Visual Comput (2007)
DOI 10.1007/s00371-007-0167-y

ORIGINAL ARTICLE

Wei Zhao
Shuming Gao
Hongwei Lin

A robust hole-filling algorithm for triangular mesh

W. Zhao · S. Gao (✉) · H. Lin
The State Key Lab of CAD&CG, Zhejiang University, Hang Zhou, China
{zhaowei, smgao, hwlin}@cad.zju.edu.cn

Abstract This paper presents a novel hole-filling algorithm that can fill arbitrary holes in triangular mesh models. First, the advancing front mesh technique is used to cover the hole with newly created triangles. Next, the desirable normals of the new triangles are approximated using our desirable normal computing schemes. Finally, the three coordinates of every new vertex are re-positioned by solving the Poisson equation based on the desirable normals and the boundary vertices of the hole. Many experimental results and error evaluations are given to show the robustness and efficiency of the algorithm.

Keywords Hole-filling · Poisson equation · Harmonic Function · Triangular mesh

1 Introduction

Triangular meshes are used widely to represent an object in 3D modeling systems and computer graphics applications. And they can be obtained in various ways such as a 3D scanner and computer-aided design software. However, due to the limitations of the generation methods, the resulting triangular mesh models cannot be utilized directly by other applications often because of their incompletion, i.e., containing some defects, such as holes, self-intersecting triangles, gaps, etc. Therefore, certain repairs must be done before taking these models into actual applications, and hole-filling is an important one among them. Another important application of the hole-filling algorithm is in the area of feature suppression. In many computer-aided engineering applications, detailed geometric features are not necessary, so that the triangles involved in the detailed features need to be deleted and the resulting hole must be filled.

Many hole-filling approaches have been proposed in the literature. These approaches can be divided into two categories: voxel-based and triangle-based. In the voxel-based approaches, a mesh model is first converted into a volumetric representation which consists of discrete volumes named voxels, and then different methods are utilized to patch up the holes in volumetric space. Davis et al. [7] used volumetric diffusion to fill the gaps and Curless et al. [6] employed space carving and iso-surface extraction to fill holes. Ju [12] constructed an inside/outside volume using an octree grid and re-constructed the surface by contouring. Joshua and Szymon [14] used a min-cut algorithm to split space into inside and outside portions, and patched the holes simultaneously in a globally sensitive manner. Voxel-based approaches work well for complex holes but they are all time-consuming and may generate incorrect topology in some cases.

In the triangle-based approaches, the holes are patched by dealing with the triangles directly. Holes with regular boundary over a relatively planar region can be easily patched via planar triangulation, which has been described in detail by a number of textbooks and papers [8, 11, 17]. However, filling a complex hole over an irregular region is much more difficult. To solve this problem, Carr et al. [1] used radial basis function to construct an implicit surface to cover the hole. This method works well for convex surfaces and can handle irregular holes. But difficulties arise when the underlying surface is too complex to be described by a single-value function. Liepa et al. [15] presented an umbrella operator to fair the triangulation over the hole to estimate the underlying geometry. However, the $O(n^3)$ performance of the triangulation method limits this method from being used widely. Jun [13] proposed a hole-filling method based on a piecewise scheme. His method divides a complex hole into several simple holes and all sub-holes are sequentially filled with planar triangulation; sub-division and refinement are then employed to smooth the new triangles. The negative side of the method is that too many overlaps or twists may make it crash and iterative refinement is a time-consuming process. Chen et al. [2] proposed 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 Bayesian classification [3] and sharpness dependent filter [4] is then applied if there exists any sharp feature on the hole boundary.

Some hole-filling algorithms for parametric surfaces have been presented [5, 16, 19]. Since the boundaries of the holes handled are usually made up of a B-spline curve, conserving continuity is more important for these hole-filling algorithms.

Ideally, hole-filling algorithm should posses the following properties: (1) able to cover an arbitrary hole for any model (robustness), (2) capable of filling large holes in a reasonable amount of time (efficiency), (3) enable the patched surface to match the missing geometry well (precision).

Unfortunately, due to the complexity and diversity of the holes, no existing hole-filling methods satisfy all the above desirable properties. In particular, the robustness is hard to achieve. That is, most existing methods have difficulties in dealing with complex and highly curved holes. In this paper, we present a novel hole-filling algorithm for mesh models. The advancing front mesh technique is employed first to generate a new triangular mesh to cover the hole. Then we utilize the Poisson equation to optimize the new mesh. The algorithm is intended to be simple, fast and robust.

