Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Mesh parameterization based on edge collapse*

Jiang QIAN†, Xiu-zi YE, Cui-hao FANG, San-yuan ZHANG
(State Key Lab of CAD & CG, School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)
†E-mail: winston_qian@yahoo.com.cn

Received June 8, 2008; Revision accepted Sept. 1, 2008; Crosschecked Apr. 28, 2009

Abstract: Parameterizations that use mesh simplification to build the base domain always adopt the vertex removal scheme. This paper applies edge collapse to constructing the base domain instead. After inducing the parameterization of the original mesh over the base domain, new algorithms map the new vertices in the simplified mesh back to the original one according to the edge transition sequence to integrate the parameterization. We present a direct way, namely edge classification, to deduce the sequence. Experimental results show that the new parameterization features considerable saving in computing complexity and maintains smoothness.

Key words: Edge collapse, Vertex removal, Mapping, Edge classification
doi:10.1631/jzus.A0820428 **Document code:** A **CLC number:** TP391.7

INTRODUCTION

Mesh parameterization, generally grouped according to the base domain in use (spherical parameterization, planar parameterization, etc.), is widely used in texture mapping (Levy *et al*., 2002), mesh editing (Lee *et al*., 1999), mesh morphing (Lee *et al*., 1999), and mesh compression (Khodakovsky *et al*., 2000). An important category of mesh parameterization is to parameterize over the simplified mesh, e.g., multiresolution adaptive parameterization of surfaces (MAPS) (Lee *et al*., 1998; Guskov *et al*., 2000). The construction of the base domain plays an important role in this type of parameterization as the bulk of distortion is closely related to the base complex (Khodakovsky *et al*., 2003). The parameterizations fit in this group make non-exceptional use of vertex removal, which is followed by the re-triangulation of the hole left by the removed vertex and its incident triangles (Fig.1) (Schroeder *et al*., 1992). Instead, mainstream algorithms of mesh simplification use edge collapse to construct the simplified mesh (Garland and Heckbert, 1997; Heckbert and Garland, 1997; Luebke, 2001). The choice of vertex removal is mainly based on the following reasons:

(1) The vertices of the base domain are the vertices of the original mesh, exempting the parameterizations from the necessity of processing the newly introduced vertices.

(2) The vertices adjacent to the removed vertex naturally form a parameterized circle centered by the removed vertex.

However, the use of the vertex removal scheme introduces a delicate problem: how to choose the optimum triangulation. Khodakovsky designed the priority criteria to evaluate the possible triangulations, in which the triangulation that admits a low distortion was chosen. But the solution remained imperfect because of its complexity. When the degree of the removed vertex increased, the number of possible triangulations formed the Catalan series (Cohen, 1999), which unfortunately increases exponentially.

In this study, we solve the problem by applying edge collapse to constructing the base domain. Specifically, we make the following contributions:

(1) The edge collapse has been studied and in consequence, we can easily construct the base domain

* Project supported by the National Natural Science Foundation of China (Nos. 60273060, 60333010 and 60473106) and the Research Fund for the Doctoral Program of Higher Education of China (No. 20030335064)

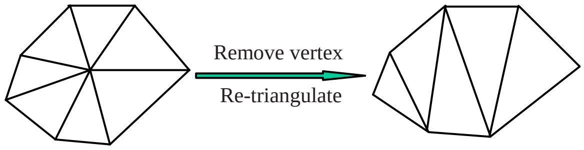


Fig.1 Vertex removal and re-triangulation

that tends to have 'equilateral' triangles. Smoothness of the parameterization surely benefits from these base domains.

(2) The edge collapse scheme evaluates edges instead of the possible triangulations, and the involved computation thus decreases significantly.

(3) The intersection set of edge collapse and vertex removal, namely the half-edge-collapse scheme, is used to capture the features of the original mesh.

In addition, Khodakovsky proposed edge classification to construct the globally smooth parameterization (GSP). The core of their algorithm is to acquire the knowledge of cross sequence between the triangles of the base domain and the edges of the original mesh, and the recording procedure of the algorithm is recursive. Thus, edge classification is time consuming and easy to introduce errors. We use a direct way to locate important results from the mappings of parameterization. The results of edge classification are utilized to map the new vertices back to the original mesh. The bi-directional mapping ensures the integrity of our parameterization.

