3D Models Simplification Algorithm for Mobile Devices

Xun Wang
College of Computer Science and Information Engineering, Zhejiang Gongshang University
Hangzhou 310018, China
Email: wx@mail.zjgsu.edu.cn

Bai-Lin Yang
College of Computer Science and Information Engineering, Zhejiang Gongshang University
Hangzhou 310018, China
Email: ybl@mail.zjgsu.edu.cn

***Abstract*—As we know, the mobile device screen is small. The lower accuracy of the model is relatively weak and the capacity to handle high detailed model is very limited. What's more, the existing three-dimensional simplification algorithms are for the personal computer and they are not suitable for the mobile terminals. Thus, we propose a novel 3D model simplification algorithm based on feature points. We will find the model surface curvature of each vertex, based on the model feature points. For these points, we present a new method to obtain it. Experimental results show that the algorithm can accurately reflect the model changes of the local surface geometry, and effectively keep the details of the model characteristics.**

***Index Terms*—simplification, edge-collapse, similar curvature, shape feature**

I. Introduction

With the increasingly widespread application of computer graphics, three-dimensional geometric data size and complexity grows rapidly. In recent years, although the mobile communication technology develops rapidly, PDA (personal digital assistant), cell phone and other mobile computing devices' processing power and capacity of storage is relatively low, followed by a series of mobile computing which attract the attention of many researchers and scholars, while the mobile terminal on the calculation of three-dimensional graphics is one of challenging issues.

As the three-dimensional geometric data is big and relative low. Researcher solve this problem by raising the processor's processing speed, increasing memory capacity and network bandwidth and other hardware measures, but they are not enough. Therefore, a three-dimensional geometry data compression in computer graphics and networking technology becomes an important topic. Currently, the geometry compression has already had a series of classic and simplified algorithm for the network progressive transmission of multi-resolution method. Item [1] proposed a mesh simplification algorithm to remove vertices. According to certain rules, the algorithm each time selects a vertex from the model, and if it is found, then we delete the vertex and associated edges and triangles. Next, we delete it in the net lattice model, re-formed whole triangulation. The iterative process continues until no vertex can be removed so far. This method has the advantage that the simplified model is the culmination of a subset of the original model, and the model can be directly used for the original vector and texture information; however, its disadvantage is that the simplified model in the visual effects is poor. Item [2] proposed a vertex aggregation of mesh simplification algorithm, which first determines the importance of vertices that are connected with a large surface, or more importantly, a higher number of the vertex curvature, called feature points, then three-dimensional space unit is divided. In the same unit, there are vertices that could replace those features; the accuracy of cell division determines the final outcome of visual effects. Although we can simply and rapidly implement these algorithms, we have to change the topology of the model and the model cannot control the level of detail. These cannot determine how many triangles can be generated. Item [3] proposed a mesh simplification algorithm for edge contractions. Edge contraction is determined in the order of importance based on the edge and is continuously removed from the model side, removing one side of each which is reduced by one vertex and two triangles. Garland [4], who extended the contraction of the edge method. However, the distance is less than the non-adjacent vertices of a characteristic value which can also be contracted, so that it can be connected to the area which does not naturally be linked. Garland's method also changed the model topology. These can be divided into two categories: single-resolution and multi-resolution compression. For single resolution, a 3D mesh is compressed into one single representation with high compression ratio. A complete survey is given in [5]. Triangle strips representation supported by OpenGL and other graphics libraries reduce the number of times in the same vertex, which is accessed in graphics adapters. Deering [6] generalized it to the reference of any

doi:10.4304/jsw.7.11.2599-2605

previously decoded vertex by storing them into a register cache. Taubin introduced Topological Surgery [7]. TS is the connectivity preserving single-resolution technique optimized for network transmission and it is accepted as a standard format for VRML [8]. However, single resolution techniques are not suitable for network transmission because the user has to wait for the entire compressed data to be downloaded. Multi-resolution compression is to compress the original meshes into small base mesh and a sequence of renewment. For example, bone (16 MB) can be compressed into 8 batches. The base mesh is 78KB and can be transmitted over 28.8 Kbps modem in a few seconds. Hoppe's PM [9] is based on the Taubin technique. After a sequence of edge collapsing operations, the original mesh is reduced to the base mesh. Each edge collapsing removes one vertex at a time from the mesh. The connectivity and geometric information of the removed vertex are stored in the renewment. The inverse process is called---vertex split. A new vertex is inserted into the current mesh with the data stored in the corresponding renewment. PM provides a control of the accuracy and supports view-dependent renewment [10]. However, PM is not efficient in the sense of compression ratio. It requires about 8 bits (improved to 5 to 7 by Hoppe [11]) per vertex to encode the connectivity and 15 to 25 for the geometry [12].

