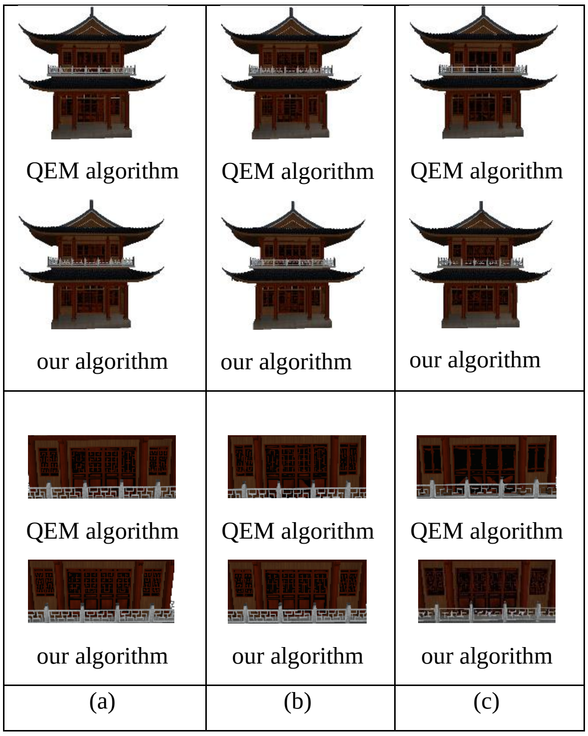
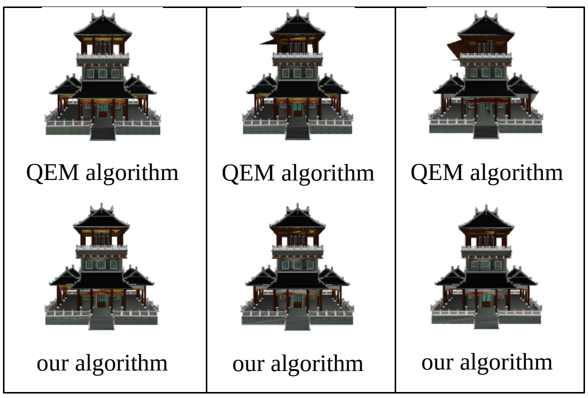


Figure 4. Simplified results using the QEM algorithm and our algorithm: (a) The simplification rate is 25% (b) The simplification rate is 50% (c) The simplification rate is 75%



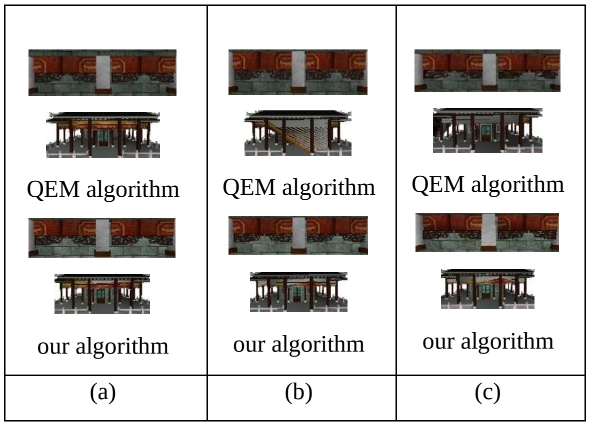


Figure 5. Simplified results using QEM and our algorithm: (a) The simplification rate is 25% (b) The simplification rate is 50% (c) The simplification rate is 75%

Visual interpretation to Fig.4 and Fig.5 shows that better simplification results are obtained at lower simplification rates using both the QEM algorithm and this paper's algorithm. But as the simplification rate increases from 25% to 50% and 75%, better results are obtained using this paper's algorithm.

