A Simulated Annealing Algorithm for Progressive Mesh Optimization

Junjie Peng, Feixiang Chen*
School of Information Science and Technology, Beijing Forestry University, Beijing, China
Email: peng_junjie@126.com, bjcfxiang@163.com

Qi Wang
Graduate School, Beijing Forestry University, Beijing, China
Email: sevenwang@live.cn

***Abstract*—This paper presents a novel progressive mesh model based on collapse tree, which can accomplish the disordered expansions and collapses of the simplified mesh and is able to restore the mesh just in a specific area. With arbitrary collapse and expansion, we design an edge collapse algorithm for mesh optimization. Its summary process can be described as follows: generating a series of new simplified meshes through the two operations from the original mesh, meanwhile, controlling the process by simulated annealing algorithm to optimize the mesh constantly, looking forward to a better simplified mesh on global scale.**

***Index Terms*—Mesh Simplification; Simulated Annealing Algorithm; Collapse**

I. Introduction

Mesh model is used to describe the three-dimensional shape of the spatial entity, and is frequently applied to the fields such as geographic information systems, computer graphics, and virtual reality. With the development of science and technology, the acquisition of space 3D data became more and more easily, the spatial mesh data generated by a large number of points is increasingly large. Complex and dense meshes can exhibit entities' physical realism and sense of hierarchy, but the large amount of data brings inconvenience to model storage, transmission, calculation and graphics rendering, especially in the devices with the limit of the hardware conditions [1]. Mesh simplification could not only preserve the original feature of the model, but also perform the entity with the fewer number of meshes, so that the model can meet the users' actual needs and reduce the overhead at the same time.

The most basic problems of mesh simplification are the simplification methods and error metric. Judging from the methods of mesh simplification, the most representative methods are Schreorder's vertex decimation [2], Rossignac's vertex clustering [3], Hoppe's edge collapse [4], and Hamann's triangle decimation [5]. Vertex decimation is easy to realize, and can guarantee the vertices of the simplified model are the subset of the original model, but it needs to re-mesh the removed polygonal area. Rossignac's scheme is only suitable for the specific algorithm, and cannot guarantee the model topology, while triangle decimation is efficient but cannot be applicable to other meshes, and its implementation is more complicated. Edge collapse is relatively wide used in practice, it can be applied to any meshes with flexible expansibility, and what's more, it is easy to achieve a progressive mesh.

Error metric is to measure the similarity between the simplified mesh and the original mesh, but there is no generally accepted metrics so far. Taking the efficiency and practicality into consideration, general algorithms commonly use partial metric to guide the process of the mesh simplification. Schreorder used the average distance of the point to the plane as a local error metric in his algorithm [2]; Turk took the neighborhood curvature as the measure standard and can maintain the characteristic geometric detail of meshes [6]; Local quadratic error metric method is applied to calculate the approximate error by Garland [7]. Some algorithms, both closely integrated. In some of the algorithms, the two basic problems are closely Contact and blend together, for instance, Eck's simplification model based on wavelet technology [8], and Cohen's envelope mesh simplification model of the global error [9]. Some new scheme introduce concept in other areas to measure the cost [10].

Most existing algorithms [11-17] do the mesh simplification based on the local optimum to current mesh. This could improve the efficiency of the algorithm, however, would result in a cumulative error in the simplifying process, thereby reducing the accuracy of the algorithm. Hoppe gave a progressive mesh optimization method based on energy assessment, it could get a better effect of simplification, also to establish detail mesh model of different levels. However, the algorithm is not efficient, and the progressive mesh model including most other general models comes to a poor effect when establishing models at the level of higher detail in a specific area.

This work was supported by the Special Funds for the Scientific Research of the Forest Public Welfare Industry under Grant (No.200904003-6).

*Corresponding author, Feixiang Chen, bjcfxiang@163.com.

doi:10.4304/jmm.8.4.323-330

In this paper, we propose a progressive mesh optimization algorithm with less computation and it could preferably achieve the refinement of a particular region based on the collapse tree. General progressive mesh achieve the level of details by recording the relevant information in each step without noticing the relationship between every step, so the progressive mesh is limited to the collapse sequence. In our scheme, not only the collapse information but also the relation information is stored by collapse tree. So collapses and expansions in disorder are available and users can refine a specific area without limits. By introducing the simulated annealing algorithm we optimize the simplified mesh through the two operations, finally resulting in better simplified mesh.