Gaubin's PFS [13] is much more efficient in encoding at the expense of looser granularity. It groups renewments into a batch to achieve high compression ratio.MPEG-4 [14] accepts PFS as the standard compression scheme. However, PFS is not widely implemented in current 3D players [15]. CPM [16] groups vertex splits into batches. Each batch contains about 30 percent of the previous decoded mesh. In each batch, the connectivity information of the removed vertex is encoded with the identifier of the two cut-edges. CPM applies Buffer fly subdivision scheme to predict the displacement of the new vertex. The error between the predicted and the original position is stored as the geometric data. The amortized connectivity encoding takes 3.6 bits for geometry per vertex. Most of the initial mesh compression techniques use triangle strips as their encoding strategy [21, 22, 23], and/or vertex buffers [24]. The Edge Breaker algorithm [36], however, uses a different strategy: it turns a mesh into a sequence of five-symbol strings using an edge conquest. Using this method, a guaranteed 3.67 bit/vertex rate for connectivity is presented in [26]. A very efficient decompression of an Edge Breaker code is introduced in [27], while a better rate for regular models is proposed in [25, 28].

Vertices of the above methods are dependent on the importance of the selection that first is removed or aggregated which is part of the grid. [29].The various costs of the simplified method were compared with different conclusion; [30] which used in the secondary error scale (quadric error metrics, QEM) simplification can lead a very good result. But QEM is based on the size of the secondary weights to keep the details of the characteristics of no advantage; [31] proposed a method for user interaction. User interface options need to be retained by the characteristics of the range, this method can preserve the model feature, but the effect of algorithm depends on the user-defined. All of the above simplified algorithms targeted for mobile terminals are not enough, so the study of new features for mobile computing terminal three-dimensional geometry data compression method has important significance.

II. Basic Concepts

Because three-dimensional model of the polygon surface is composed of a series of triangles, it is not the second order different surface graphics. In theory, curvature does not exist, but it can fit surface by way of discrete points to define each vertex's curvature.

Fitting method used here for the least squares method [32]. Its definition can be expressed as:

$$\sum_{i=1}^{t}[S_i^*(u,w) - F_i]^2 = \min\left(\sum_{i=1}^{t}[S_i(u,w) - F_i]^2\right) \quad (1)$$

Where t is the number of discrete points, S (u, w) for the discrete points of a fitting surface, $S^*(u,w)$, the least squares fitting surface is the most "recent" discrete points given by the fitting surface. F_i means the original scattered points. By calculating S (u, w), the extreme surface can be calculated least square fitting $S^*(u,w)$ set

$$S(u,w) = \sum_{i=0}^{n_u}\sum_{j=0}^{n_w} N_{i,k}(u)N_{j,L}(w)V_{ij} \quad (2)$$

Written in matrix form

$$\begin{aligned} &N'(u,w) \\ &= [N_{0,k}(u)N_{0,L}(w), N_{0,k}(u)N_{1,L}(w)\cdots N_{0,k}(u)N_{n_w,L}(w), \\ &\cdots N_{1,k}(u)N_{0,L}(w)N_{1,k}(u)N_{n_w,L}(w)\cdots N_{n_{u,k}}(u)N_{n_w,L}(w)] \\ &= \left[N'_0(u,w), N'_1(u,w)N'_2(u,w)\cdots N'_{(n_u+1)(n_w+1)}(u,w)\right] \\ &\qquad N'_{i+j}(u,w) = N_{i,k}(u)N_{j,L}(w) \\ &\qquad i = 0, 1, 2, \cdots, n_u \quad j = 0, 1, 2, \cdots, n_w \end{aligned} \quad (3)$$

$$V = \left[V_{00}, V_{01}, V_{02}, \cdots V_{0n_w}, V_{10}, \cdots V_{n_{un_w}}\right]^T \quad (4)$$

The original surface is as the parameter value. Auxiliary surfaces must meet the following two conditions: endpoint coincidence, the auxiliary surface should be greater than or equal to the range point cloud which consists of discrete surface range; and all the projection points are unique, that should not overlap existing projection.