In the zooming views of Fig. 4 and Fig. 5, it is obvious that the detailed features of the rich texture regions of the experimental data used in this algorithm are better preserved compared to the QEM algorithm. Fewer regions on the model undergo texture distortion, and the regions with less texture information and flatter areas are simplified to a greater extent. And the algorithm in this paper has a good effect on maintaining the topology of the model which the model is not prone to large deformations in the simplification.

A comparison of the results using Hausdorff distance for experimental data1 is shown in Figure 6. When the simplification rate is 10% and 20%, the difference between this algorithm and the QEM algorithm is about 10% of the error value of the QEM algorithm; when the simplification rate is 40% and 70%, the difference between this algorithm and the QEM algorithm is about 35% of the error value of the QEM algorithm; when it is other integer simplification rate, the difference between this algorithm and QEM algorithm is about 25% of the error value of the QEM algorithm. In terms of the simplification error, this algorithm is significantly better than the QEM algorithm, which means that the model deformation can be well constrained in the simplification and more detailed features of the model are retained.

Figure 7 shows the time consumption of experimental data 1 using the QEM algorithm and this paper's algorithm, the time consumption of this paper's algorithm is about 1.3 times of the time consumption of the QEM algorithm. The algorithm in this paper has more computational processes so the algorithm takes longer, but the percentage of time increase is not high, and the increase in time cost is acceptable considering the merits of the simplified results.

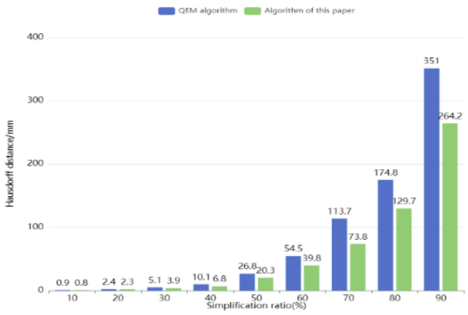


Figure 6. Comparison of simplification errors of different methods

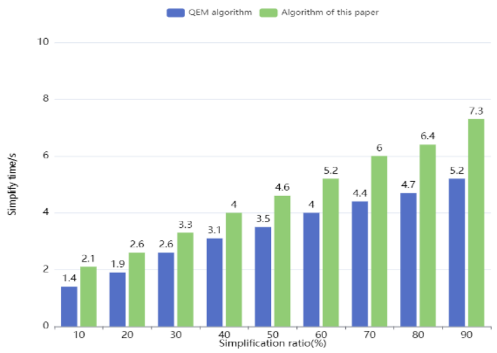


Figure 7. Comparison of simplification time of different methods

The above experimental results show that the quadratic error metric mesh simplification algorithm based on "local-vertex" texture feature improvement proposed in this paper can well preserve the detailed features of the rich texture areas in the model. And this is an effective way to simplify the 3D building model, which can well preserve the topological relationship of the 3D building model.

4. CONCLUSIONS AND FUTURE WORK

This paper proposes a quadratic error metric mesh simplification algorithm based on "local-vertex" texture features based on the QEM algorithm, which considers the image texture features and vertex curvature in the mesh simplification to change the error matrix of the mesh vertices and optimize the collapse cost.
Experiments on experimental data demonstrates that the proposed algorithm is an effective simplification algorithm for 3D building models. The experiments illustrate that the algorithm in this paper is an effective simplification algorithm for 3D building models, with an error accuracy improvement of about 25%. The algorithm enables the edge collapse to occur as much as possible in areas with less texture information and flatter areas on the model, making it possible to retain the maximum number of detailed features of the model in the simplification and generating a 3D building model with higher quality.

Our future work will focus on the following aspects. The first is the need to automatically calculate the appropriate simplification rate according to the viewpoint position, so as to automatically generate a suitable discrete LOD model for 3D representation; the second is to add a special treatment for the model boundary gap in the method, and to consider the compression and simplification of the texture image in the model simplification; the last is to use a method of image quality evaluation in simplification to evaluate the model simplification quality.