2 Preliminaries

2.1 Basic concepts

A triangular mesh is defined as a set of vertices and a set of oriented triangles that join these vertices. If two triangles share a common edge, the two triangles are adjacent triangles. An edge usually links two triangles. If it connects only one, the edge is called a boundary edge. A boundary vertex refers to the vertex on boundary edge. A hole is a closed loop of boundary edges. Boundary triangles are those triangles that own one or two boundary vertices. All triangles who share one common vertex are called the 1-ring triangles of the vertex. All edges who share one common vertex are called the 1-ring edges of the vertex. And, all vertices on 1-ring edges of a vertex (except itself) are called 1-ring vertices of the vertex. The vertex normal refers to the average normal of all the 1-ring triangles of the vertex. The triangle normal is the normal of the plane on which the triangle lies and can be computed as the average normal of its three vertices. Figure 1 illustrates these basic conceptions.

All mesh models in this paper are assumed to be oriented, manifold and connected, and a given hole is assumed to have no islands. To efficiently support our hole-filling algorithm, a vertex-based topological structure is also used in this work, which records 1-ring vertices, 1-ring triangles and 1-ring edges of every vertex of the model.

2.2 Algorithm overview

In order to make hole-filling simple, fast and robust, we propose a novel algorithm. The key ideas consist of two aspects. First, the advancing front mesh technique instead of the traditional 3D polygon triangulation method is em-

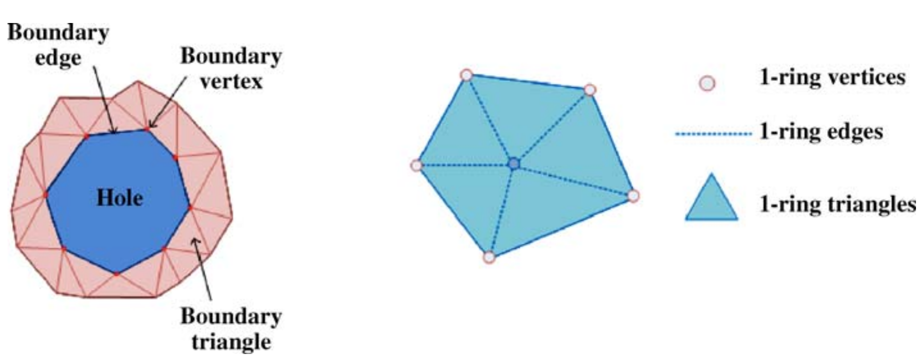


Fig. 1. Basic concepts related to a hole in a mesh model

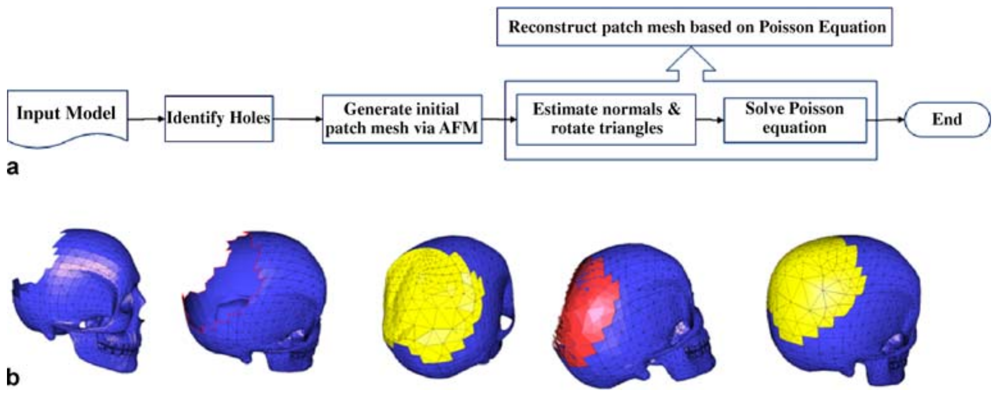


Fig. 2a,b. Algorithm flowchart and its illustration. **a** The flowchart. **b** The illustration of flowchart

ployed to generate the initial patch mesh to make the algorithm more robust and efficient. Second, the triangles involved in the initial patch mesh are modified by estimating their desirable normals instead of relocating them directly, and are re-positioned finally by solving the Poisson equation according to the desirable normals and the boundary vertices of the hole to make the algorithm more accurate. In this work, two schemes of computing desirable normals, e.g., harmonic-based scheme and geodesic-based scheme, are given and used so as to adapt to different situations. When the surrounding shape of the hole is relatively planar, the harmonic-based scheme is adopted. Otherwise, the geodesic-based scheme is utilized.

The main steps of the proposed hole-filling algorithm include:

Step 1. Identify holes in triangular mesh;

Step 2. For each hole in mesh model:

(1) Generate its initial patch mesh via the advancing front mesh (AFM) technique.
(2) Refine the patch mesh based on the Poisson equation as follows:
   - Compute desirable normals using the harmonic equation or geodesic interpolation.
   - Rotate triangles by local rotation.
   - Solve Poisson equation and obtain the new coordinates of every vertex.
   - Update the coordinates and obtain the smoothed patch mesh.

Figure 2 shows the flowchart of the proposed hole-filling algorithm. The detailed algorithms will be described in the following sections.