This paper is organized as follows. In Section 2, we introduce the mesh parameterization that uses edge collapse to construct the base domain. The direct edge classification and the method to preserve features are also discussed. Section 3 illustrates how to map the new vertices introduced by edge collapse back to the original mesh and thus completes the bi-directional mapping. The experimental results are discussed in Section 4 with an analysis of our algorithm and the vertex removal based algorithm. Conclusions are given in Section 5.

APPLICATION OF EDGE COLLAPSE TO PARAMETERIZATION

In this section we describe the main components of our algorithm.

Notations

We adopt and make some changes to the notations used in MAPS and by Garland and Heckbert (1997).

A triangular mesh is denoted as a pair (V, K), where V is the set of mesh vertices $V_i=(x_i, y_i, z_i)\in\mathbb{R}^3$ $(1\le i\le N)$ and K an abstract simplicial complex which contains all the topologies. The K is a set of subsets of $\{1, 2, \ldots, N\}$. There are three types of subsets: vertices $V_i\in K$, edges $E=\{V_i, V_j\}\in K$, and faces $F=\{V_i, V_j, V_k\}\in K$. The 1-ring neighborhood of a vertex V_i is $N(V_i)$, and 1-ring neighboring faces are $F(V_i)$. We can use the conformal map to map the neighborhood $N(V_i)$ into a plane (Duchamp *et al*., 1997). The map is denoted by $Z^a(V_i)$, and $\theta(F, V)$ is the interior angle of face $F=\{V_i, V_j, V_k\}\in K$ at the vertex V_i. For a face F, its 1-ring neighboring triangles are $N_f(F)$.

In this paper, a collapse of an edge (V_i, V_j) into a vertex V^* is denoted by $(V_i, V_j)\rightarrow V^*$. The location of V^* on the edge (V_i, V_j) is set by two ratios: $C_i=V_iV^*/(V_iV_j)$ and $C_j=V_jV^*/(V_iV_j)$, where $C_i+C_j=1$. A sequence of edge collapses constructs a mesh hierarchy. The original mesh $(P, K)=(P^L, K^L)$ is successively simplified into a series of homeomorphic meshes (P^l, K^l) with $0\le l\le L$, where (P^0, K^0) is the coarsest or base mesh, namely base domain. V^0, E^0 and F^0 are the vertex, edge and face sets of the base mesh, respectively. The hierarchy of simplification is similar to MAPS. The set of related edges is marked as un-removable by the algorithm when an edge is collapsed, and for an edge E_i, the set is denoted by $R_e(E_i)$ (Fig.2).

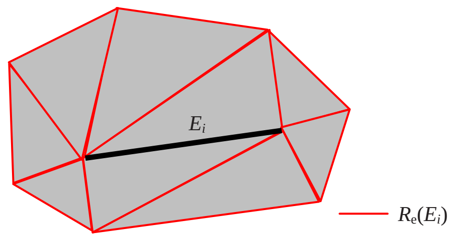


Fig.2 E_i and its related edges $R_e(E_i)$ and related faces $R_f(E_i)$ (the gray faces)

Framework of the algorithm

The edge-collapse framework is presented as follows. Firstly, the collapsed edges are prioritized based on a certain measure of errors. The linear sequence of edge collapse will then be carried out on the prioritized edges. Topological dependence that marks the related edges as un-removable defines levels of simplification in a hierarchy. An atomic mapping of a collapse edge is undertaken whenever an edge is collapsed. The mapping of the parameterization and the criteria to prioritize the collapsed edges are independent operations, and thus we have freedom to design these operations for our applications.

A level of simplification and mapping is determined as follows:

Step 1: Prioritize all the edges of the mesh (P^l, K^l) and set up the stack of collapsing edges. The selection of priority and positions of the collapsed vertex should ensure the quality of the triangle shape in the base mesh.

Step 2: The edge E_i on top of the stack is popped out for collapsing. Let V_j, V_k be the ends of E_i. We construct the conformal map of $Z^a(V_j)$ and $Z^a(V_k)$ and then map V_j, V_k in collapsed triangles through $Z^a(V_j)$ and $Z^a(V_k)$, respectively. Special care must be taken in constructing the conformal maps. The construction procedure will be illustrated in the next subsection.

Step 3: For those vertices that have already been mapped in $R_f(E_i)$, we remap them in the collapsed triangles through $Z^a(V_j)$ and $Z^a(V_k)$.

