

Filling Holes in Triangular Meshes in Engineering

Ping Hu

School of Automotive Engineering, Faculty of Vehicle Engineering and Mechanics,
State Key Laboratory of Structural Analysis for Industrial Equipment,
Dalian University of Technology, Dalian 116024, P.R. China
Email: pinghu@dlut.edu.cn

Changsheng Wang, Baojun Li, and Mingzeng Liu

School of Automotive Engineering, Faculty of Vehicle Engineering and Mechanics,
State Key Laboratory of Structural Analysis for Industrial Equipment,
Dalian University of Technology, Dalian 116024, P.R. China
Email: {wangcshe, labeljazz, mzliu}@gmail.com

Abstract—In this paper, a novel hole-filling algorithm for triangular meshes is proposed. Firstly, the hole is triangulated into a set of new triangles using modified principle of minimum angle. Then the initial patching mesh is refined according to the density of vertices on boundary edges. Finally, the patching mesh is optimized via the bilateral filter to recover missed features. Experimental results demonstrate that the proposed algorithm fills complex holes robustly, and preserves geometric features to a certain extent as well. The resulted meshes are of good quality for engineering.

Index Terms—hole-filling, triangulation, refinement, bilateral filter

I. INTRODUCTION

Polygonal representations of 3D objects, and particular triangular meshes, have become prevalent in numerous application domains. As 3D optical scanners become widespread, triangle meshes can be easier created and widely applied in the fields of CAD and reverse engineering. Even with high-fidelity scanners, the data obtained is often incomplete. The existence of holes makes it difficult for many operations based on meshes, such as model rebuilding, rapid prototyping and finite element analysis. In the process of One-Step inverse forming finite element analysis, the existence of holes may affect the robustness of the analysis and generate uncertain results. An illustration is shown in Fig. 1. Fig. 1(a) shows the initial mesh of beetle, Fig. 1(b) the result of unfolding the beetle directly via KMAS/One-Step and Fig. 1(c) the result of unfolding after inner holes filled. The hole-filling algorithm also can be used in other important applications, such as mesh parametrization [1], feature suppression [2], and meshes merging [3].

Surface feature refers to crest lines which are formed by points where the largest in absolute value principal curvature takes a positive maximum (negative minimum) along its corresponding curvature line [4]. It is a powerful shape descriptor and important for the appearance and expression of geometric models. Feature preserved hole-filling is significant in applications, which can effectively restore the initial data and improve the precision of finite element analysis. A feature preserved hole-filling algorithm is discussed in this paper.

Various mesh hole-filling approaches have been proposed in recent years. These approaches can be mainly classified into two categories [5]: volume-based and triangle-based. The volume-based approaches indirectly repair the model using an intermediate volumetric grid, while the triangle-based approaches identify and fill the holes directly on the model.

Volume-based approaches: The input model is first converted into an intermediate volumetric grid, where each grid point is associated with a positive or negative sign indicating it is inside or outside the model. Next, a polygonal surface is reconstructed that separates the grid points of different signs. Davis et al. [6] apply a volumetric diffusion process to extend a signed distance function through this volumetric representation until its zero set bridges whatever holes may be present. The diffusion method is further extended by Guo et al. [7] using oriented voxel diffusion method. Nooruddin et al. [8] develop a similar approach for the simplification and the repairing of polygonal meshes. Ju [9] traces and patches hole boundaries on a dual surface of the primal grid, which yields an inside/outside partitioning on the primal mesh. Roughly speaking, volume-based methods excel in their robustness in resolving complex holes, however, they are time consuming and the generated mesh may be incorrect in some cases.

Mesh-based approaches: The holes are explicitly searched and filled directly on triangle mesh. Liepa [10] first fills the hole with a minimum area triangulation of its 3D contour [11], then the triangulation is refined so that the triangle density agrees with the density of the

Manuscript received January 1, 2011; revised June 1, 2011; accepted July 1, 2011.

This work was funded by the Key Project of the NSFC (No. 10932003, u0935004), NSFC (No. 60873181), “863” Project of China (No. 2009AA04Z101), “973” National Basic Research Project of China (No. 2010CB832700) and the Fundamental Research Funds for the Central Universities.

Corresponding author: Ping Hu, pinghu@dlut.edu.cn.