II. Problem Description

Mesh model contains the information of both data points' space coordinates and network topology: data points are the basic data in mesh model and are also the original data of constructing the network model; network topology information represents the association relationship among vertices, edges and faces, which reflected in graphics is the model's meshes. Thus, the mesh model can defined as:

$$M = (K, V) \tag{1}$$

where V represents the set of vertices and V= {v_1, v_2… v_n}, K is topology information of mesh model.

Mesh simplification can be divided into two situations, one is simplifying in a certain residual threshold, and the other is under a certain compression ratio. Due to error metrics in vary algorithms are different and there is no unified standard, the application of first situation is restricted to the specific algorithms. So in this paper, we choose the latter to describe our mesh simplification.

Mesh simplification is to reduce meshes (or vertices) to the greatest extent while preserving the original mesh shape features as many as possible [18]. When the number of vertices is fixed, determining which simplified mesh is better should measure the matching degree between simplified mesh and original mesh. Therefore, we need to design a simplification method and establish a matched method of error metrics as well. By simplifying the mesh according to the error metrics, more satisfied meshes are obtained.

III. Problem Solution

A. Method of Mesh Simplification

There are many ways to simplify meshes and different methods have different advantages and disadvantages as we have discussed. Taking the expandability and simplicity into consideration, we finally adopt the edge collapse as the basic operation for our simplification process. Any meshes such as quad meshes [19] can be subdivided to triangle meshes, to describe the algorithm conveniently and uniformly, we take triangle mesh as the example, and introduce the basic concepts.

(1) Collapse point

Collapse point refers to the new vertex generated in the process of mesh simplification. During multiple collapse processes, collapse point can be collapsed again. Hence, the collapse point can be divided into two kinds. One is visible collapse point, that hasn't been collapsed again after being created, and it is one of vertices in the current mesh. In the view of users it is visible. The other kind is intermediate collapse point, which has been collapsed more than twice and is invisible. Visible collapse point and invisible collapse point is relative to a specific mesh. As shown in Fig. 1 collapse point v_{12} is visible in M_2 (v_{12} is the vertex of M_2), while it is invisible in M_3. Therefore, for a certain simplified mesh, the vertices can be divided to original data vertex and visible collapse point, let primitive mesh model as $M_0 = (K_0, V_0)$, then any simplified mesh can be represented as:

$$M = (K, V_s, V_v) \tag{2}$$

where $V_s \in V_0, V_s \bigcup V_v = V$ and V_v is the set of visible collapse vertices in M.

(2) Relevant vertex

In the process of collapses, it is necessary to record the relevant vertex of collapse edge in order to reversed recovery which is the base of progressive mesh. Relevant vertex is another vertex that constitutes triangle besides the two of collapse edge. Generally, collapse edge that is not the border corresponds to two relevant vertices, while the border corresponds to only one relevant vertex. As shown in Fig. 2, the relevant vertices corresponding to collapse point v12 are v_6 and v_7. Hence, let collapse vertices be v, the vertices been collapsed are v_1 and v_2, the relevant vertices r_1 and r_2, thus one collapse can be noted as:

$$v(v_1, v_2, r_1, r_2) \tag{3}$$

(3) Collapse tree

In our scheme, the collapse tree is the basic unit to record collapse operations. Its data structure is binary tree owed to its name. For any visible collapse point $v \in V_v$ in simplified mesh, there is a corresponding collapse tree, such as v_{12} in M_2, v_{123} in M_3 shown in Fig. 2.