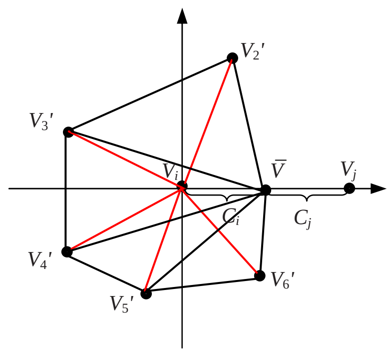
Step 4: Collapse the edge E_i and mark $R_e(E_i)$ as un-removable in the level of simplification.

A level of simplification is completed when all the edges are marked as either un-removable or collapsed. The flags that mark the edges as un-removable are cleared, and a new level of simplification will be carried out on the new mesh. The algorithm terminates when the number of left triangles satisfies the simplification ratio set by the user.

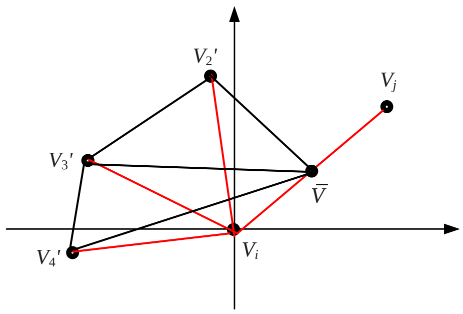
Constructing conformal maps of collapsed edge

As in MAPS (Lee *et al.*, 1998), we also use conformal maps during simplification to produce a global parameterization. However, the conformal maps must be carefully designed for the collapsed edge since an edge-collapsed scheme will be used in our parameterization. We construct the conformal maps as follows:

(1) For an end V_i of the collapsed edge $E=(V_i, V_j)\rightarrow V^*$, V_j is placed on the positive part of the X-axis while other ends of $N(V_i)$ will be enumerated anti-clockwise in the plane (Fig.3). The ratios C_i and C_j locate the position of V^* in the plane. The collapsed triangles in the mapping plane are $V^*V_2'V_3'$, $V^*V_3'V_4'$, $V^*V_4'V_5'$, $V^*V_5'V_6'$, and the point location algorithm is carried out then to find out which triangle contains the origin, namely the mapping point of V_i in the mapping plane. V_j is placed on the X-axis to ensure that V_i will always be mapped in the collapsed triangles. Or else, V_i is possible to be mapped outside any collapsed triangles in the plane. Fig.3b shows the possible mapping that fails to map V_i (V_i is not contained in collapsed triangles $V^*V_2'V_3'$ or $V^*V_3'V_4'$). If V_i is not the vertex of the original mesh, we do not have to map V_i.



(a)



(b)

Fig.3 Conformal map of a vertex
(a) A conformal map; (b) A map failing to map the center

(2) The vertices that have been mapped in $R_f(E)$ should be remapped in $Z^a(V_i)$ or $Z^a(V_j)$ in the following procedure. Let the barycentric coordinates of the vertex be denoted by (α, β, γ) and the mapping face by F in $R_f(E)$. We compute the mapping vertex's coordinates in $Z^a(V_i)$ or $Z^a(V_j)$ when $F\in F(V_i)$ or $F\in F(V_j)$ accordingly. If $F\in F(V_i)\cap F(V_j)$, the mapping vertex is likely to be mapped in both $Z^a(V_i)$ and $Z^a(V_j)$. We can select either mapping result.

Direct edge classification

Khodakovsky *et al.*(2003) proposed edge classification in GSP. The algorithm records the cross sequence of base mesh faces for every edge in the original mesh, but the process is complex and highly subject to faults. In addition, edge classification fits into the vertex removal scheme, while what we use is the edge collapse scheme. This study proposes a direct edge classification which records the same knowledge of the cross sequence. The procedure is efficient and robust. Because a geometric error has been considered in the design of our algorithm, results of edge classification are reasonable and applicable for the parameterization.

Take an edge $E=(V_i, V_j)$ for example. The ends V_i and V_j are mapped in F_i^0 and F_j^0, respectively, and it is then possible for E to have the cross sequence as follows:

(1) F_i^0 and F_j^0 are the same face in the base domain. Then E is completely contained in F_i^0 and crosses no face in the base mesh (Fig.4a).