3 Hole identification

The first step of hole-filling is to detect all the holes in the given triangular mesh model. Since a vertex-based topological structure is used in this work, all boundary vertices can be easily identified by checking the numbers of their 1-ring triangles and 1-ring edges, i.e., if the two numbers of a vertex are not equal, the vertex is a boundary vertex. Once a seed boundary vertex is identified, from it a set of connected boundary edges can be traced. If all the boundary edges form a closed loop, they make up a hole. In this way, all the holes can be identified.

4 Initial patch mesh generation

In this work, we adopt the advancing front mesh (AFM) technique [10] to generate an initial patch mesh over the hole. The method consists of the following six steps:

Step 1. Initialize the front using the boundary vertices of the hole.

Step 2. Calculate the angle θ_i between two adjacent boundary edges(e_i and e_{i+1}) at each vertex v_i on the front.

Step 3. Starting from the vertex v_i with the smallest angle θ_i, create new triangles on the plane determined by e_i and e_{i+1} with the three rules shown in Fig. 3.

Step 4. Compute the distance between each newly created vertex and every related boundary vertex; if the distance between them is less than the given threshold, they are merged.

Step 5. Update the front.

Step 6. Repeat Steps 2 through 5 until the whole region has been patched by all newly created triangles.

This method can always patch the hole, whatever its shape [10], and guarantee the robustness of our algorithm. The triangular mesh created by AFM is called the initial patch mesh, because these triangles cover the hole in

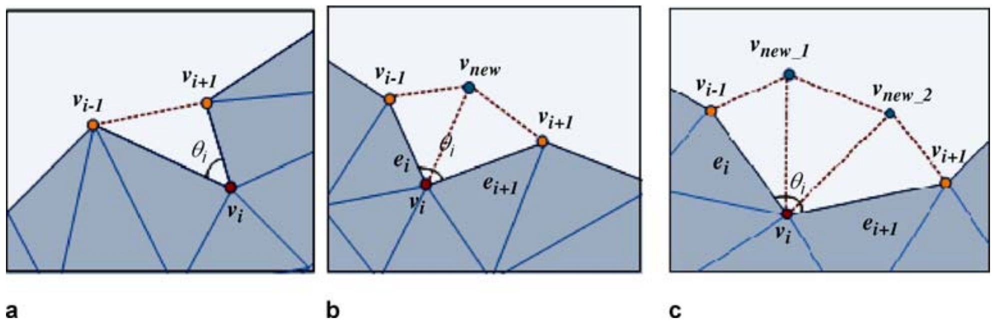


Fig. 3a–c. Rules for creating triangles: **a** $\theta_i \leq 75°$; **b** $75° < \theta_i \leq 135°$; **c** $\theta_i > 135°$

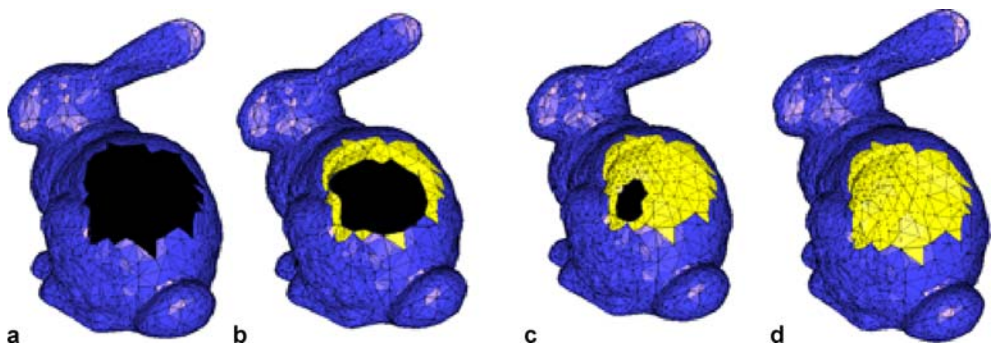


Fig. 4a–d. Process of AFM. **a** The original hole; **b** and **c** two intermediate stages; **d** the initial patch mesh created

a manner of a piece of a patch. The complete AFM process is illustrated in Fig. 4.

5 Refinement based on Poisson equation

Because the new vertices on the initial patch mesh are ragged and scraggly, they cannot adapt to the surrounding shape of the hole well. Therefore they need to be re-positioned in order to make the patch mesh connect the boundary vertices smoothly and approximate the missing geometry more accurately. In this work, the Poisson equation is employed to refine the patch mesh.

5.1 Poisson equation

In our implementation, we choose the Poisson equation with Dirichlet boundary conditions [18, 22] to refine the patch mesh. The Poisson equation with the Dirichlet boundary is formulated as

$$\Delta f = \operatorname{div} h, \quad f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{1}$$

where f is an unknown scalar function, $\Delta f = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is Laplacian operator, h is the guidance vector filed, $\operatorname{div} h$ is the divergence of h, and f^* is a known scalar function providing the boundary condition. It can be verified that the Poisson equation is the equivalent to the minimization problem

$$\min_f \int_\Omega |\nabla f - h|^2, \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{2}$$

where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ is the gradient operator.