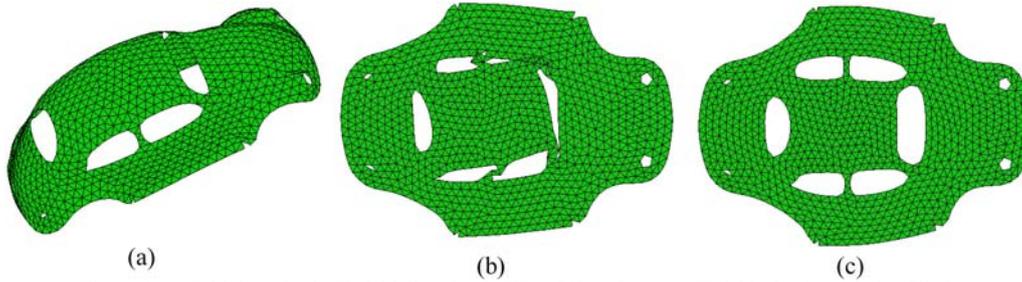


Figure 1. (a) Initial mesh; (b) Unfold directly via KMAS/One-Step; (c) Unfold after inner holes filled.

surrounding mesh triangles [12], at last a fairing technique based on an umbrella operator [13] is used to smooth the filled hole. Brunton et al. [14] first unfold hole boundary onto a plane and triangulate it, then embed the triangular mesh as a minimum energy surface in R^3 . Zhao et al. [15] optimize the vertex position by solving Poisson equation after getting the initial patch mesh by advancing front method. A piecewise hole-filling algorithm is proposed by Jun [16], it splits the complex hole into several simple holes and then fills them with planar triangulation method. Li et al. [17] adopt the same strategy based on the concept of edge expansion.

Several methods consider hole-filling with feature preserved recently. Chen et al. [18] propose a hole-filling method which can fill the hole and recover its sharp feature involved in the hole area. With this method, holes are filled using a radial basis function, a feature enhancement process based on sharpness dependent filter [19] is then applied if there exists any sharp feature on the hole boundary. Li et al. [20] present a hole-filling algorithm by using polynomial blending technique, in which a new curve “fairness” measure of the blending curves is introduced and some desirable results with feature preserved are obtained. The main drawbacks of both methods are: several parameters are chosen by a subjective judgment and for the mesh model of inconspicuous features, the result cannot be used in engineering.

In this paper, a new approach of hole-filling is proposed which can preserve geometric features to a certain extent. Moreover, the hole-filling mesh models are of excellent quality for engineering. We first triangulate the hole using modified principle of minimum angle, then the triangulation is refined according to the density of vertices on boundary edges, finally the refined mesh is optimized via the bilateral filter [21, 22] to recover the missed features.

The rest of the paper is organized as follows: some preliminaries are given in section II; section III describes the detail of the algorithm; in section IV, experimental results are discussed; this paper is concluded in section V.

II. PRELIMINARIES

A triangle mesh is defined by a set of oriented triangles joining a set of vertices. Two triangles are adjacent if they share a common edge. Adjacent triangles are consistently oriented if their common edge is of opposite directions. A mesh is oriented if all its triangles are consistently

oriented. An edge is adjacent to a triangle if it is a part of the triangle.

In contrary to an interior edge, a boundary edge is an edge adjacent to exactly one triangle. A boundary vertex is a vertex that is adjacent to a boundary edge. A hole is a closed cycle of boundary edges. A singular vertex is a vertex that has more than two adjacent boundary edges. A non-manifold edge is an edge that has more than two adjacent triangles. A manifold mesh has no non-manifold edges and no singular vertices, but may have boundaries. Holes can be identified automatically by looking for boundary vertices. Given a seed boundary vertex, a series of boundary edges can be traced until a closed loop of boundary edges is identified. If the orientation of one edge of the loop coincides with the direction of the edge in the triangle, the hole is anticlockwise; otherwise, the hole is clockwise.

In fact, if two separate holes have vertices in common, the vertices would be singular, and if a given hole has islands, the mesh would not be connected. In this work, we just consider the simple case. That is, we assume that all meshes are triangular, oriented, manifold, connected, and that two separate holes have no vertices in common, and a given hole do not have islands. The original mesh and the mesh that fills the hole are, respectively, called the surrounding mesh and the patching mesh.

III. HOLE-FILLING ALGORITHM

A. Algorithm overview

In order to make hole-filling process simple and robust, and the resulted meshes are of good quality, we propose a novel algorithm. The algorithm is summarized and illustrated in Fig. 2. The key ideas consist of three aspects. First, the modified principle of minimum angle is adopted to generate the initial patching mesh. Second, the patching mesh is refined to enhance its quality. Third, the patching mesh is optimized via the bilateral filter to recover missed features.