(2) F_i^0 and F_j^0 have a common edge. We record the cross sequence of E to be (F_i^0, F_j^0) (Fig.4b).

(3) F_i^0 and F_j^0 have only a common vertex V (Fig.4c). E has two possible cross sequences: $(F_i^0, F_{i+1}^0, \ldots, F_j^0)$ and $(F_i^0, F_{i-1}^0, \ldots, F_j^0)$, as shown in Fig.5. We compute $\sum\theta(F, V)$ of the two sequences and record the sequence that has a smaller sum of angles.

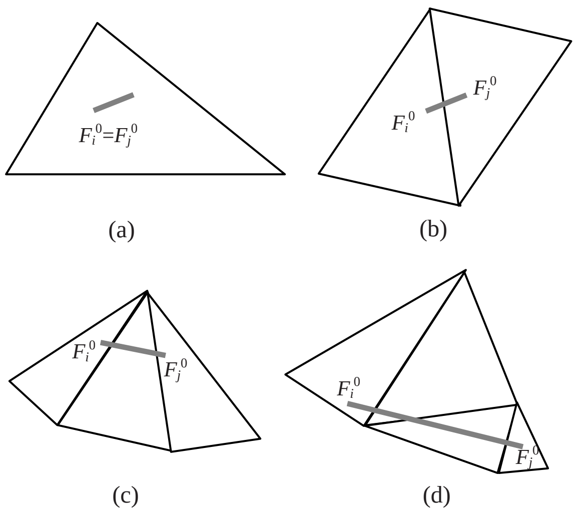


Fig.4 Possible mappings of edge ends

(a) F_i^0 and F_j^0 are the same triangle; (b) F_i^0 and F_j^0 have a common edge; (c) F_i^0 and F_j^0 have only a common vertex; (d) F_i^0 and F_j^0 are not adjacent

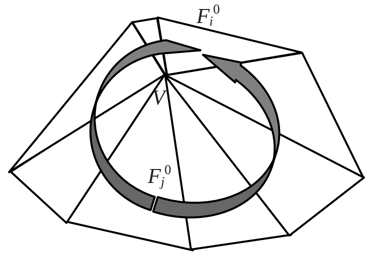


Fig.5 Possible crossings when F_i^0 and F_j^0 have a common vertex V

(4) F_i^0 and F_j^0 have no common vertices. Fig.4d shows that it is only possible for E to cross four faces. We find $F_1^0 \in N_f(F_i^0)$ and $F_2^0 \in N_f(F_j^0)$, where F_1^0 and F_2^0 are adjacent to each other. The cross sequence of E will be $(F_i^0, F_1^0, F_2^0, F_j^0)$.

MAPPING OF NEW VERTICES INTRODUCED BY EDGE COLLAPSE

The vertex removal scheme does not introduce new vertices, while the edge collapse produces new vertices in the base mesh. These vertices should be mapped back to the original mesh. The results of the edge classification map these new vertices. Steps of the algorithm are listed as follows:

Step 1: Let the edge of the original mesh E cross the base mesh faces $(F_1^0, F_2^0, \ldots, F_n^0)$. For F_i^0, $F_{i+1}^0 \in (F_1^0, F_2^0, \ldots, F_n^0)$, F_i^0 and F_{i+1}^0 must have a common edge E^0, the mapping of which on the original mesh should cross E with a point P_i. Thus, E is recorded to have cross points $P_1, P_2, \ldots, P_{n-1}$.The locations of these points can be computed as follows: if n=2, we flatten the faces of the base domain out using a 'hinge' map at their common edge and calculate the point position in this flattened domain; if n>2, we simply divide the edge E equally with points $P_1, P_2, \ldots, P_{n-1}$.

Step 2: Let F denote a face of the original mesh, in which E_i, E_j, E_k are three edges and V_l, V_m, V_n three vertices. All the faces crossed by E_i, E_j, E_k are denoted by S. For $F^0 \in S$, we construct the intersection set with the points recorded on E_i, E_j, E_k and V_l, V_m, V_n.

For example, in Fig.6, the intersection sets are $V_1V_mV_2$, $V_1V_2V_3V_4V_l$ and $V_4V_3V_n$. V_1, V_2, V_3 and V_4 are the points calculated in Step 1.