To clarify the ideas above, we give different measurement methods to obtain the scattered data, derived mathematical model, limited space, here only four control points generated by auxiliary surface and determine its parameters are shown.

Construction of the four control points from the supporting surface for the rectangular shows as:

$$S = [1-u, u]\begin{bmatrix} V_{00} & V_{01} \\ V_{10} & V_{11}\end{bmatrix}\begin{bmatrix} 1-w \\ w\end{bmatrix} =$$

$$[(1-u)V_{00}+uV_{10},(1-u)V_{01}+uV_{11}]\begin{bmatrix}1-w\\ w\end{bmatrix}= (1-u)\,(1-w)V_{00}+(1-w)uV_{10}+(1-u)wV_{01}+uwV_{11} \quad (5)$$

First, it computes space point Q to the secondary surface projection, then sets point for Q P projection to the auxiliary surface projection of points in one direction, at last, Q projection to the base surface can be expressed as

$$d=\sqrt{(Q_x-P_x)^2+\ (Q_y-P_y)^2+\ (Q_z-P_z)^2} \quad (6)$$

where d is the distance between P and Q.

Different P values corresponding to different points, of course, where d, determined by the minimum d_{min} P. Q point is the intersection of the vertical projection to the surface, using the optimized method of steepest descent to seek for d_{min}, and then determines the secondary Projection point on the surface and the parameters u, w.

In differential geometry, the surface has two directions, if they are both orthogonal and conjugate surface at this point, it is called the main direction, and while in the surface curvature is called the principal curvatures. Surface point (non-umbilical points) are the principal curvatures of surfaces all at this point in the direction of maximum curvature $k_{max}$and minimum curvature k_{min} . The product of the k_{max} , k_{min} Surface at this point is called the Gaussian curvature k_g. Besides, their average $k_{max}+\frac{k_{min}}{2}$, this point in the surface is called the mean curvature k_m, fitted by least squares method in double cubic B-spline surface[33].

The partial derivative of formula (1) can be obtained by sampling the first derivative, second derivative and mixed partial derivatives. And note $\partial Q\,(u,v)/\,\partial u$、$\partial Q\,(u,v)/\,\partial v$、$\partial Q\,(u,v)/\,\partial u\,\partial u$、$\partial Q\,(u,v)/\,\partial u\,\partial v$、$\partial Q\,(u,v)/\,\partial v\,\partial v$ are $Q_u\,(u,v)$、$Q_v\,(u,v)$、$Q_{uu}\,(u,v)$、$Q_{uv}\,(u,v)$ 、$Q_{vv}\,(u,v)$ derivative, and then we can complete it after the partial derivatives as follows:

$$Q_u(u,v)\ =\sum_{i=1}^{4}\sum_{j=1}^{4}B_{i,j}N'_{i,k}(u)M_{j,L}\,(v) \quad (7)$$

$$Q_u(u,v)=\sum_{i=1}^{4}\sum_{j=1}^{4}B_{i,j}N_{i,k}(u)M'_{j,L}(v) \quad (8)$$

$$Q_{uv}(u,v)=\sum_{i=1}^{4}\sum_{j=1}^{4}B_{i,j}N'_{i,k}(u)M'_{j,L}(v) \quad (9)$$

$$Q_{uu}(u,v)=\sum_{i=1}^{4}\sum_{j=1}^{4}B_{i,j}N''_{i,k}\,(u)M_{j,L}(v) \quad (10)$$

$$Q_{vv}(u,v)=\sum_{i=1}^{4}\sum_{j=1}^{4}B_{i,j}N_{i,k}(u)M''_{j,L}(v) \quad (11)$$