B. Triangulation

After hole identification, the first step in filling it is to triangulate it with modified principle of minimum angle. In order to calculate the angles between two adjacent boundary edges at each vertex on the loop, we should decide the convexity/concavity of boundary vertices. Convexity/concavity is an important geometric property of the mesh vertices, Vivodtzev et al. [23] use the mean curvature $K_{dir}(\mathbf{v}_i)$ and the normal \mathbf{n}_i of vertex \mathbf{v}_i to

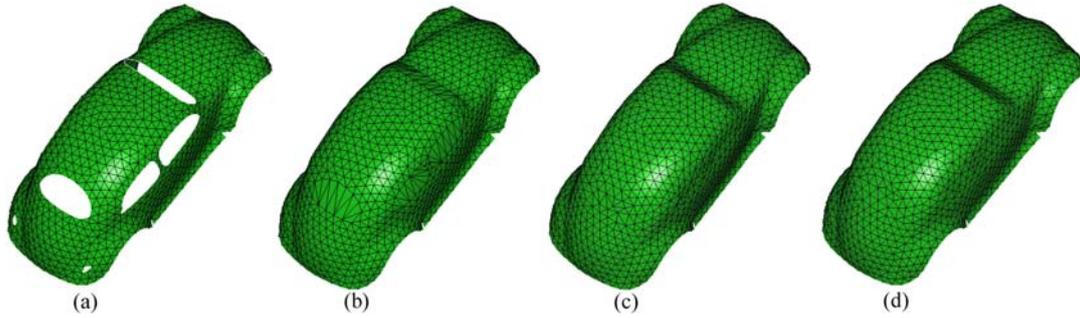


Figure 2. The process of the hole-filling algorithm. (a) The beetle model with holes; (b) The result after hole triangulation via modified principle of minimum angle; (c) The result after patching meshes refinement; (d) The result after patching meshes optimization.

distinguish the convexity/concavity as follows:

$$Bool(\mathbf{v}_i) = \begin{cases} convex & \text{if } K_{dir}(\mathbf{v}_i) \cdot \mathbf{n}_i \leq 0 \\ concave & \text{if } K_{dir}(\mathbf{v}_i) \cdot \mathbf{n}_i > 0 \end{cases} \quad (1)$$

We do not want to use above formula to determine the convexity/concavity, since the mean curvature $K_{dir}(\mathbf{v}_i)$ involves complex computations. In fact, it is much easier to obtain the convexity/concavity of a boundary vertex, if we can get the normal of the boundary vertex and the normal of the oriented triangle constituted of the boundary vertex and its two adjacent vertices.

Denote LV as the collection of ordered vertices on the hole boundary, as illustrated in Fig. 3, for each vertex \mathbf{v}_i in LV , denote \mathbf{n}_i^t as the normal of the oriented triangle $(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1})$, \mathbf{n}_i^v the normal of vertex \mathbf{v}_i , θ_i the acute angle formed by edges e_i and e_{i+1} , we get the follow result: if $\mathbf{n}_i^t \cdot \mathbf{n}_i^v > 0$ and the hole is anticlockwise, or $\mathbf{n}_i^t \cdot \mathbf{n}_i^v < 0$ and the hole is clockwise, the vertex \mathbf{v}_i is concave; otherwise, the vertex \mathbf{v}_i is convex. Therefore, the angle value is assigned to each boundary vertex:

$$angle(\mathbf{v}_i) = \begin{cases} 2\pi - \theta_i & \text{if concave} \\ \theta_i & \text{if convex} \end{cases} \quad (2)$$

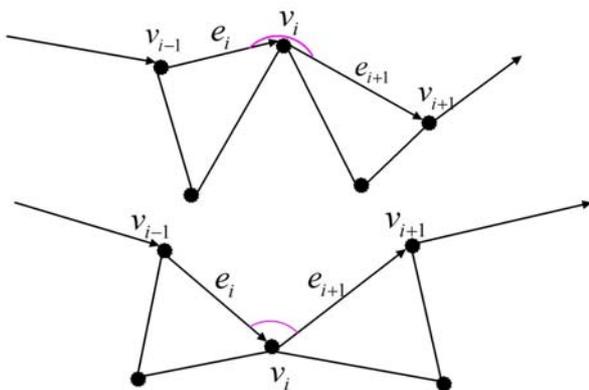


Figure 3. Judgement for convex/concave vertices.

Denote NT the collection of new triangles to be added to the hole and $sum(LV)$ the number of vertices in LV . Given an angle threshold σ , main steps of triangulation are as follows:

Step 1: If $sum(LV) > 3$, go to Step 2, otherwise go to Step 3.

Step 2: Search for the vertex \mathbf{v} with minimum angle in LV , denote α as the angle constituting of two normals of its adjacent vertices in LV . If $\alpha > \sigma$, search for the vertex with minimum angle except \mathbf{v} and calculate the angle α until $\alpha \leq \sigma$. Denote the vertex as \mathbf{v} and its two adjacent vertices in LV as $\mathbf{v}_1, \mathbf{v}_2$. Then add the oriented triangle $(\mathbf{v}_1, \mathbf{v}, \mathbf{v}_2)$ to NT and delete vertex \mathbf{v} from LV .

Step 3: Add the oriented triangle to NT .