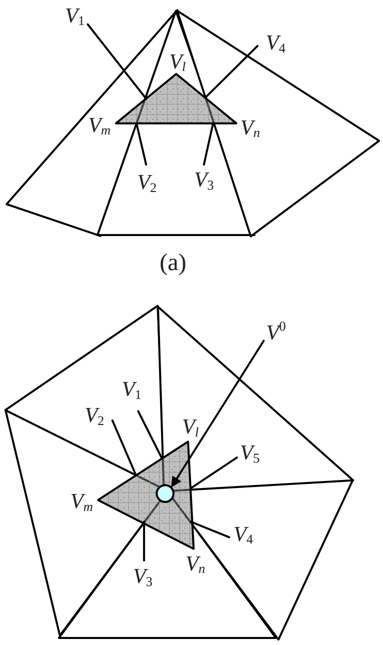


Fig.6 Mapping of new vertices to the original mesh
(a) Intersection sets; (b) New vertex mapped in the face of the original mesh

Step 3: We compute the area of these intersection sets and compare their sum with the area of triangle F. If the sum is greater than the area of F, then no new vertex has been mapped in F. Otherwise, S has a common vertex, which has been mapped in triangle F. In Fig.6a, $Area(V_1V_mV_2)+Area(V_1V_2V_3V_4V_l)+Area(V_4V_3V_n)=Area(F)$, and thus no vertex of the base mesh is mapped in F. In Fig.6b, $Area(V_lV_1V_5)+Area(V_2V_mV_3)+Area(V_3V_nV_4)=Area(F)$. The common vertex V^0 of S is mapped in $F=(V_l, V_m, V_n)$. The position of V^0's mapping in F is set the same as the barycenter of mapping points calculated in Step 1, e.g., V_1, V_2, V_3, V_4 and V_5 in Fig.6b.

Preserve of feature lines

Features of the original mesh can either be detected automatically or specified by the user. Considering only paths of edges in our parameterization, we describe the algorithm to guarantee that a certain path of edges on the original mesh gets mapped to an edge of the base domain.

We must classify the edges of $R_e(E_i)$ so as to process feature edges accordingly (see Fig.7 for the classification).

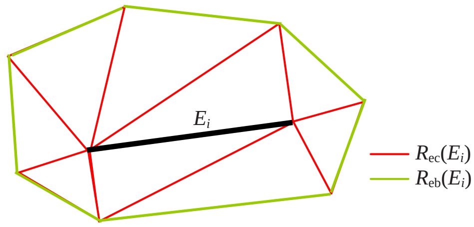


Fig.7 Classification of $R_e(E_i)$. Obviously $R_e(E_i)=R_{eb}(E_i)\cup R_{ec}(E_i)$

The feature lines will cross the $R_e(E_i)$ in the following ways:

(1) The feature lines pass only the edges of $R_{eb}(E_i)$. In such a case, the parameterization in the previous section needs no modification.

(2) The feature lines cross only an end of edge E_i (Fig.8a). We half-collapse the edge E_i into the end V_1.

(3) The feature lines pass the edge E_i. We still adopt the form of half-edge-collapse, and either end of E_i can be chosen as the collapsed vertex, as shown in Fig.8b.

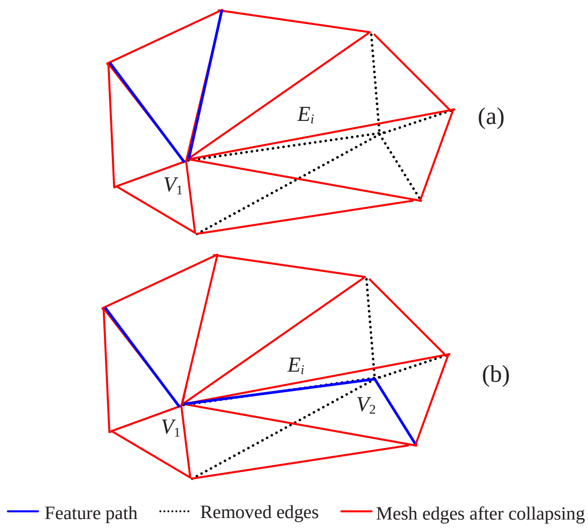


Fig.8 Half-edge collapse to preserve features
(a) Feature lines cross only an end of E_i; (b) Feature lines cross the edge E_i

ANALYSIS AND EXPERIMENTAL RESULTS

Algorithm analysis

GSP (Khodakovsky *et al.*, 2003) rates the triangulations to guide the simplification algorithm. By using edge collapse in our algorithm, we make a simple algorithm analysis of complexity.