According to formula (6) - (10), different geometry in the theory of curvature can be calculated on the surface that contains the Gaussian curvature and mean curvature[34, 35]:

$$k_g=\frac{AC-B^2}{|Q_u\times Q_v|^4} \quad (12)$$

$$[N'(u,w)V-F][N'(u,w)V-F]^T\ =$$

$$\begin{bmatrix}(N'_0(u,w)N'_0(u,w)) & \cdots & (N'_0(u,w)N'_1\,(u,w)) & \cdots & N'_0(u,w)N'_{n_u\times n_w}\,(u,w)\\ \left(N'_1(u,w)N'_0(u,w)\right) & \cdots & \left(N'_1(u,w)N'_1(u,w)\right) & \cdots & N'_1(u,w)N'_{n_u\times n_w}(u,w)\\ \vdots & & \vdots & & \vdots\\ N'_{n_u\times n_w}(u,w)N'_0(u,w) & \cdots & N'_{n_u\times n_w}(u,w)N'_1(u,w) & \cdots & N'_{n_u\times n_w}(u,w)N'_{n_u\times n_w}(u,w)\end{bmatrix}\times\begin{bmatrix}V_0\\ V_1\\ \vdots\\ V_{n_u\times n_w}\end{bmatrix} =\begin{bmatrix}N'_0F\\ N'_1F\\ \vdots\\ N'_{n_u\times n_w}F\end{bmatrix} \quad (13)$$

$$k_m=\frac{A|Q_v|^2-2BQ_u\cdot Q_v+C|Q_u|^2}{2|Q_u\times Q_v|^3} \quad (14)$$ Among

Among them, (A B C) =$[Q_u\times Q_v]\cdot[Q_{uu}\ \ Q_{uv}\ \ Q_{vv}]$

Principal curvature can be Gaussian and mean curvature calculated:

$$k_1=k_m+\sqrt{k_m^2-k_g}$$

$$k_2=k_m-\sqrt{k_m^2-k_g} \quad (15)$$

Absolute curvature which called k_a is calculated:

$$k_a=\ |k_{min}|+|k_{max}| \quad (16)$$

III. Algorithm Design

In the mobile computing, terminal shows three-dimensional model. Because of its small screen, the relatively low accuracy of model is required. However, its relative capacity to deal with the high degree of model detail is very weak. Although the compression is chosen as a general method, the compression process is still a simple, efficient geometry method. This article will take edge collapses a basic geometry compression algorithm strategy for folding studies. As terminals provide mobile computing features of less precision, it is necessary to facilitate the achievement of future progressive transmission, and reconstruction of models in the mobile computing, which use a subset of the fixed-point strategy. Fixed-point strategy in the subset indicates that how to choose the fold point to minimize the least change in the appearance of the model is crucial. Curvature of the model is defined by the curvature of the surface, so the curvature of the shape of the model structure plays a vital role in the appearance of property. While the bigger the curvature vertex model surface is, the appearance of the model is greater. Based on the above analysis, the model presented here has a change in the size of the local curvature of the folding strategy. The goal of this method is to make the fold approximation curvature of the model that small change in surface curvature, to achieve as much as possible without changing the purpose of the appearance of the original model.

A. Identification of Local Feature Points

In a geometric model, there are always some vertices which play a key role in the shape and appearance of the model. When these vertices are changed, the shape of the model will produce a larger change in appearance. Therefore, in order to keep the shape and appearance of the original model, the model should try to avoid these features to be weakened, or even be eliminated. The model that describes the key features of the vertex to the geometric model can be called the feature points.

The feature points should be retained in the model, which means that the feature points should be identified first. In addition, the feature points are usually located in the local area of the model. That area of the sampling point p_i (ie, k-nearest neighbor) is marked k. If the k value is too small, it can't reflect the local nature of the original surface, whereas if the value is too large, the performance is not local in nature and will increase the amount of calculation. Point of sampling surfaces is uniform and other k values given in the upper have lower limits, and that 3 <k <50. k values are between the surface sufficient to reflect the local nature of the original, which also shows that k values between 6 and 25 is correct. In this paper, the k value of another method for the sample point p_i sets up a small ball of radius δ, and then with obtained nearest points k in the intersection field. Based on this, points located within the area of the ball was required, otherwise not. The selection for small radius values can also be an adaptive value of unity.