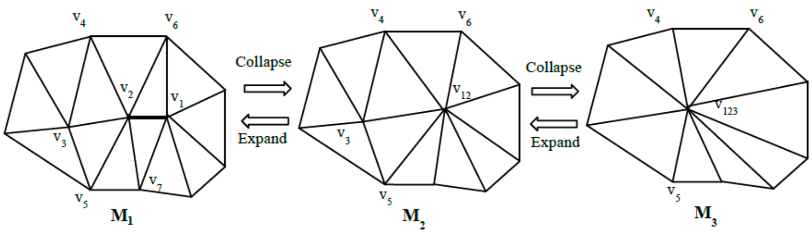

Figure 1. Collapse and Expansion

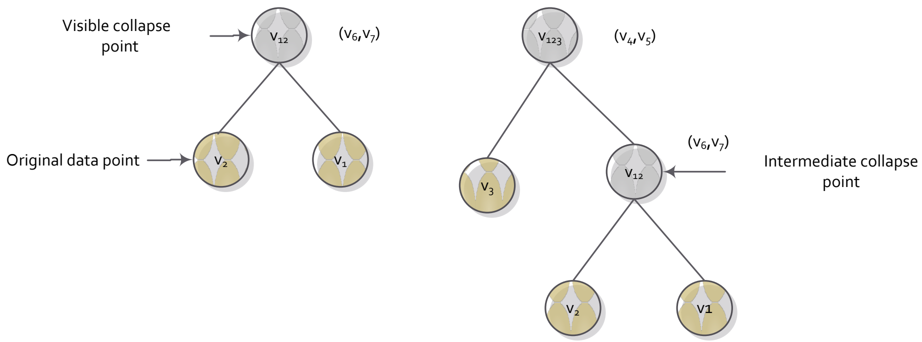

Figure 2. Collapse Tree

The pairs beside the collapse vertices are relevant vertices in Fig. 2, particularly for the collapse vertex that has only one relevant vertex the vacant position puts 0 instead. The two child of any non-leaf node corresponds to the two vertices of collapse edge, and every collapse operation equals two collapse trees correspond to the two vertices of collapse edge composes to a new binary tree, and the tree root is the new vertex after collapse. So every visible collapse vertex has a collapse tree rooted with it. We use $T(v)$ to represent the collapse tree rooted with v. Thus, the results of this progressive simplification are the simplified mesh and the collapse trees of the mesh's visible collapse points, which store the collapse information of the simplification process.

For a collapse tree, we can get its leaf nodes which are the vertices of the original mesh noted as S by tree traversal. Then it is easy to obtain the relevant area Star(S) of the original mesh from the points set S. while with the collapse tree root v, we can get the relevant area Star(R) of the current mesh, then calculate the local error between the Star (S) and Star(R) to regional mesh. The error is called collapse error $ET(v)$ and we use it to represent the collapse error of the collapse tree rooted with v, in particular $ET(v_0=0)$ if $v_0 \in V_0$.

B. Expansion of Simplified Mesh

Progressive mesh [20] can be recovered though the reversed operations based on the recorded collapse information. In this paper, the reversed operation is called expansion. For a general collapse progressive mesh model, the model can be expanded successfully only through expansions strictly in accordance with the order of collapse. However, with the extra information stored in collapse tree, our simplified mesh can be expanded in disorder

The main factors to block the expansions in disorder are the changes of relevant vertices. The relevant vertices recorded in the process of collapse will change during the process of collapse and expansion, expansions in order can ensure that the relevant vertices can be restored in sequence, so that it can guarantee the accuracy of the relevant vertices in every expansion operation. But if being expanded disorderly this can't be guaranteed.

Therefore, in order to realize the disordered expansion, we need to solve the two problems: (1) How to update the information of the collapse points when its relevant vertices are collapsed in the later process; (2) How to deal with the change when the relevant vertices are expanded.

To the first problem, according to the collapse tree we can track back to the visible point in current mesh from the original relevant vertex and replace it. As Fig. 2, the relevant vertex of v_{123} is v_4, and we can track back to v_{46} through the collapse tree, in this way the mesh can be expanded successfully.

The solution to solve the second problem is to update the information about relevant vertices of the collapse vertex. For example, when collapse point A (A_1, A_2, B, C) is expanded, we should update the collapse point D (D_1, D_2, A, E) of which A is one of the relevant vertex. The specific updating method should be discussed in the two situations:

(1) In the two vertices generated by the expansion of collapse point A, there is only one vertex that is linked with D, let the vertex be A_1, then the relevant vertex A should be updated as A_1.

(2) When the collapse point D is linked with the two vertices generated by expansion of A, the situation will be more complicated. The brief derivation is given as follows. According to the assumed condition, D is linked with both A_1 and A_2, then there must be one vertex among A's relevant vertices B and C being D's expansion vertices (D_1or D_2) or the ancestor node of D's expansion vertices. Suppose Ps (B) =D_1 (Ps (P) stands for any node on the vertex P's root path in the collapse tree). If D is expanded at present there will be an ambiguity as shown in figure 3, but as Ps (B) =D_1 and B is the relevant vertex of A, D_1 is linked with A_1, A_2, situation b can be eliminated. That is to say, D's relevant vertex A should be updated to A_2, which is linked to D_2.

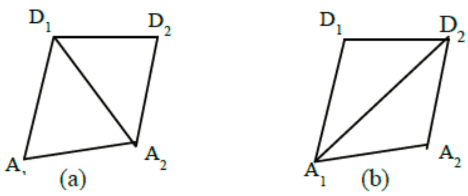

Figure 3. Two ways of expansion

Taking the mesh M_3 as an example in Fig. 1, the edge v_4v_6 is folded to v_{46} (v_4, v_6, v_{123}, 0), and then we firstly expand v_{123} and then v_{46} instead of the collapse sequences, as follows in Fig. 4.