The threshold σ is associated with the maximum angle of two normals constituting of two arbitrary vertices on the hole boundary. $\sigma = 45^\circ$ when the maximum angle is less than 55° , else σ be 85% of the maximum angle.

C. Refinement

In order to fill a hole with a mesh that approximates the density of the surrounding mesh, we refine the patching mesh by the algorithm given by Pfeifle and Seidel [12]. The basic idea is to compute edge length data for the vertices on the hole boundary and diffuse these values into the interior of the patching mesh by subdividing triangles to reduce edge lengths, and relaxing interior edges to maintain a Delaunay-like triangulation.

Given a density control factor α and the patching mesh, main steps of refinement are as follows:

Step 1: For each vertex \mathbf{v}_i on the hole boundary, compute the scale attribute $\sigma(\mathbf{v}_i)$ as the average length of the edges that are adjacent to \mathbf{v}_i in the surrounding mesh. Initialize the patching mesh as the given hole triangulation.

Step 2: For each triangle $(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k)$ in the patching mesh, compute the centroid \mathbf{v}_c and the corresponding scale attribute $\sigma(\mathbf{v}_c) = (\sigma(\mathbf{v}_i) + \sigma(\mathbf{v}_j) + \sigma(\mathbf{v}_k)) / 3$. For $m = i, j, k$, if $\alpha \| \mathbf{v}_c - \mathbf{v}_m \| > \sigma(\mathbf{v}_c)$,

$\alpha \| \mathbf{v}_c - \mathbf{v}_m \| > \sigma(\mathbf{v}_m)$ and the maximum inner angle of triangle $(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k)$ is less than $5\pi/6$, then replace the triangle $(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k)$ with three triangles: $(\mathbf{v}_c, \mathbf{v}_j, \mathbf{v}_k)$, $(\mathbf{v}_i, \mathbf{v}_c, \mathbf{v}_k)$ and $(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_c)$ in the patching mesh, and then relax the edges $(\mathbf{v}_i, \mathbf{v}_j)$, $(\mathbf{v}_i, \mathbf{v}_k)$ and $(\mathbf{v}_j, \mathbf{v}_k)$ (Fig. 4).

Step 3: If no new triangles were created in Step 2, the patching mesh is complete.

Step 4: Relax all interior edges of the patching mesh.

Step 5: If no edges were swapped in Step 4, go to Step 2, otherwise go to Step 4.

Generally, density control factor $\alpha = \sqrt{2}$ is a good choice to yield a patching mesh that visually matches the density of the surrounding mesh. To relax an edge means, for the two triangles adjacent to the edge, check whether each of the two non-mutual vertices of these triangles lies outside of the circum-sphere of the opposing triangle (see Fig. 4). If this test fails, the edge is swapped.

D. Optimization

As noted in the introduction, the mesh optimization process for the refined patching mesh can be regarded as bilateral filter [16, 17]. The bilateral filter is an anisotropic mesh denoising algorithm that is effective, simple and fast, which filter vertices of the mesh in the normal direction using local neighborhoods, and it can remove noise from meshes while preserving features. The main idea is to define a local parameter space for every vertex using the tangent plane to the mesh at a vertex.

Let S denote the noise-free surface, and M be the input mesh with vertices that sample S with additive noise. Let $\mathbf{v} \in M$ be a vertex of the input mesh, d_0 its signed-distance to S , and \mathbf{n}_0 the normal to S at the closest point to \mathbf{v} . The noise-free surface S is unknown and so is d_0 , the bilateral filter estimate the normal to the

surface as the normal \mathbf{n} to the mesh, and d estimate d_0 as the application of the filter, updating \mathbf{v} as follows:

$$\hat{\mathbf{v}} = \mathbf{v} + d \cdot \mathbf{n} \tag{3}$$

The bilateral filter follows the formulation of the bilateral filtering for image $I(\mathbf{u})$ at coordinate $\mathbf{u} = (x, y)$:

$$\hat{I}(\mathbf{u}) = \frac{\sum_{\mathbf{p} \in N(\mathbf{u})} W_c(\|\mathbf{p} - \mathbf{u}\|) W_s(|I(\mathbf{u}) - I(\mathbf{p})|) I(\mathbf{p})}{\sum_{\mathbf{p} \in N(\mathbf{u})} W_c(\|\mathbf{p} - \mathbf{u}\|) W_s(|I(\mathbf{u}) - I(\mathbf{p})|)} \tag{4}$$

Where $N(\mathbf{u})$ is the neighborhood of \mathbf{u} . The closeness smoothing filter is the standard Gaussian filter with parameter σ_c : $W_c(x) = e^{-x^2/2\sigma_c^2}$, and a feature-preserving weight function $W_s(x) = e^{-x^2/2\sigma_s^2}$. In practice, $N(\mathbf{u})$ is defined by the set of points $\{\mathbf{q}_i\}$, where $\|\mathbf{u} - \mathbf{q}_i\| < \rho = 2|\sigma_c|$.