Let the number of vertices in mesh be denoted by v and the number of edges in mesh by e. Statistically, e=3v. For a mesh vertex V_i, if the number of vertices in $N(V_i)$ are n+2, possible triangulations for $N(V_i)$ are $C(n)=\mathrm{C}_{2n}^{n}/(n+1)$, known to be the Catalan series. Note that every triangulation has n triangles to be rated. Suppose that the computation of GSP to evaluate a triangle is 1, and then the time complexity of GSP for a level of simplification is $O(nvC(n))$. Also, suppose that the computation of GSP to evaluate an edge is 1, and then our algorithm has $O(e)=O(3v)$ complexity. If we choose n=4 on average, GSP has an $O(56v)$ complexity. Note that GSP has an order of magnitude over our algorithm and Catalan series increases exponentially with n. Take n=13 for example (We encountered in the experiments the situation where an even deeper level of simplification has been carried out). Only the triangulations to be rated are $C(13)=\mathrm{C}_{26}^{13}/(13+1)=742\,900$. However, the edge-collapse algorithm will not increase computation when the degree of mesh vertices increases. Instead, complexity of our algorithm is always proportional to e. The results in the next subsection will show that the smoothness of our parameterization is also satisfying.

Experimental results

Our algorithms were implemented with VC++ and OpenGL. Fig.9 gives the mapping results of parameterization and base domains for the mannequin and horse models. We can find that the base domains have satisfying 'triangle quality'. Fig.10 shows the mapping results of the fandisk model that has features to be preserved. We tagged the feature edge whose dihedral angle was below a certain threshold. Though the simple detection algorithm in our implementation did not detect all the feature lines, our parameterization preserved feature lines during simplification.

Fig.11 shows the results of uniform and loop smooth remeshing (Lee *et al.*, 1998) of the torus model. Fig.12 gives the loop smooth remeshing of the duck model. The smoothness of parameterization obviously benefits from the base domains.

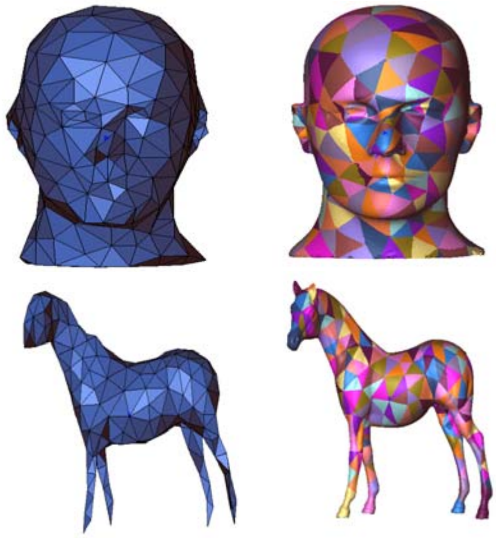
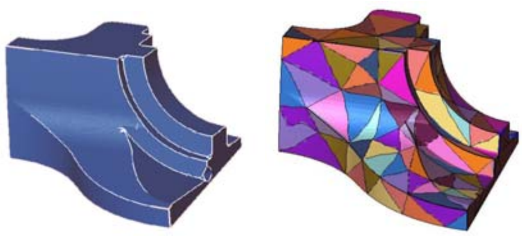
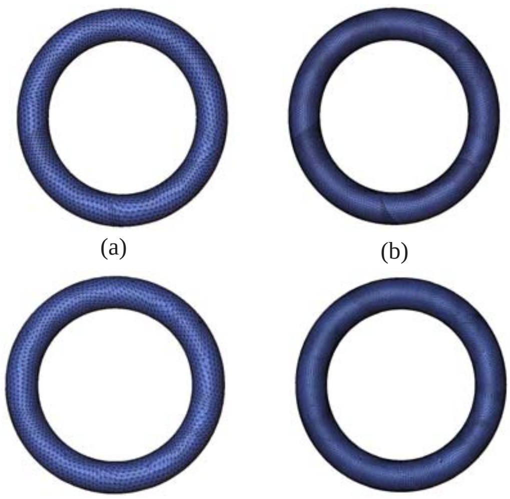