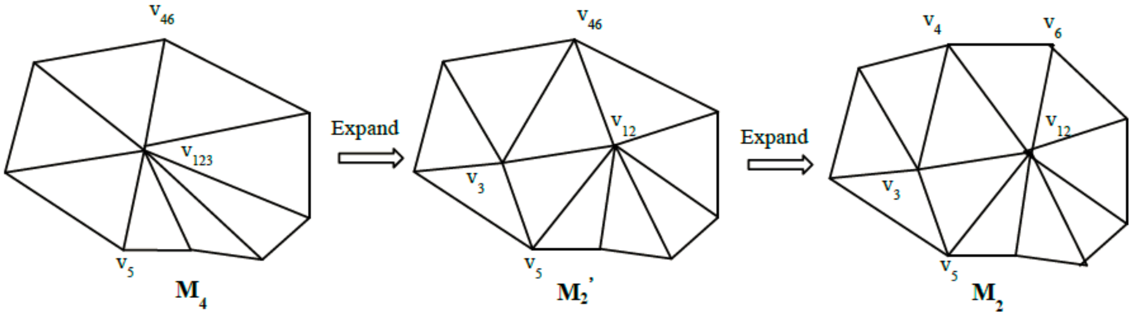

Figure 4. Expansion in disorder

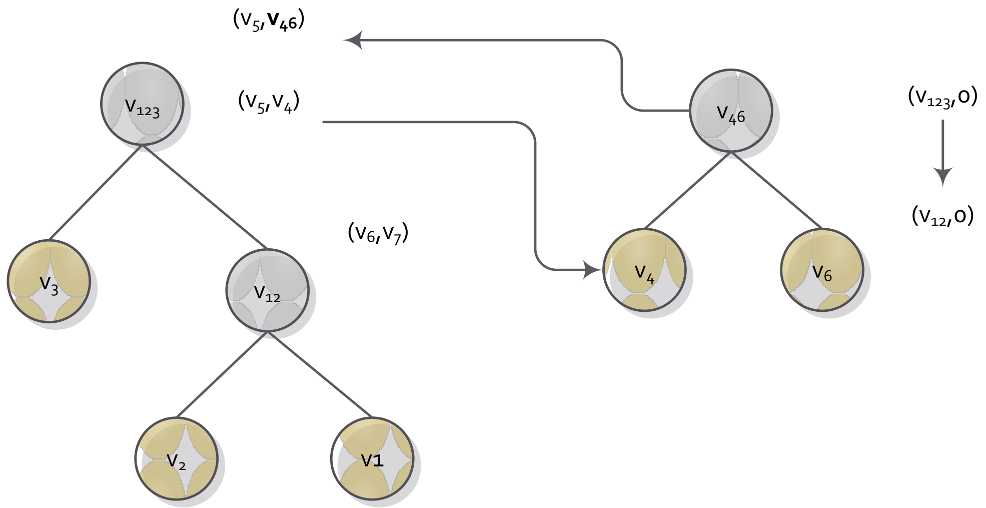

Figure 5. Trace collapse information

When expanding v_{123} (v_{12}, v_{13}, v_4, v_5), there is a problem that v_4 is no longer in mesh (v_4 is invisible). But with the collapse tree, we can track back to root node corresponding to v_4, then expand it, and get mesh M_2', meanwhile update v_{46} (v_4, v_6, v_{123}, 0), which takes v_{123} as relevant vertices. It's easy to reckon that v_{12} is linked to v_4 and v_3 according to the rules above, so we choose v_{12}, connected to v_6, to replace v_{123}, thus M_2' can be restored to M_2 by expanding v_{46}, and then M_2 can be restored to the original mesh by spreading v_{12}.

C. Progressive Mesh Model Based on Collapse Tree

Based on the theoretic foundation, we can build a progressive mesh model based on collapse tree. The model takes the collapse points as the base to store the collapse information and the hierarchical relation between the collapse points, which makes it possible to implement the disordered expansions of any visible point in simplified mesh, and the algorithm in the following section is based on this.

Generally, the algorithms just use collapse steps to store the information of progression. Those mesh model can be expanded successfully by inversing the collapse sequence. However they are inappropriate to the expansions just in one certain area. For example, in the progressive transmission, detail presentation and virtual reality in some limited situation [21], we need the detailed information of certain area, and then we cannot but traverse most collapse information in other areas as well which is actually unnecessary. Our mesh model, by contrast, can expand the mesh by choosing collapse points in certain areas and avoid redundant transmission and operations.