The guidance vector field on a discrete triangle mesh is defined to be a piecewise constant vector function whose domain is the set of all points on the mesh surface. The constant vector is defined for each triangle, and this vector is coplanar with the triangle. Given a discrete vector field h on a mesh, its divergence at vertex v_i can be defined to be

$$(\operatorname{div} h)(v_i) = \sum_{T_i \in N_i} \nabla B_{i,k} \cdot h|T_k| \tag{3}$$

where N_i is the 1-ring vertices of v_i, $|T_k|$ is the area of triangle T_k, and $\nabla B_{i,k}$ is the gradient vector of the vertex within T_k. The discrete gradient of the scalar function f on a discrete mesh is expressed as

$$\nabla f(v) = \sum_i f_i \nabla \phi_i(v) \tag{4}$$

where $\phi_i(\cdot)$ is a piecewise linear basis function whose value is 1 at vertex v_i and 0 at all other vertices. The term

f_i is a scalar (vector) value attached to vertex v_i, and it is one of the coordinates of v_i in our implementation.

The discrete Laplacian operator is

$$\Delta f(v_i) = \sum_{v_j \in N_i} \frac{1}{2}(\cot\alpha_{i,j} + \cot\beta_{i,j})(f_i - f_j) \tag{5}$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ are the two angles opposite to edge (v_i, v_j) (see Fig. 5) and f_i is one of the coordinates of the vertex indexed i in our reconstruction step. Finally, the discrete Poisson equation is expressed as: $\Delta f \equiv \operatorname{div}(\nabla f) = \operatorname{div} h$.

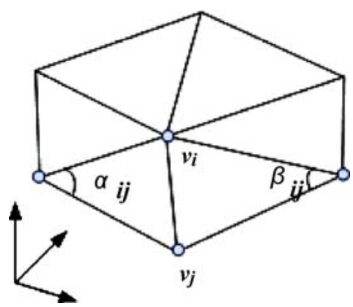


Fig. 5. Angles for the Poisson equation

The discrete Poisson equation is actually a sparse linear system

$$A\boldsymbol{x} = \boldsymbol{b} \tag{6}$$

where the unknown vector $\boldsymbol{x}$ represents special coordinates of all vertices on the reconstructed patch mesh, the coefficient matrix A is determined by Eq. 5, and the vector $\boldsymbol{b}$ is a known vector field obtained from the collection of divergence values at all boundary vertices formulated by Eq. 3, which is taken as the boundary condition.

The Poisson equation implies that in order to reconstruct the patch mesh we need a guidance vector field, i.e., h, defined on the triangles of the patch mesh. In this work, we construct the guidance field by triangles rotation.

5.2 Desirable normals computation and triangles rotation

Before performing triangles rotation, a desirable normal for each triangle should be calculated first. Because of the variety of holes, it is hard to find an almighty estimating scheme to compute desirable normals for all kinds of holes. In this work, we use two normal estimating methods, harmonic-based and geodesic-based, for different situations. The former is faster but only adapts to relatively planar holes and the second works well for the highly curved holes. The user will be asked to choose one to use at the beginning of hole-filling.

5.2.1 Harmonic-based desirable normal computing

The earliest purpose of discrete harmonic functions is to map a disk-like surface S_T onto a plane S^* [9, 20]. The basic idea is to find a piecewise linear mapping $f : S_T \to S^*$ to minimize the Dirichlet energy

$$E = \frac{1}{2}\int_{S_T} \|\text{grad}_{S_T}\|^2 \tag{7}$$

subject to the Dirichlet boundary condition $f|_{\partial S_T} = f_0$. As for triangle $T = \{v_1, v_2, v_3\}$, the Dirichlet energy can be expressed as

$$\begin{aligned}\int_{S_T} \|\text{grad}_T f\|^2 = \frac{1}{2}\big(&\cot\theta_3 \|f(v_1) - f(v_2)\|^2 \\ &+ \cot\theta_2 \|f(v_1) - f(v_3)\|^2 \\ &+ \cot\theta_1 \|f(v_2) - f(v_3)\|^2\big)\end{aligned} \tag{8}$$

where the angles are shown in Fig. 6. The equation for the minimization problem can therefore be re-expressed as the following linear system

$$\sum\nolimits_{v_j \in N_i} \omega_{i,j}(f(v_j) - f(v_i)) = 0, \quad v_i \in V_I \tag{9}$$

where $\omega_{i,j} = \cot\alpha_{i,j} + \cot\beta_{i,j}$, the angles $\alpha_{i,j}$ and $\beta_{i,j}$ are shown in Fig. 5, and N_i refers to the 1-ring vertices of vertex v_i. The associated matrix is symmetric and positive, and thus the linear system is uniquely solvable. The system can be solved efficiently by the conjugate gradient method.