In a model, the K values are determined by the curvature of all vertices, according to a reference value which can be distinguished on the vertex. Obviously, the reference value must be associated with the compressed model. This model of local area uses the average of the curvature of the vertex as a reference value, which is characterized by a vertex point, if and only if

$$C_{vertex} > \overline{C_{vertex}} \quad \text{and}$$

$$\overline{C_{vertex}} = \sum_{i=1}^{k} C_i / k \tag{17}$$

Where C_{vertex} is Curvature of the vertex, $\overline{C_{vertex}}$ is the average local curvature of the vertex in the model.

In order to maintain the appearance shape of models, although some feature points remain, for some sparse model surface, it should be retained. Therefore, this article gives a peak value of the model ω_i [16, 17], to the extent that the model surface is sparse. This paper gives the value of the model vertices calculated by Gaussian kernel function, is defined as follows:

$$\omega_i = e^{-a_i/r^2} \tag{18}$$

Where $\overline{C_{vertex}} = \sum_{i=1}^{k} C_i / k$

$$a_i = \frac{1}{k} \sum_{a_i \in \prod (p_i)} \|p_i - p_j\|^2, \; r = \max_{q_i \in \prod (p_i)} \|p_i - q_i\| \tag{19}$$

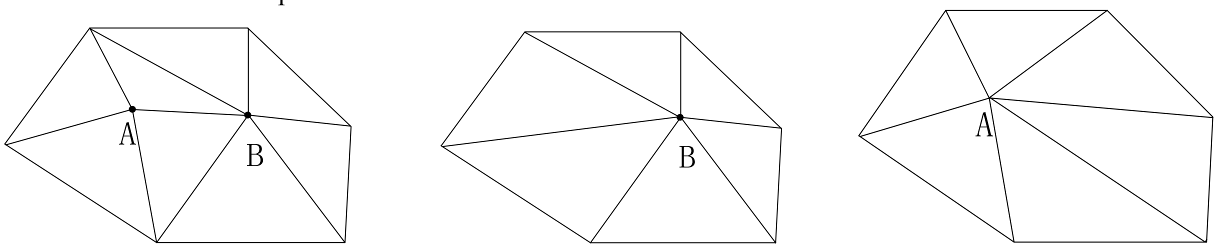


Figure 1. Edge collapse based on vertex curvature.

B. The Seclection of K Value

As the points in sample surfaces, the neighborhood set of sample points pi (that is, k nearest neighbor) in which k is an important parameter. If the k value is too small, it won't reflect the local nature of the original surface, whereas if the value is too large, the performance is not local in nature and will increase the amount of computation. In [13], Hoppe and others get a good value of k based on experiments, and then all of the samples use the same k value. Mit ra, etc. [16] is. discussed in a Noise-free case, k values is the size of the sample vector which estimate the impact point. Take an example of how to better select the k value of the formula; lack of that party act a little complicated, but the repeated iterations also get a good k value. For uniform surface sampling points, according to the density of sampling difference, Anderson [10] gives the k values of the upper and lower limits, and that is the range (3, 50) . k value is sufficient to reflect the local nature of the original surface, and this also shows that the value of k is correct between 6 and 25 [24]. That is the range of (16, 25), one of the reasons is that the point of this article deals with the nature of the surface with sample text as the same in [26, 10]; Another reason is that in large number of experiments, the range of k values has been able to meet the needs of this article, and the experimental results also show better results. This value of k, another approach is to set a sample point pi— A small ball of radius D, and then seek cross the obtained k-nearest neighbor points. The ball which is located within a small neighborhood of points is required, otherwise not. The selection of radius values can also be adaptive value of unity. Here is this small radius of the adaptive selection algorithm (where the input variable is be request samples, and the output variable is the ball radius):