D. Measurement of Mesh Error

Error measure is used to quantify the difference between the input model and the output model. We need the difference to tell which simplified mesh is better. So how to measure the match degree between the simplified mesh and the original is one of the key problems that should be solved and what's more the error metric is the basis of simplification, and is used to guide the simplification process.

There are several schemes to measure the error. One simple way can be defined as follows, as the distance between corresponding sections of the meshes.

$$E(M_1, M_2) = \max_{v \in M_1} d_v(M_2) \tag{4}$$

where $d_v(M)$ is the distance from vertex v to mesh M, $d_v(M) = \min_{w \in M} \| v - w \|$ ($\| \cdot \|$ means the Euclidean distance of the two vectors).

Sometimes we need a mean result, with the distance define above, it's easy to get the mean distance.

$$E(M_1,M_2)=\frac{1}{|M_1|}\int_{M_1} d_v(M_2)ds \tag{5}$$

Note that this definition of distance is not symmetric, that is to say $E(M_1,M_2)$ may not equals $E(M_2,M_1)$. So in this paper, we adopt a two-sided distance (Hausdorff distance) to measure the difference between the two mesh models [22]. Let $E(M_1,M_2)$ be the distance between mesh M_1 and M_2, then

$$E(M_1,M_2)=\max(\max_{v\in M_1} d_v(M_2),\max_{v\in M_2} d_v(M_1)) \tag{6}$$

E (M_1, M_2) is used to measure the maximum deviation between the two models in our scheme.

IV. ALGORITHM DESCRIPTIONS

Simulated Annealing Algorithm (SAA) is a calculation model simulating the annealing process of objects. It is first proposed in 1983 by S. Kirkpatrick, et al. [23], and it is a widely used random search algorithm for solving global optimization problems [24, 25]. Simulated annealing algorithm is different from the traditional random search strategy, it not only introduces the appropriate random factors, but also borrows the natural mechanism of the annealing process in the physical system [26], and consequently there is a greater probability of global optimal solution. Therefore, we apply SAA to the mesh simplification problem to get a global optimal mesh or better simplified mesh from a global perspective. Combining the principles of the general simulated annealing algorithm with mesh simplification, the proposed algorithm consists of the following steps:

Step 1: (Data Preparation) Input the original mesh M_0. Calculate the number of collapses according to the compression rate.

Step 2: According to the number of collapses, collapse the original mesh M_0 randomly to generate the initial mesh as the state of the initial solution M, and in the process record each collapse step in accordance with the proposed method. Set the parameters, including the initial temperature T (sufficiently large), and the iterations L for every T.

Step 3: In the mesh of current solution, take a visible collapse point randomly to do the expand operation, then select one edge to fold, so that a new solution M' could be produced.

Step 4: Calculate the energy difference ΔE before and after step 3.

Step 5: If the energy is reduced (ΔE<0), accept the new solution as current solution, otherwise, accept M' as the current solution with the probability of $e^{-\Delta E(M_1,M_2)/T}$.

Step 6: If the iterations does not exceed the setting and the accuracy does not meet the requirements, turn to step 4.

Step 7: Finish. Output the current solution as the result.

A. Generation of Iinitial Mesh

Our simplification is under a certain compression rate, so the number of faces in the ultimate mesh could be calculated by the compression rate and the original mesh. Then it's easy to get the initial mesh for the algorithm containing specific number of faces. The initial mesh is generally not optimal, and we used it as the initial solution of the simulated annealing algorithm. There are two solutions to generate initial mesh, one is to do the collapse operation on the edge randomly; the other to get the initial mesh is by the existing schemes of mesh simplification, then use SSA to optimize the initial mesh, something like the scheme in [27, 28]. According to the compression proportion, the initial mesh is generated after certain collapses from the original mesh, and the collapse information required by the progressive mesh model is recorded as well.