It is observed that if we directly use the harmonic equation to obtain the altered coordinates of new vertices through the above method, most of the resulting mesh will be depressed, as shown in Fig. 7b. Obviously this is undesirable. In order to solve this problem, in this work the equation is constructed to estimate a new normal (desirable normal) for each vertex of the patch mesh instead of the vertex coordinates. Specifically, in Eq. 9, $f(v)$ refers to the normal rather than the coordinates of vertex v. Note that the

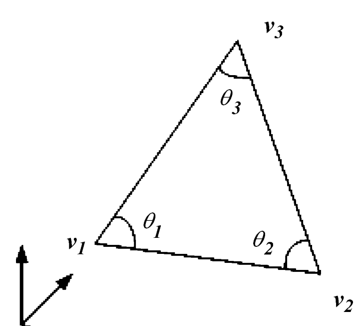


Fig. 6. Angles for the harmonic equation

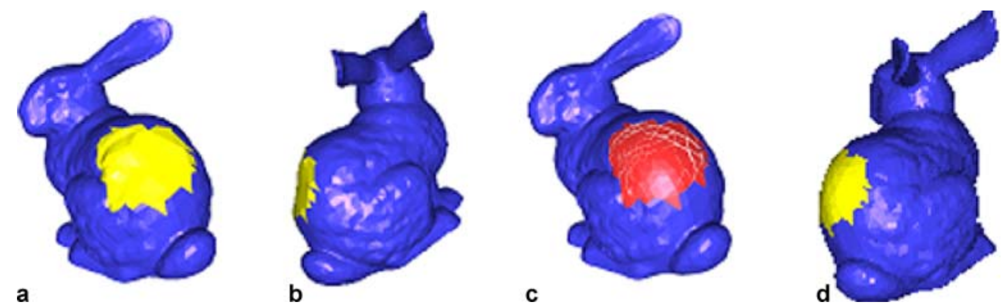


Fig. 7a–d. Reconstruction of the patch mesh. **a** The initial patch mesh (*yellow*). **b** The reconstructed mesh based on the harmonic equation. **c** The torn patch mesh after performing triangles rotation (*red*). **d** The reconstructed mesh based on the Poisson equation

system has to be solved three times, once for x-, once for y- and once for the z-component of the new normal.

5.2.2 Geodesic-based desirable normal computing

If the missing geometry of a hole is of high curvature and cannot be described by a quadratic function, the initial patch mesh will be sunken since the Dirichlet energy is quadratic. For such a case, we use geodesic interpolation to compute the desirable normals.

We first compute the geodesic distance from each new vertex, denoted by v, to all boundary vertices. The normal of boundary vertex v_b is then weighted by $\text{dis}(v, v_b)^{-n}$ and the final normal of v is the sum of weighted normals of all boundary vertices. Here $\text{dis}(v, v_b)$ denotes the geodesic distance from v to v_b. In our practice, $n \in [3, 5]$ is appropriate. As for overly complex holes, users can specify a threshold radius r, and the normal of vertex v is estimated by those boundary vertices whose geodesic distances to v are less than r. Our experimental results show that the geodesic-based normal estimation method is suitable for highly curved holes.

5.2.3 Triangle rotation

After the desirable normals of all new vertices are calculated, new normals of all the triangles involved in the initial patch mesh can be easily determined from the desirable normals of their three vertices. Then the triangles need to be rotated to the new orientation. The rotating is achieved by applying a local transformation to each triangle of the initial patch mesh. Suppose n_i, n_i' and c_i are the original normal, the new normal and the center of triangle f_i, then the rotating matrix R_i of f_i is obtained by rotating n_i to n_i' around c_i, by which the three vertices of triangle f_i are transformed to obtain their new coordinates. The local rotation is illustrated by Fig. 8.

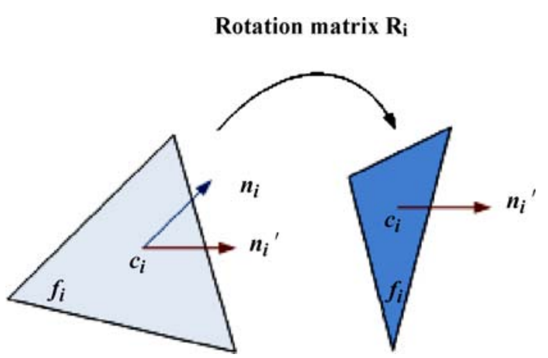


Fig. 8. Rotation of a triangle

5.3 Patch mesh reconstruction based on Poisson equation

After the triangle rotation is performed, all triangles of the patch mesh turn to a new direction. Essentially, the new normal of each triangle implies the “right” direction of the triangle. Because the local transformation applied to each triangle may be different, triangles of the patch mesh are usually torn apart and not connected to each other anymore (see Fig. 7c). It is these torn triangles that construct a guidance vector field the Poisson equation requires.