ACKNOWLEDGEMENTS

The research is supported by the National Key Research and Development Program of China (Grant 2021YFE0117500), the R&D Program of Beijing Municipal Education Commission (KM202210016004), and the Pyramid Talent Training Project of Beijing University of Civil Engineering and Architecture (JDYC20200322).

REFERENCES

Boulch, A., de La Gorce, M., & Marlet, R., 2014: Piecewise-planar 3D reconstruction with edge and corner regularization. *Paper presented at the Computer Graphics Forum.*

Chen, J., Li, M., & Li, J., 2015: An Improved Texture-related Vertex Clustering Algorithm for Model Simplification. *Computers Geosciences, 83*, 37-45.

DeHaemer Jr, M. J., Zyda, Michael J., 1991: Simplification of Objects Rendered by Polygonal Approximations. *Computers Graphics, 15*(2), 175-184.

Garland, M., & Heckbert, P. S, 1997: Surface Simplification using Quadric Error Metrics. *Paper presented at the Proceedings of the 24th annual conference on Computer graphics and interactive techniques.*

Garland, M., & Heckbert, P. S.,1998: Simplifying Surfaces with Color and Texture Using Quadric Error Metrics. *Paper presented at the Proceedings Visualization'98* (Cat. No. 98CB36276).

González, C., Castelló, P., Chover, M., Sbert, M., Feixas, M., & Gumbau, J., 2013: Simplification Method for Textured Polygonal Meshes based on Structural Appearance. *Signal, Image Video Processing, 7*(3), 479-492.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., & Stuetzle, W., 1993: Mesh Optimization. *Paper presented at the Proceedings of the 20th annual conference on Computer graphics and interactive techniques.*

Huang, J., Wang, X., & Wang, J., 2020: Mesh Simplification Algorithm based on Edge Curvature Metrics and Local Optimization. *International Journal of Modeling, Simulation,Scientific Computing, 11*(01), 1950042.

Li, M., & Nan, L., 2021: Feature-preserving 3D mesh simplification for urban buildings. *ISPRS Journal of Photogrammetry Remote Sensing, 173*, 135-150.

Li, Q., Sun, X., Yang, B., & Jiang, S., 2013: Geometric structure simplification of 3D building models. *ISPRS Journal of Photogrammetry Remote Sensing, 84*, 100-113.

Michaud, C., Mellado, N., & Paulin, M., 2017: *Mesh Simplification with Curvature Error Metric. Paper presented at the Eurographics 2017*.

Mohanaiah, P., Sathyanarayana, P., & GuruKumar, L., 2013: Image texture feature extraction using GLCM approach. *International journal of scientific research publications, 3*(5), 1-5.

Rossignac, J., & Borrel, P., 1993: Multi-resolution 3D approximations for rendering complex scenes, Berlin, Heidelberg.

Schroeder, W. J., Zarge, J. A., & Lorensen, W. E., 1992: Decimation of Triangle Meshes. *Paper presented at the Proceedings of the 19th annual conference on Computer graphics and interactive techniques*.

Turk, G., 1992: Re-tiling polygonal surfaces. *Paper presented at the Proceedings of the 19th annual conference on Computer graphics and interactive techniques.*

Wang, B., Wu, G., Zhao, Q., Li, Y., Gao, Y., & She, J., 2021: A Topology-Preserving Simplification Method for 3D Building Models. *Isprs International Journal of Geo-Information, 10*(6), 422.

Wang, Y., Zhang, L., Mathiopoulos, P. T., & Deng, H., 2015: A Gestalt rules and graph-cut-based simplification framework for urban building models. *International Journal of Applied Earth Observation Geoinformation, 35*, 247-258.

Williams, N., Luebke, D., Cohen, J. D., Kelley, M., & Schubert, B., 2003: Perceptually guided simplification of lit, textured meshes. *Paper presented at the Proceedings of the 2003 symposium on Interactive 3D graphics.*