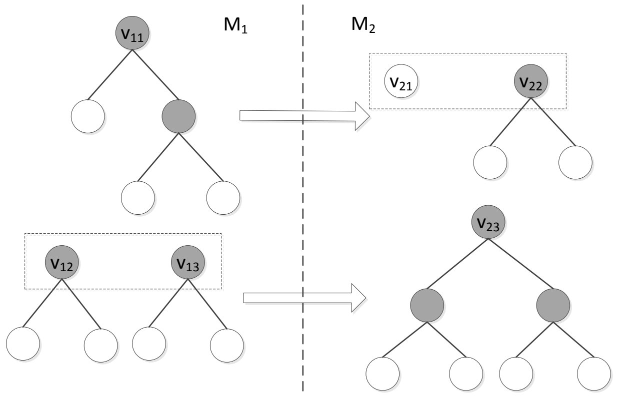

Figure 6. The changes after collapse and expansion

B. Transition of State

In order to ensure the certain number of triangle faces in the mesh (i.e. the compression rate remains unchanged), we do expand and collapse operation respectively for one time to the current mesh as the state transition in SSA. The concrete steps are: select a visible point from the collapse points in current mesh to do expand operation, and then select the one edge of the mesh to collapse. For the new solution, we need to determine whether it is acceptable or not based on simulated annealing algorithm. In order to avoid large number of repetition calculations, the basic method is compared the difference between the two meshes. The normal algorithms are just to compare the relative error between the current mesh and the former mesh, so the difference error is shown as:

$$\Delta E(M_1,M_2)=E(M_2,M_1) \tag{7}$$

However, the method would lead the error to be accumulated. So we firstly compare the current mesh and the former to the original mesh, and then use their difference value.

$$\Delta E(M_1,M_2)=E(M_2,M_0)-E(M_1,M_0) \tag{8}$$

The amount of the theoretical error calculation is very large, so in the actual application, representative sample points in both mesh surfaces are usually used to do