Once a discrete guidance vector fields is given, its divergence at a vertex can be computed. It is observed that local rotation to the triangles changes the gradients of each newly created vertex and three new gradient vectors are obtained (for x, y and z, respectively). If we consider the new gradient vector fields as the guidance fields in the Poisson equation, a piecewise continuous scalar function, i.e., a connected mesh, can be reconstructed, which keeps the gradients unchanged in the least squares sense.

The specific algorithm of reconstructing the patch mesh based on the Poisson equation includes the following steps:

Step 1. Compute gradients of each new vertex on their adjacent triangles using Eq. 4.

Step 2. Compute the divergence of every boundary vertex using Eq. 3.

Step 3. Determine the coefficient matrix A according to Eq. 5.

Step 4. Initialize the vector $\boldsymbol{b}$ using divergences of all boundary vertices.

Step 5. Set up and solve the Poisson equation and obtain the new coordinates of all vertices of the patch mesh.

Step 6. Update the vertices' coordinates and reconstruct the patch mesh.

Note that during computation of the divergences, the transformed vertices and the new normals of triangles

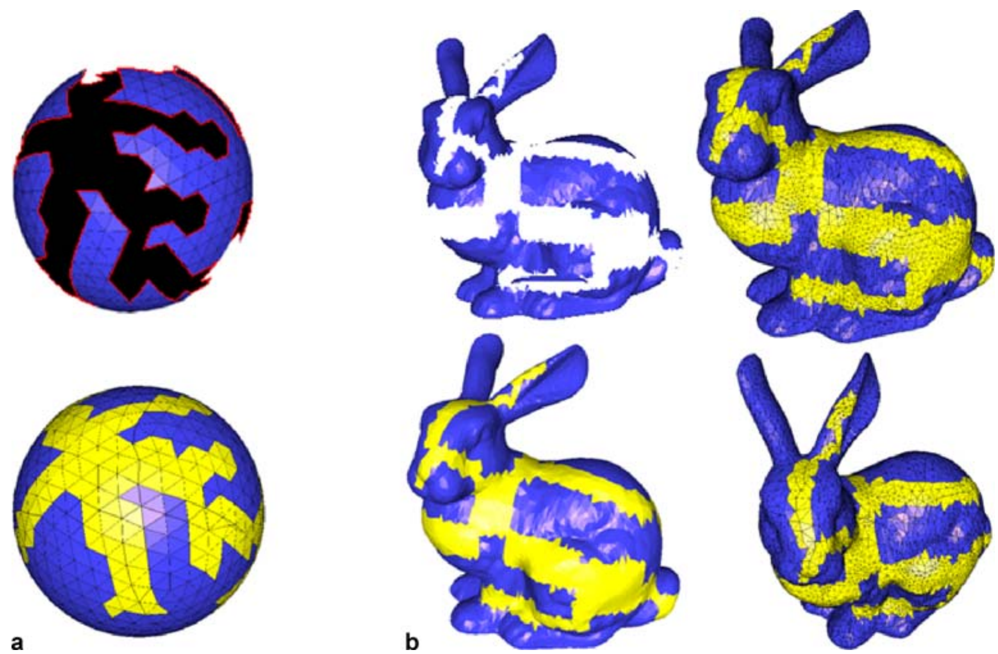


Fig. 9a,b. Two examples with complex holes. **a** A sphere with a complex hole and the result. **b** A bunny model with a complex hole and the result