Following (4), the pseudo-code for applying a bilateral filter to a single vertex \mathbf{v} is given in the following:

```

{q_i} = neighborhood(v)
K = |{q_i}|
sum = 0
normalizer = 0
for i = 1 .. K
    t = ||v - q_i||
    h = <n, v - q_i>
    w_c = exp(-t^2 / (2*sigma_c^2))
    w_s = exp(-h^2 / (2*sigma_s^2))
    sum += (w_c * w_s) * h
    normalizer += w_c * w_s
end
Vertex v-hat = v + n * (sum / normalizer)
    
```

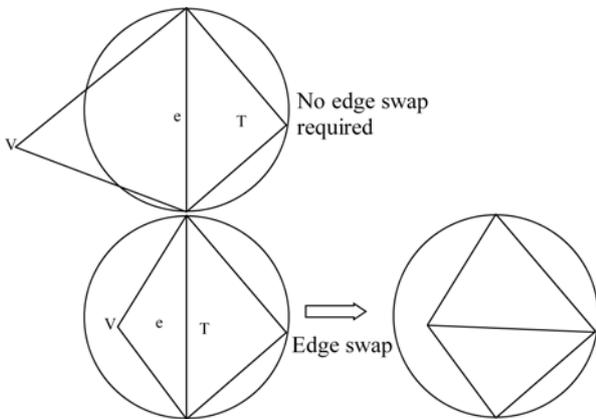


Figure 4. Edge relaxation.

Where $\{\mathbf{q}_i\}$ is the 1-ring neighborhood of \mathbf{v} and K is the number of $\{\mathbf{q}_i\}$. The plane that approximates the noise-free surface should be a good approximation of the local surface and preserve sharp features. The first requirement leads to smoothing the surface, while the latter maintains the noisy surface. Generally, we compute the normal at a vertex as the weighted average (by the area of the triangles) of the normal to the triangles in the 1-ring neighborhood of the vertex.

The parameters of the bilateral filter are σ_c , σ_s , the kernel size ρ and the number of iterations, which can be set by user-assisted method. Two parameters, σ_c and σ_s are interactively assigned, while the radius of the neighborhood is assigned to σ_c , and we set $\rho = 2\sigma_c$. The σ_s is set to the standard deviation of the offsets in

the selected neighborhood. One may choose a large σ_c and perform a few iterations, or choose a narrow filter and increase the number of iterations. Multiple iterations with a narrow filter have the effect of applying a wider filter, and result in efficient computation.

IV. IMPLEMENTATIONS AND RESULTS

This section illustrates how our algorithm is implemented on computer and demonstrates the experiment results. The algorithm is implemented in VC++6.0 and OpenGL.

In all the results shown in the paper, we set σ_c and σ_s be 0.1 and 0.5 of the current vertex density respectively, the number of iterations of bilateral filter is three.

Several mesh models are used to test the proposed hole-filling algorithm. Fig. 5(a) is a mesh model of car body external panel. To illustrate the power of feature preservation, a hole has been created in the blending area (Fig. 5(b)). Fig. 5(c) is the result of Liepa [10], and Fig. 5(d) is the result of our hole-filling algorithm. Fig. 6 also presents the complex hole-filing results by Liepa and ours. Fig. 6(a) is the initial mesh model, while Fig. 6(b) and Fig. 6(c) are the results of Liepa and our algorithm respectively. The comparison of model a and b in Fig. 5 and Fig. 6 between the method of Liepa and our method illustrates that our algorithm can preserve geometric features to a certain extent, while the method of Liepa produces irregularly shaped excavations in the blending areas.

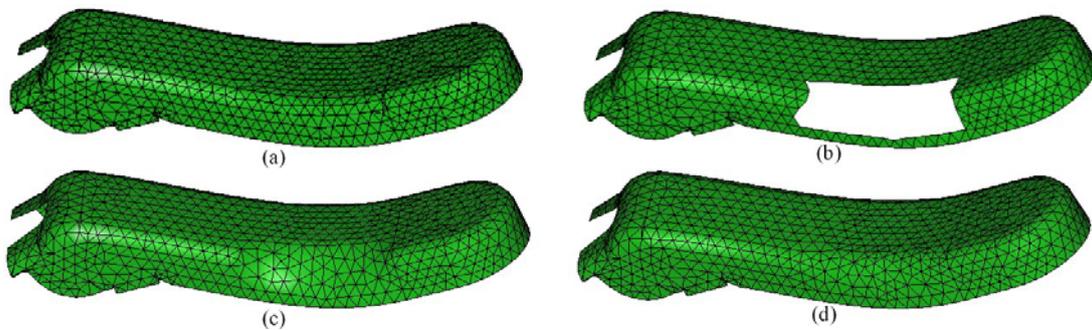


Figure. 5 Patch results of model a. (a) The initial mesh model; (b) A hole is created in a blending area; (c) The result of Liepa [10]; (d) The result of our method.