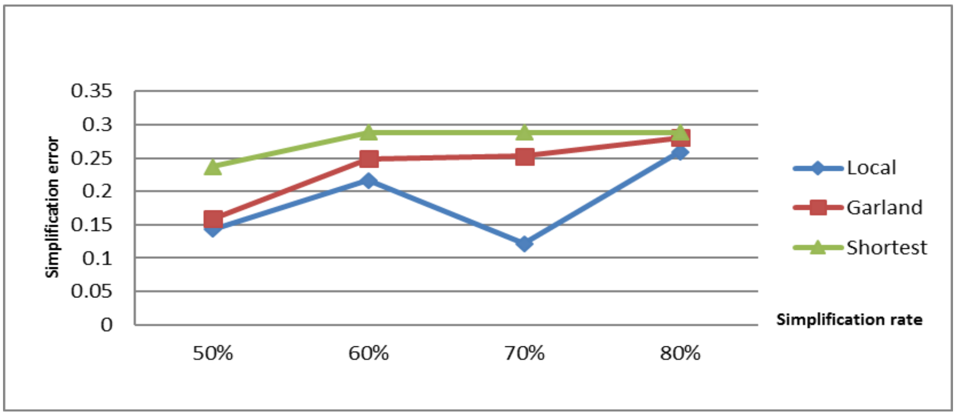

Figure 7. The Compare results of the model cow

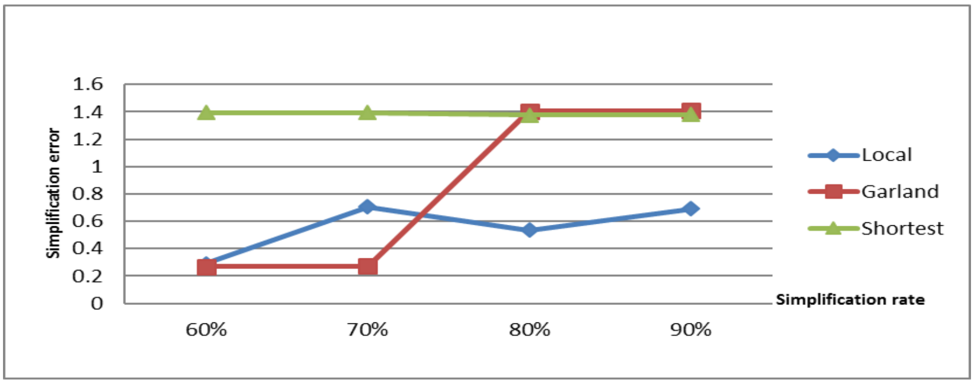

Figure 8. The Compare results of the model Porsche

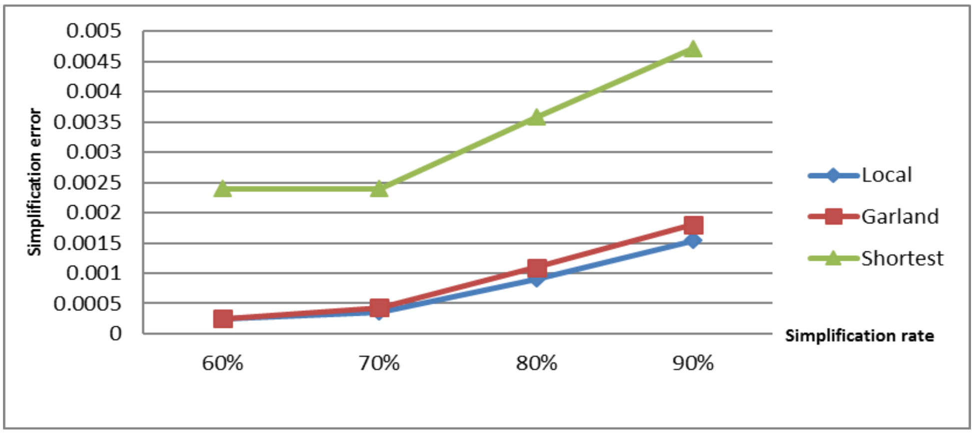

Figure 9. The Compare results of the model bunny

approximate calculation (these sampling points including at least the vertex of each mesh). The same approximate method is adopted in this paper. The process of state transition will involve six collapse trees (including the original data points). The Fig. 6 shows the trees of both former state and new state respectively.

TABLE I. BASIC DATA OF THE TEST MODEL

Model	Vertex	Face	Area	Diagonal
Cow	2903	5804	21.1821	5.603207
Porsche	5247	10474	276.7388	15.814117
Bunny	35947	69451	0.0571	0.250246

V. EXPERIMENTAL RESULTS AND ANALYSIS

In order to test the effect of our simplification algorithm, we do some comparison test between our algorithm and some other classical algorithms (including the Garland method and the Shortest Distance method). The error is calculated using the Metro. Metro is a tool designed to evaluate the difference between two triangular meshes, and it adopts an approximated approach based on surface sampling and point-to-surface distance computation. This experiment is carried out without introducing a new point, and therefore the collapse point can only choose one from the two vertices of the collapse edge with smaller error.

In the experiments we have carried out the tests with three models respectively, the basic data of the three models are shown in Tab. 1. For each model, the comparisons of the three simplification algorithms are carried out at the different compression ratios. The results are shown in Fig. 7, 8, 9.

From the results, we can conclude that our algorithm has lower error relative to the other algorithms at higher simplification rate. In addition, the algorithm can be used as a method of optimization to other algorithms. For a more detailed explanation, we use other algorithms to simplify the model at the fixed compression ratio, take the result as the initial solution, and then optimize it with our algorithm, thereby to obtain more preferably simplified mesh.

VI. Conclusion

The progressive mesh model based on collapse tree can collapse and expand the mesh disorderly. Depending on this characteristic, it is able to convert the mesh optimization problem to the combination optimization problem. In this paper, we adopt the simulated annealing algorithm to search the more optimal simplified mesh on global scale. During the optimization process, by tracking the relevant information of the original mesh according to collapse tree and calculating the error with the original information, we can avoid progressive deviation caused by calculating the current mesh merely. Moreover, taking the visible collapse points as the expansion unit can meet the requirement of refining certain areas in simplified model, which can be used to improve the mesh's progressive transmission and multiple-level detail model.

References

[1] Y. P. Zhang, H. Xiong, et al. "A survey of simplification and multiresolution techniques for massive meshes", *Journal of Computer-Aided Design & Computer Graphics*, 2010.

[2] W. Schroeder, J. Zarge, W. Lorensen, "Decimation of triangle meshes", In: *ACM SIGGRAPH Computer Graphics*, vol. 6, no. 2, pp. 65-70, 1992.

[3] J. Rossignac, P. Borrel, "Multi-resolution 3D approximations for rendering complex scenes", In: *Modeling in Computer Graphics, Eds.Springer-Verlag*, pp. 455-465, 1993.

[4] H. Hoppe, T. DeRose, T. Duchamp, et. al. "Mesh optimization", In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques. ACM*, vol. 27, pp. 19-26, 1992.

[5] B. Hamann, "A data reduction scheme for triangulated Sur-faces", *Computer aided geometric design*, vol. 11, no. 2, pp. 197-214, 1994.

[6] G. Turk, "Re-tiling polygonal surfaces". In: *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 55–64, 1992.

[7] M. Garland, P. S. Heckbert, "Surface simplification using quadric error metrics", In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co*, pp. 209-216., 1997.