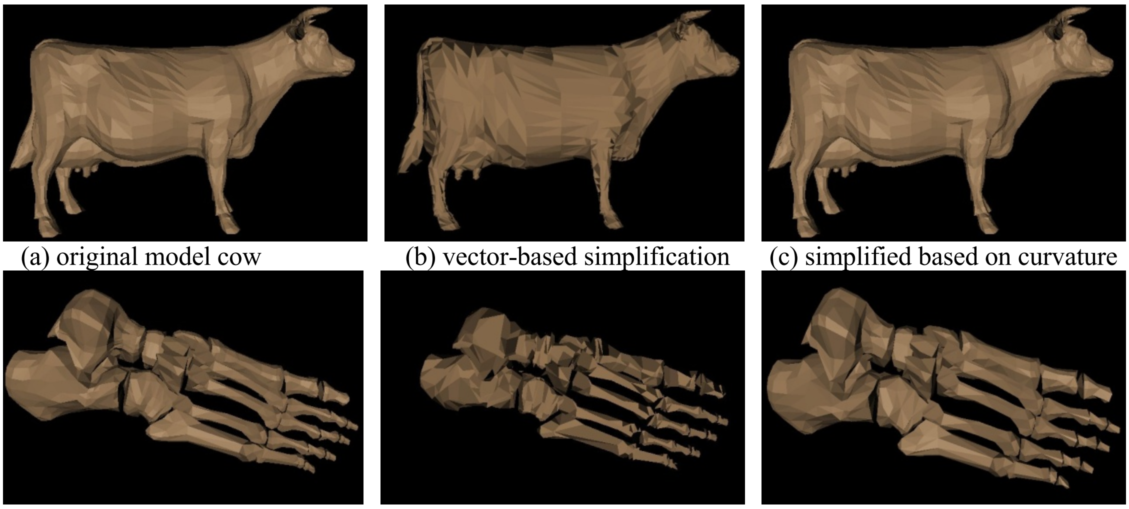
```
Calculate Radius (Point& p)
Sum=0;
{q_i}=p.GetNeighbor ();
For i: =1 to k
    d=p-q_i;
sum+=d;
end
return sum/k;
```

C. *Curvature of the Edge Collapse based on Vertex*

From the content above, the value of k determines the input model of the local area. In the local area, any two vertices adjacent to A and B, whose curvature is both less than the local mean curvature, and if curvature of the vertex A is less than the curvature of vertex B, the vertex A fold to vertex B ; on the other hand, the vertex B fold to vertex A, as shown.

Finally, from the conclusion above, we can deduce a complete and feature-based vertex curvature of the edge collapse to retain geometric compression algorithms. Specific steps of the algorithm are as follows: Compression algorithm:

Step1 calculate the curvature of the vertex;
Step2 determine the model of the local area k;
Step3 Start from the vertex 0, incremental processing, through all the vertices;
Step4 According to the curvature of the vertex, select and process the state of vertices ;
Step5 a right that can be processed vertices, the curvature of the edge collapse based on vertex;



(a) original model cow (b) vector-based simplification (c) simplified based on curvature

(d) original model bones (e) vector-based simplification (f) simplified based on curvature

Figure 2. Vector-based change and simplify the comparison based on curvature changes

TABLE I.
COW MODEL DATA

Simplified based on curvature	Original model	1 simplified	2 simplified	3 simplified
Vertices	2154	1705	935	855
Faces	4204	2906	1383	1704
Grid display frames	135fps	267fps	315fps	336fps
Model sizes	183KB	86KB	46KB	35KB

TABLE II.
COW MODEL DATA

Simplified vector-based	Original model	1 simplified	2 simplified	3 simplified
Vertices	2154	1705	935	855
Faces	4204	2906	1383	1704
Grid display frames	105fps	207fps	295fps	306fps
Model sizes	183KB	86KB	46KB	35KB

TABLE III.
BONES MODEL DATA

Simplified vector-based	Original model	1 simplified	2 simplified	3 simplified
Vertices	2904	1435	905	854
Faces	5804	2906	1083	1604
Grid display frames	165fps	287fps	295fps	306fps
Model sizes	128KB	76KB	33KB	25KB

TABLE IV.
BONES MODEL DATA

Simplified based on curvature	Original model	1 simplified	2 simplified	3 simplified
Vertices	2904	1435	905	854
Faces	5804	2906	1083	1604
Grid display frames	135fps	267fps	315fps	336fps
Model sizes	128KB	76KB	33KB	25KB

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this paper, three-dimensional model is simplified in the mobile terminal, and the PC machine is different under the same model. In the condition of PC machines, the model focuses more on the details of the computational complexity, but it is difficult in the mobile terminal (PDA) manner. Our algorithm is based on the feature points, while the calculation is simple, fast execution, and requires fewer system resources, so it is suitable for the PDA. We use two models: cow model and bones model as shown in Figure 2. (a) and (d) are cow and bones model of the original model and the models in (b) and (e) is for computer Application. [13] Sun Yat-sen, respectively Mou Lin's algorithm is effective and wide, the proposed algorithm in right (c) and (f) is effective.