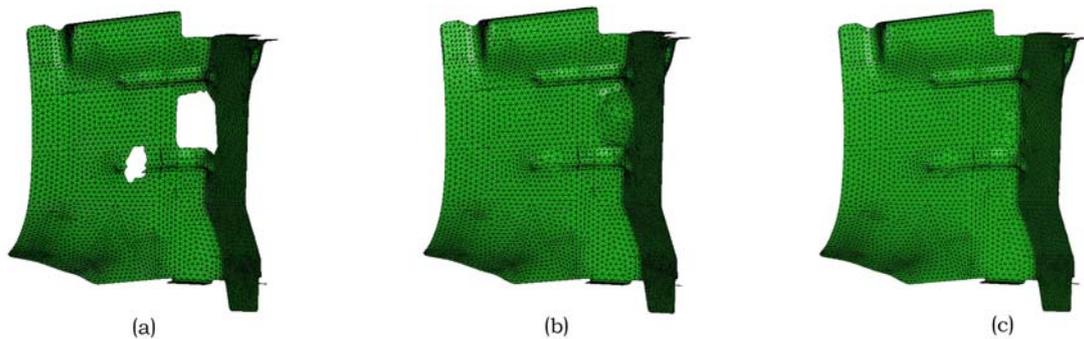


Figure 6. Patch results of model b. (a) The initial mesh model; (b) The result of Liepa [10]; (c) The result of our method.

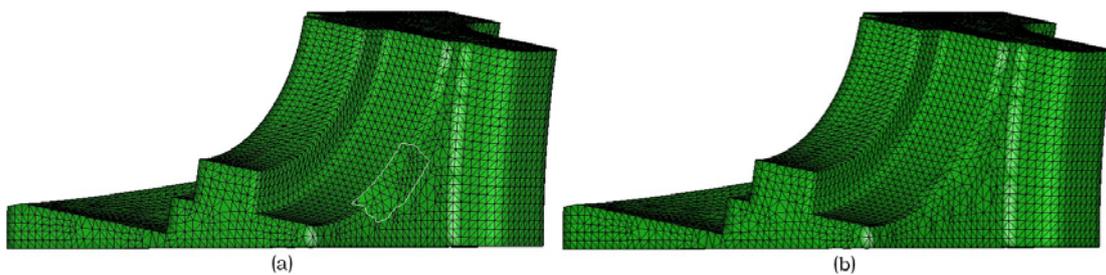


Figure 7. Patch result of model c. (a) The initial mesh model; (b) The result of our method.

Another three examples are given. Fig. 7 shows the patch result of model c (Findisk), while Fig. 8 and Fig. 9 show the patch results of model d and e. Quality statistics of patching meshes of the models are given in Fig. 10 and table 1. In Fig. 10, the horizontal axis denotes the quality of triangles, while the vertical axis denotes the number of triangles. We give some other statistics of models in table 1, including the number of new created vertices, the number of new created triangles, the minimum, maximum, and mean quality of the patching meshes. We adopt the method of Field [24] to assess the quality of triangle by the ratio of twice the radius of inscribed circle to the radius of circum circle of a triangle. Generally, the triangle is of good quality if the ratio is not less than 0.7, and the closer to 1, the better of the triangle. Fig. 9 and table 1 illustrates the patching meshes are of good quality, which can well approximate the density of surrounding

mesh and can be used in engineering, while the algorithm can preserve geometric features to a certain extent as well.

V.CONCLUSION

In this paper, a novel hole-filling algorithm is proposed. The method first triangulates the hole into a set of new triangles, and then refines the patching meshes according to the density of surrounding vertices, and finally the patching meshes are optimized via the bilateral filter. The proposed method is simple, stable and robust. The patching meshes are of good quality for numerical computation and other applications, such as finite element analysis. Modified principal of minimum angle and bilateral filter is crucial for feature preservation. Experiments illustrate the method can preserve features to a certain extent.

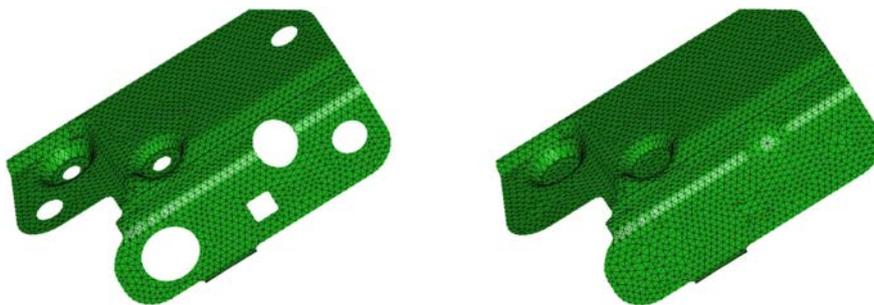


Figure 8. Patch result of model d.