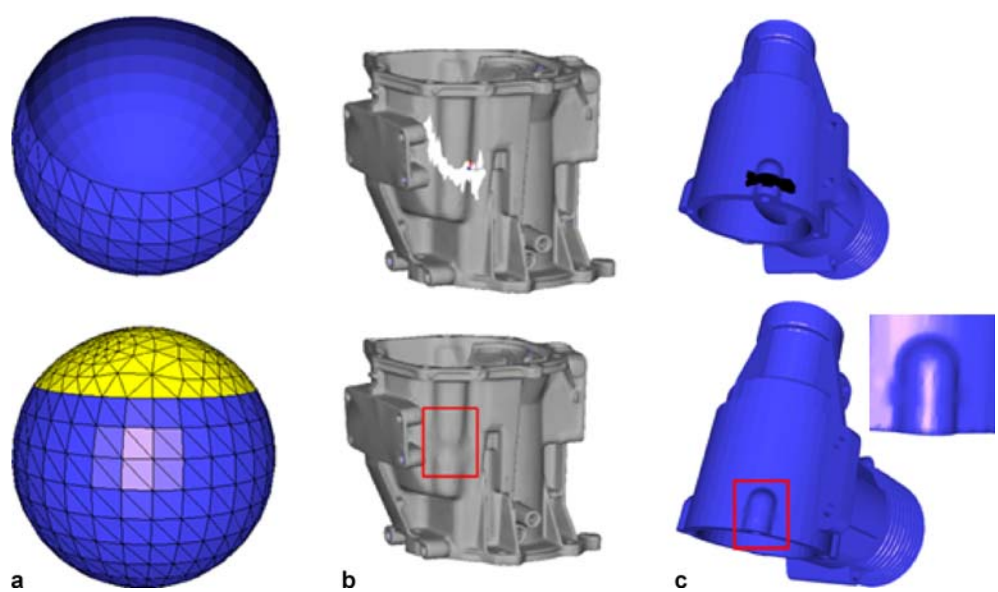


Fig. 10a–c. Examples with large and highly curved holes. **a** An incomplete sphere and the result. **b** An example from Jun and the result. **c** A similar case to **b** and our result. Geodesic-based normal estimation scheme ($n = 5$) is taken in this case. The new vertices number is 83 and the time of computing geodesic distance is 742 ms

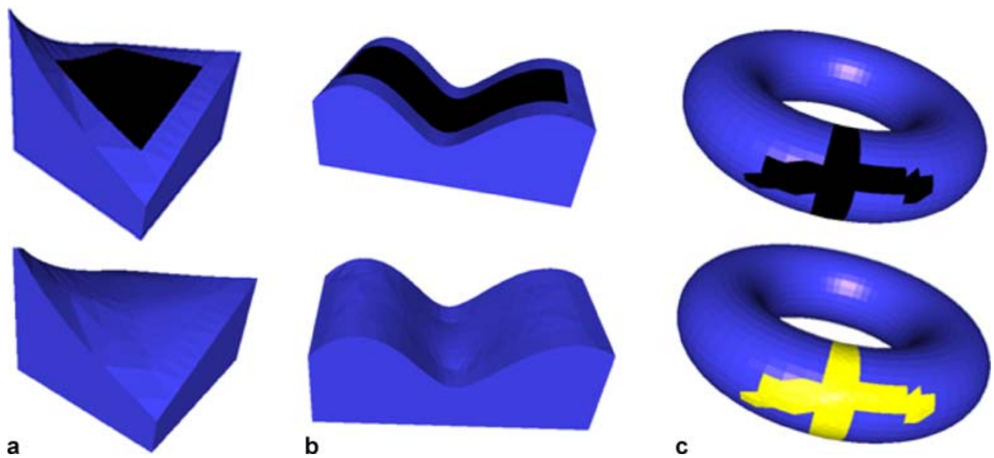


Fig. 11a–c. Three groups of models for error evaluation: **a** saddle; **b** span; **c** torus

Table 1. Error evaluation

Model name	Normals estimation scheme	Vertices num. on patch	Triangles num. on patch	Average distance	Face area	Average error
Saddle	Harmonic	280	648	0.10	4573	0.0015
Span	Geodesic ($n = 4$)	281	672	0.11	769	0.0039
Torus	Harmonic	111	192	0.05	4107	0.0024

Table 2. Computing time

Model name	Vertices num.	Triangles num.	New created vertices num.	Time of solving harmonic equation	Time of solving Poisson equation
Buddha	293 232	144 628	578	78 ms	97 ms
Dragon	100 207	20 250	281	33 ms	36 ms
Cow	25 289	50 238	181	8 ms	10 ms

rather than the original ones are utilized. Similar to the harmonic equation, this system has to be solved three times to obtain three coordinates of each new vertex.

Intuitively, the Poisson equation constructs a triangular mesh with known topology (connectivity) but unknown geometry (vertex coordinates), and solving the Poisson equation is analogous to stitching together the previously torn initial patch mesh again. As an example, Fig. 7d shows the patch mesh reconstructed from the patch mesh depicted in Fig. 7c based on the Poisson equation.

6 Implementation and analysis

The proposed hole-filling algorithm has been implemented with VC++7.1 and OpenGL. All experimental results in this paper were obtained on a 2.0 GHz Pentium IV personal computer with 1024 MB memory.

Many examples have been used to test the robustness, efficiency and accuracy of the method. Two mesh models with complex holes and the results of hole-filling with our approach are shown in Fig. 9. Figure 9a shows a sphere with a complex hole and the hole-filling result. Figure 9b depicts the Stanford bunny with a complicated hole and the hole-filling results. The experimental results show that our approach works well for complex holes and can produce high-quality mesh.

Figure 10a shows an incomplete sphere with 1/3 geometry lost and the hole-filling result. The experimental result indicates that our hole-filling algorithm approximates the missing geometry very well. Figure 10b,c are used to make a comparison between our algorithm and the work done by Jun [13]. Figure 10b is from [13], from which it can be seen that the filled region is subsided obviously. Figure 10c shows the hole-filling effect of our algorithm, and it appears to be more successful than the Jun's algorithm.