As Cow original model has 2904 vertices and 5804 faces, bones model has 2154 vertices and 4204 faces. Table 1 and Table 2 show the speed of model drawing and simplified in mobile terminal. After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the save as command, and use the naming convention prescribed by your conference for the name of your paper. In this new created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

The experimental results can be seen in Figure.1.This simplification was much smoother and uniform simplified. In addition, the execution time also increased the number, as shown in Table 3 and Table 4. In the improved algorithm, after a simplified model with number of vertices, the number and size of the triangle are generally less than half the size of the original model. Overall, the algorithm can be used for any type of model and effectively reduce the number of vertices and faces, while preserves effective storage space of the simplified model. With the model reduced, faces display is faster. This real-time display and handheld device applications have particular significance.

V. CONCLUSIONS AND FUTURE WORK

Memory storage and the screen for small mobile terminals are both small, and computing capacity is weak, moreover, limited wireless network bandwidth influence its performance. The paper proposes a three-dimensional model based on surface curvature and characteristics of the vertex points to retain the edge collapse simplification algorithm. Through the bi-cubic B-spline surface, we can approximate the three-dimensional model of the surface model obtained on the surface of the curvature of each vertex, then the vertices of the sparse model surface level and the model surface curvature localized area to set a threshold, at last, the model vertex curvature of the local region were compared with the threshold to determine the selected feature points. The algorithm is efficient, and better keeps the details of the model characteristics, which satisfies the mobile terminal and network needs. Experimental results show that, taking into account the curvature of the simple Method can better preserve the details of the model features, so this feature of the text is applicable to more obvious model.

Among the future work, we will also add texture to the model, so that the simplified model with texture, while adding the appropriate model data redundancy code.

ACKNOWLEDGMENT

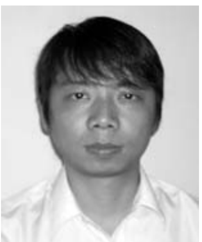
This work was supported in part by the National natural science foundation of China (Grant No. 61170098, 61170214), Zhejiang natural science key foundation (Grant No. Z1080232, No. Z1101340), and it was also supported by Scientific Research Fund of Zhejiang Provincial Education Department (Grant No. Z201018041).

REFERENCES

[1] Schroeder W J, Zarge J A. Decimation of triangle meshes. Computer Graphics, 1992, 26 (2): 65-70
[2] Low K L, Tan T S. Modeling simplification using vertex clustering. In: Proceedings of the ACM Siggraph, Los Angeles, CA, USA, 1997: 75-82
[3] Hoppe H. Progressive meshes. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 1996: 99-108
[4] Garland M, Heckbert P S, Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 1997: 209-216
[5] Oliver Matias van Kaick, Helio PedrinI. A Comparative Evaluation of Metrics for Fast Mesh Simplification. Computer Graphics forum, 2006, 25 (2): 197-210
[6] Kho Y, Garland M. User-guided simplification. In: Proceedings of ACM Symposium on Interactive 3D Grphics, San Diego, CA, USA, 2003: 123-126
[7] Rogers D F. An Introduction to NURBS: with historical perspective. USA: Academic Press, 2001.
[8] Mei Xiangming, HUANG Jing-z. Differential geometry. Beijing: Higher Education Press, in 2002.
[9] Piegl L, Tiller W. The NURBS Book. 2nded. Berlin: Springer, 1997.
[10] Mitra N J, Nguyen A. Estimating surface normals in noisy point cloud data. In Proc. Of the nineteenth conference on Computational geometry, San Diego, 2003.
[11] Gumhold S, Wang X, Macleod R. Feature extraction from point clouds. In 10th International Meshing Roundtable, Sandia National Laboratories, 2001: 293-305.
[12] Pauly M, Kobbelt L, Gross M. Multire solution modeling of point-sampled geometry. CS Technical Report #378, September 16, 2002.
[13] Schroder W., Zarge J., Lorensen W. Decimation of triangle meshes, Computer Graphics, 26 (2), 1992: 65-70
[14] Kobbelt L., Campagna S, Seidel H. A general framework for mesh decimation. In: Proceedings of Graphics Interface'98, 1998: 43-50
[15] Wu E.H., Fei G.Z. A real-time generation algorithm of progressive mesh for changing environments. International Journal of Virtual Reality, 4 (3), 2000: 49-54
[16] Lindstrom P. Out-of-Core construction and visualization of multire solution surfaces. In: Proceedings of ACM Symposium on Interactive 3D Graphics'03, 2003: 93-102
[17] Garland M., Shaffer E. A Multiphase approach to efficient surface simplification. In: Proceedings of IEEE Visualization'02, 2002: 117-124
[18] Hoppe H. Progressive meshes. In: Proceedings of ACM SIGGRAPH'96, 1996: 99-108