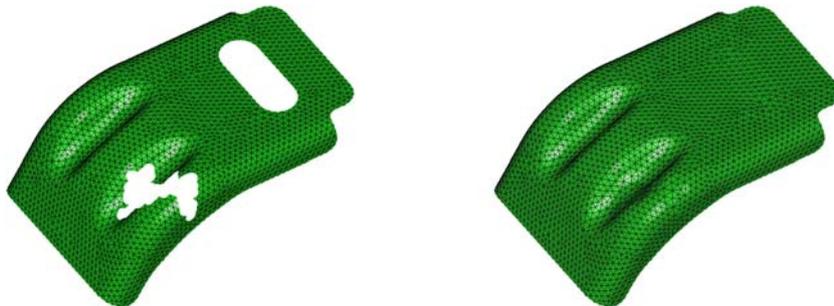


Figure 9. Patch result of model e.

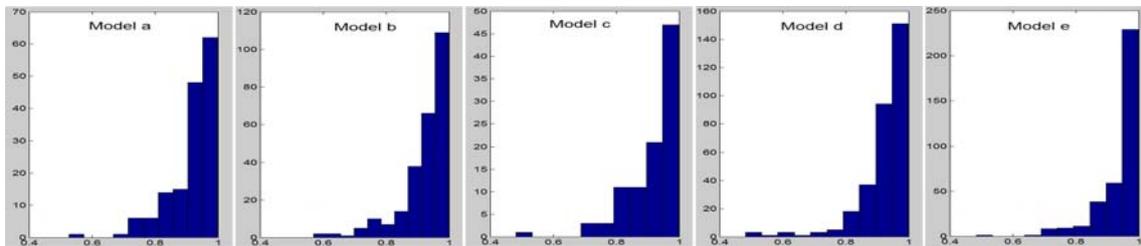


Figure 10. Quality Statistics of patching meshes.

TABEL 1. Quality statistics.

Model	New created vertices num	New created triangles num	Min quality	Max quality	Mean quality
a	59	133	0.5271	0.9999	0.9198
b	100	254	0.5700	0.9998	0.9234
c	46	104	0.4817	0.9998	0.9815
d	100	316	0.4817	1.0000	0.9248
e	131	356	0.4839	1.0000	0.9411

ACKNOWLEDGMENT

The authors wish to thank Yanli Wu, Hui Wang, Yongxia Hao, Xiaochao Wang and Yan Zhao for their help.