[8] M. Eck, T. DeRose, T. Duchamp, et al. "Multiresolution analysis of arbitrary meshes", In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. ACM*, pp. 173-182, 1995.

[9] J. Cohen, A. Varshney, D. Manocha, et al. "Simplification envelopes", In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM*, pp. 119-128, 1996.

[10] L. P. Xing, K. C. Hui, "Entropy-based mesh simplification", *Computer-Aided Design and Applications*, vol. 7, no. 6, pp. 911-918, 2010.

[11] Z. G. Pan, K. Zhou, J. Y. Shi, "A new mesh simplification algorithm based on triangle collapses", *Journal of computer science and technology*, vol. 16, pp. 28, 2001.

[12] X. L. Liu, Z. Y. Gao, D. Peng, et al. "Collapse simplification based on sharp degree", *Journal of software* vol. 16, pp. 669-675, 2005.

[13] J. Wang, M. Y. He, "An algorithm for mesh simplification based on the importance of vertex and its application", *Application research of computes* vol. 21, pp. 41, 2004.

[14] M. Garland and P. S. Heckbert, "Simplifying surfaces with color and texture using quadric error metrics," In: *Visualization '98*, 1998.

[15] D. Luebke, C. Erikson, "View-dependent simplification of arbitrary polygonal environments" In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., pp. 199-208, 1997.

[16] F. Wan, F. He, et al. "A simplification algorithm based on appearance maintenance", *Journal of Multimedia*, vol. 5, no. 6, pp. 629-638, 2010.

[17] S. Zhang, J. Zhao, et al. "Feature aware multiresolution Animation Models Generation", *Journal of Multimedia*, vol. 5, no. 6, pp. 622-628, 2010.

[18] Z. G. Pan, X. H. Ma, J. Y. Shi, "Overview of multiple level of detail creation", *Journal of image and graphics* vol. 3, pp. 754-759, 1998.

[19] M. Tarini, N. Pietroni, P. Cignoni, et al. "Practical quad mesh simplification", *Computer Graphics Forum*, vol. 29, no. 2, pp. 407-418, 2010.

[20] H. Hoppe, "Progressive meshes", In: *Proceedings of the 23rd annual conference on computer graphics and interactive techniques. ACM, 1996*.

[21] N. Aspert, D. Santa-Cruz, T. Ebrahimi, "Mesh: Measuring errors between surfaces using the hausdorff distance", In: *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference*. IEEE, pp. 705-708, 2002.

[22] S. Kirkpatrick, D. G. Jr., M. P. Vecchi, "Optimization by simulated annealing", In: *Science*, vol. 220, no. 4598, pp. 671-680, 1983.

[23] A. Dekkers, E. Aarts, "Global optimization and simulated annealing" In: *Mathematical programming*, vol. 50, pp. 367-393, 1991.

[24] L. Ingber, "Adaptive simulated annealing (ASA): Lessons learned", Control *and cybernetics*, 1996.

[25] T. Yin, Y. Han, Y. Chen, et al. "WebVR——Web virtual reality engine based on P2P network", *Journal of Networks*, vol. 6, no. 7, pp. 990-998, 2011.

[26] R. L. Yang, J. F. Gu, "An Efficient Simulated Annealing Algorithm for Global Optimization", *Systems engineering-theory & practice*, vol. 17, no. 5, pp. 30–36, 1997.

[27] V. Vidal, C. Wolf, F. Dupont, "Combinatorial mesh optimization", *The Visual Computer*, vol. 28, no. 5, pp. 511-525, 2012.

[28] L. Chen, M. Holst, "Efficient mesh optimization schemes based on optimal Delaunay triangulations", *Computer*

Methods in Applied Mechanics and Engineering, vol. 200, no. 9, pp. 967-984, 2011.

Junjie Peng was born in the city of Enshi, Hubei Province, China, in 1987. He graduated and received his bachelor degree in management from the information management and information system major in Beijing Forestry University in 2011. Currently he is a second-year postgraduate of the same school and his main research field is topology generation in GIS.

Feixiang Chen was born in Hubei Province of China in 1977. He received the M.S. degree in computer applied technology from China University of Geosciences in 2003, and received the Ph.D degree in cartography and geographic information system from Chinese Academy of Sciences in 2006. Now he is an associate professor in Beijing Forestry University, and his main research field is mobile GIS.

Qi Wang was born in the city of Hangzhou, Zhejiang Province, China, in 1989. He graduated and received his bachelor degree in management from the information management and information system major in Beijing Forestry University in 2012. Currently he is a first-year postgraduate of the same school and his main research field is Mobile GIS.