[19] Hoppe H. View-dependent refinement of progressive meshes. In: Proceedings of ACM SIGGRAPH'97, 1997: 189-198
[20] Hoppe H. Smooth view-dependent level-of-detail control and its application to terrain rendering. In: Proceedings of IEEE Visualization'98, 1998: 35-42
[21] Hoppe H. Efficient implementation of progressive meshes. Computers & Graphics, 22 (1), 1998: 27-36
[22] Xia J.C., Varshney A. Dynamic view-dependent simplification for polygonal models. In: Proceedings of IEEE Visualization'96, 1996: 327-334
[23] Catmull E., Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design, 10 (6), 1978: 350-355
[24] Doo D., Sabin M. Behavior of recursive division surfaces near extraordinary points. Computer-Aided Design, 10 (6), 1978: 356-360
[25] R. Zhu, "Intelligent Rate Control for Supporting Real-time Traffic in WLAN Mesh Networks," Journal of Network and Computer Applications, vol. 34, no. 5, pp. 1449-1458, 2011.
[26] R. Carey, G. Bell, and C. Martin. The Virtual Reality Modeling Language ISO/IECDIS14772-1 . http: //www.vrml.org/Specifications.VRML97/DIS, 1997.
[27] P. Cignoni, C. Montani, D. Rocchini and R. Scopigno. Metro: Measuring error on simplified surfaces. IEEE Computer Graphics Forum, 17 (2): 167–174, 1998.
[28] John G. Cleary, Radford M. Neal, and Ian H. Witten. Arithmetic coding for data compression. Communications of the ACM, 30 (6): 520–540, June 1987.
[29] Daniel Cohen-Or, David Levin and Offir Remez. Progressive compression of arbitrary triangular meshes. In Proceedings Visualization 99, pages 67–72. IEEE, Computer Society Press, Los Alamitos, California, 1999.
[30] Michael Deering. Geometry compression. In Proceedings SIGGRAPH 95, pages 13–20. ACM SIGGRAPH, 1995.
[31] N. Dyn, D. Levin and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control.ACM Transactions on Graphics, 9 (2): 160–169, 1990.
[32] T. Funkhouser and C. Sequin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In Proceedings SIGGRAPH 93, pages 247–254. ACM SIGGRAPH, 1993.
[33] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In Proceedings SIGGRAPH97, pages 209–216. ACM SIGGRAPH, 1997.
[34] André Guéziec, Frank Bossen, Gabriel Taubin and Claudio Silva. Efficient compression of non-manifold polygonal meshes. In Proceedings Visualization 99, pages 73–80. IEEE, Computer Society Press, Los Alamitos, California, 1999.
[35] Stefan Gumhold and Wolfgang Strasser. Real time compression of triangle mesh connectivity. In ProceedingsSIGGRAPH 98, pages 133–140. ACM SIGGRAPH, 1998.
[36] Chandrajit L. Bajaj, Valerio Pascucci and Guozhong Zhuang. Progressive compression and transmission of arbitrary triangular meshes. In Proceedings Visualization 99, pages 307–316. IEEE, Computer Society Press, Los Alamitos, California, 1999.

Wang Xun received the Doctor's degree in department of computer science from Zhejiang University. He is a professor in the department of computer and electronic engineering of Zhejiang Gongshang University. His research interests are in multimedia information retrieval, pattern recognition, and mobile networks. and statistical machine learning.



Bai-lin Yang received the Doctor's degree in department of computer science from Zhejiang University in 2007. He is an associate professor in the department of computer and electronic engineering of Zhejiang Gong-shang University. His research interests are in moblie graphics, realtime rendering and mbile game.