REFERENCES

- [1] B.J. Li, X.K. Zhang P. Zhou and P. Hu, "Mesh parameterization based on one-step inverse forming," *Comput. Aided Des.* 42(7), pp. 633-640, 2010.
- [2] S. Gao, W. Zhao, H. Lin, F. Yang and X. Chen, "Feature suppression based cad mesh model simplification," *Computer-Aided Design*, 42(12), pp. 1178-1188, 2010.
- [3] R. Lou, J.-P. Pernot, A. Mikchevitch and P. Véron, "Merging enriched finite element triangle meshes for fast prototyping of alternate solutions in the context of industrial maintenance," *Comput. Aided Des.* 42(8), pp. 670-681, 2010.
- [4] Y. Ohtake, A. Belyaev and H.-P. Seidel, "Ridge-valley lines on meshes via implicit surface fitting," *ACM Transactions on Graphics*, 23(3), pp. 609-612, 2004.
- [5] T. Ju, "Fixing geometric errors on polygonal models: a survey," *Journal of Computer Science and Technology*, 24(1), pp. 19-29, 2009.
- [6] J. Davis, S.R. Marschner, M. Garr and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," In: *First International Symposium on 3D Data Processing Visualization and Transmission*, Padova, pp. 428-438, 2002.
- [7] T.-Q. Guo, J.-J. Li, J.-G. Weng and Y.-T. Zhuang, "Filling holes in complex surfaces using oriented voxel diffusion," In: *2006 International Conference on Machine Learning and Cybernetics, IEEE*, Los Alamitos, pp. 4370-4375, 2006.
- [8] F.S. Nooruddin and G. Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Transactions on Visualization and Computer Graphics*, 9(2), pp. 191-205, 2003.
- [9] T. Ju, "Robust repair of polygonal models," *ACM Transactions on Graphics*, 23(3), pp. 888- 895, 2004.
- [10] P. Liepa, "Filling holes in meshes," In: *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. Aire-la-Ville, Switzerland: Eurographics Association, pp. 200-205, 2003.
- [11] G. Barequet and M. Sharir, "Filling gaps in the boundary of a polyhedron," *Computer Aided Geometric Design*, 12(2), pp. 207-229, 1995.
- [12] R. Pfeifle and H.-P. Seidel, "Triangular B-splines for blending and filling of polygonal holes," In: *Proceedings of Graphics Interface*, pp. 186-193, 1996.
- [13] L. Kobbelt, S. Campagna, J. Vorsatz and H.-P. Seidel, "Interactive multi-resolution modeling on arbitrary meshes," In: *Proceedings of SIGGRAPH'98*, pp. 105-114, 1998.
- [14] A. Brunton, S. Wuhler, C. Shu, P. Bose and E.D. Demaine, "Filling holes in triangular meshes by curve unfolding," In: *IEEE International Conference on Shape Modeling and Applications*, pp. 66-72, 2009.
- [15] W. Zhao, S.M. Gao and H.W. Lin, "A robust hole-filling algorithm for triangular mesh," *The Visual Computer: International Journal of Computer Graphics*, 23(12), pp. 987-997, 2007.
- [16] Y. Jun, "A piecewise hole filling algorithm in reverse engineering," *Computer-Aided Design*, 37(2), 263-270, 2005.
- [17] G. Li, X.Z. Ye and S.Y. Zhang, "An algorithm for filling complex holes in reverse engineering," *Engineering with Computers*, 24(2), pp. 119-125, 2008.
- [18] C.Y. Chen and K.Y. Cheng, "A sharpness-dependent filter for recovering sharp features in repaired 3D mesh models," *IEEE Transactions on visualization and computer graphics*, 14(1), 200-212, 2008.
- [19] C.Y. Chen and K.Y. Cheng, "A sharpness dependent filter for mesh smoothing," *Comput. Aided Geom. Des.*, 22(5), 376-391, 2005.
- [20] Z. Li, D.S. Meek and D.J. Walton, "Polynomial blending in a mesh hole-filling application," *Computer-Aided Design*, 42, pp. 340-349, 2010.
- [21] S. Fleishman, I. Drori and D. Cohen-Or, "Bilateral mesh denoising," *ACM Transactions on Graphics*, 22(3), pp. 950-953, 2003.
- [22] T.R. Jones, F. Durand and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Transactions on Graphics*, pp. 943-949, 2003.
- [23] F. Vivodtzev, L. Linsen, B. Hamann, K.I. Joy and B.A. Olshausen, "Brain mapping using topology graphs obtained by surface segmentation," In: *Scientific Visualization: The Visual Extraction of Knowledge from Data*, Springer Berlin Heidelberg, Schloss Dagstuhl, Allemagne, pp. 35-48, 2006.
- [24] D.A. Field, "Qualitative measures for initial meshes," *International Journal for Numerical Methods in Engineering*, 47, pp. 887-906, 2000.



Ping Hu, born in Fuchuan of Guangxi province, China, 1956. He obtained the bachelor's, the master's degree and the doctor's degree in mathematics mechanics and computing mechanics of the previous Jilin University of Technology In 1982, 1984 and 1993, respectively. From 1993 to 1995, he finished his postdoctoral research in Kobe University of Japan. He was specially promoted to be an associate professor, and then a professor in 1991, and 1993, respectively. His major researching fields concern Digital vehicle and advanced forming technologies. The current main research interests include:

1) Theories and methods of digital design, analysis and manufacture of auto-body, and relative commercial code developments.

2) Elastic-plastic finite element methods with large deformation; thermal & mechanical finite element analysis for hot forming, and relative numerical technique for materials forming; failure analysis and impact analysis for design.

He is a dean in Department of Automobile Engineering in Dalian University of Technology, a professor in State Key Lab of Structural Analysis of Industrial Equipment, and CEO of KingMesh Co. Ltd., in Dalian of China. He also gets the following honors: Gainers of National Outstanding Youth Fund in China, Special Allowance of State Department and Elitist Fund of Ministry of Education; One of the young science and technology experts in Chinese mechanical and automotive industries. He is the member of NUMIFORM2001 Science Committee.

Changsheng Wang, born in Huinan of Jilin province, China, 1984. He is a PhD candidate in School of Automotive Engineering of Dalian University of Technology. His research interests include computer-aided design and computer graphics, integration of CAD/CAE.

Baojun Li, born in Laiyang of Shandong province, China, 1977. He received the Phd degree in computer-aided geometric design from Dalian University of Technology, China in 2009. He is currently a postdoctoral fellow in the School of Automotive

Engineering, Dalian University of Technology. His research interests include computer-aided design, computer graphics and integration of CAD/CAE. Dr. Li has the membership of ACM and CCF.

Mingzeng Liu, born in Huaiyang of Henan province, China, 1983. He is a PhD candidate in School of Automotive Engineering of Dalian University of Technology. His research interests include computer-aided design, computer graphics and integration of CAD/CAE.