Fig.9 The mannequin (top) and horse (bottom) model's base domains (left) and parameterization (right) results



(a) (b)

Fig.10 Parameterization results of the fandisk model
(a) Feature lines of the fandisk model; (b) Parameterization results



(a) (b) (c) (d)

Fig.11 Remeshing results of the torus model
(a) Three times of 1:4 subdivision; (b) Four times of 1:4 subdivision; (c), (d) are the corresponding results of (a), (b) after a loop smooth, respectively

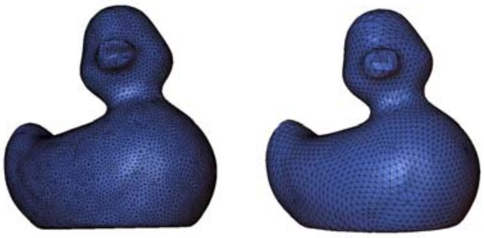

(a) (b)

Fig.12 Loop smooth remeshing of the duck model
(a) Original meshing of the duck model; (b) Remeshing of the duck model

CONCLUSION

This paper proposes a mesh parameterization based on edge collapse to construct the base domain. After obtaining the initial mapping results, we describe a direct edge classification algorithm. The intersection sets between faces of the base domain and faces of the original mesh are extracted according to the results of edge classification. The area sum of these intersection sets is compared with the area of the base triangle. The discrepancy is then utilized to map the new vertices into the base triangle and thus the parameterization is completed.

New algorithms can make use of the research findings of edge-collapse simplification to build the base domain with triangles that have a superior 'triangle quality'. The metric distortion that closely relates to the smoothness of parameterization can surely benefit from our algorithms. Moreover, the parameterization relieves heavy computation in rating the possible triangulation for optimizing the patch layout.

Further work includes collapsing the mesh edges in a non-hierarchical way and locating the mappings of the base domain vertices in a more precise way.

References

Cohen, J.D., 1999. Concepts and Algorithms for Polygon Simplification. Proc. ACM SIGGRAPH, Course Notes, p.C1-C34.

Duchamp, T., Certain, A., Derose, T., Stuetzle, W., 1997. Hierarchical Computation of PL Harmonic Embeddings. Technical Report, University of Washington, Washington, D.C., USA.

Garland, M., Heckbert, P., 1997. Surface Simplification Using Quadric Error Metrics. Proc. ACM. SIGGRAPH, p.209-216. [doi:10.1145/258734.258849]

Guskov, I., Vidimce, K., Sweldens, W., Schroder, P., 2000. Normal Meshes. Proc. ACM SIGGRAPH, p.95-102. [doi:10.1145/344779.344831]

Heckbert, P., Garland, M., 1997. Survey of Polygonal Surface Simplification Algorithms. Proc. ACM SIGGRAPH, Multiresolution Surface Modeling Course Notes, p.1-31.

Khodakovsky, A., Schroder, P., Sweldens, W., 2000. Progressive Geometry Compression. Proc. ACM SIGGRAPH, p.271-278. [doi:10.1145/344779.344922]

Khodakovsky, A., Litke, N., Schroder, P., 2003. Globally Smooth Parameterizations with Low Distortion. Proc. ACM SIGGRAPH, p.350-357. [doi:10.1145/201775.882275]

Lee, A., Sweldens, W., Schroder, P., Cowsar, L., Dobkin, D., 1998. MAPS: Multiresolution Adaptive Parameterization of Surfaces. Proc. ACM SIGGRAPH, p.95-104. [doi:10.1145/280814.280828]

Lee, A., Dobkin, D., Sweldens, W., Schroder, P., 1999. Multiresolution Mesh Morphing. Proc. ACM SIGGRAPH, p.343-350. [doi:10.1145/311535.311586]

Levy, B., Petitjean, S., Ray, N., Maillot, J., 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, **21**(3):362-371. [doi:10.1145/566654.566590]

Luebke, D., 2001. A developer's survey of polygonal simplification algorithms. *IEEE Comput. Graph. Appl.*, **21**(1):24-35. [doi:10.1109/38.920624]

Schroeder, W.J., Zarge, J.A., Lorensen, W.E., 1992. Decimation of Triangle Meshes. Proc. 19th Annual Conf. on Computer Graphics and Interactive Techniques, p.65-70. [doi:10.1145/133994.134010]