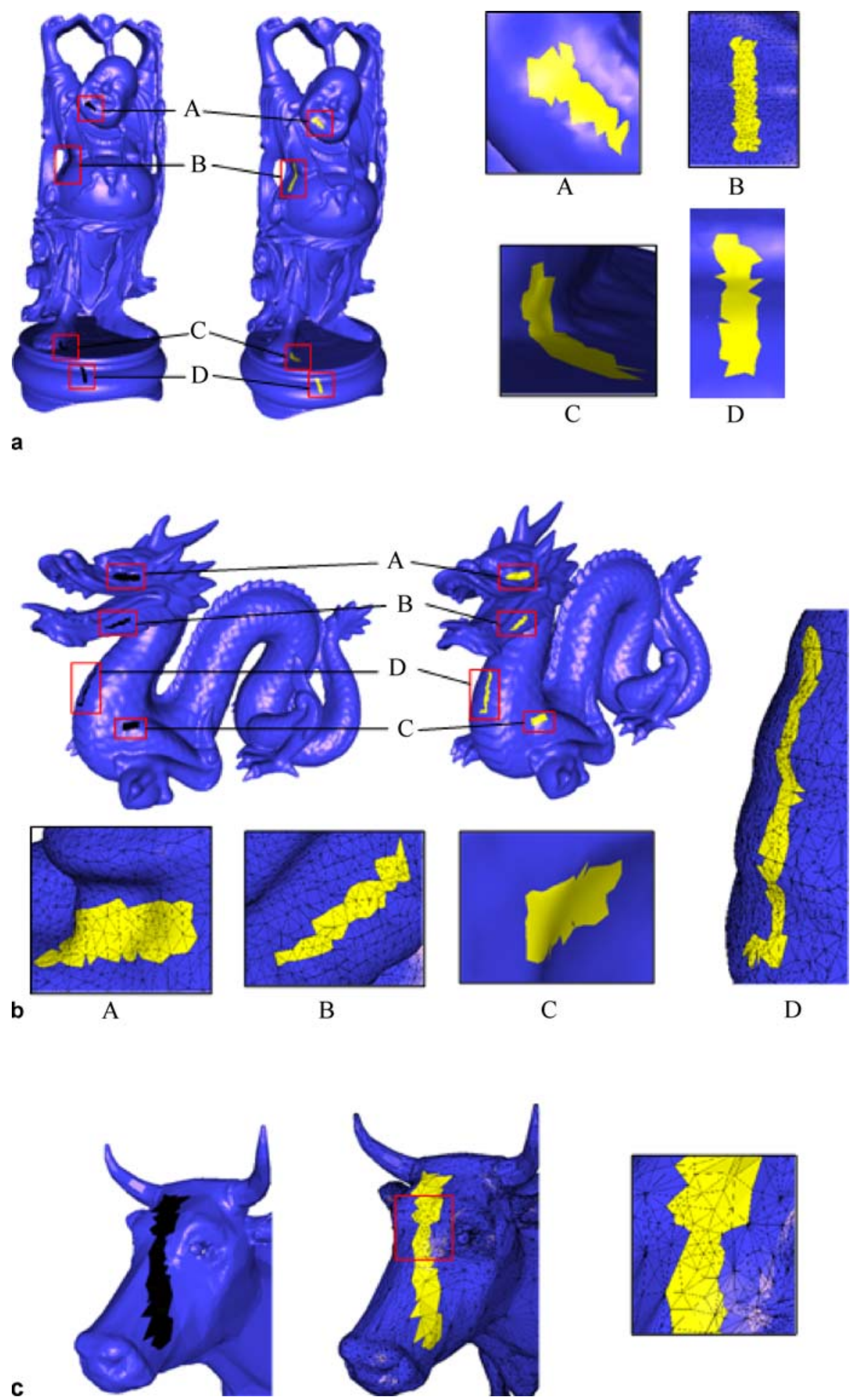


Fig. 12a–c. Three more complex examples: **a** buddha; **b** dragon; **c** cow

Three mesh models converted from their corresponding B-rep models are used to evaluate the precision of our algorithm, as shown in Fig. 11. The distances between the vertices on the patch mesh and the original analytic surface are used to evaluate the precision of our algorithm. The quotient of the average distance and the square root of analytic surface area is considered as the error of our algorithm. The evaluated errors of the three models are listed in Table 1. It can be seen that the average errors are all less than 0.5%.

Some other examples are shown in Fig. 12 and their computing times are shown in Table 2. According to the results, our algorithm is faster than voxel-based algorithms.

In our algorithm, computing geodesic distance is a time-consuming step. In this work, Surazhsky's method [21] is applied to compute geodesic distance between two vertices on a mesh. Fortunately, only the vertices on the patch mesh must be taken into account for computing the geodesic distances and the number of these vertices is much less than that of all the vertices of the whole model. Furthermore, only when the hole is highly curved is the geodesic-based scheme required.

Solving the Poisson equation and harmonic equation is another time-consuming step. When the number of the new vertices on the patch mesh is too large, it is hard to achieve interactive rates because of solving the two equations. But fortunately, only the vertices on the patch mesh need to be determined by solving the equations. According to our experiments, when the number of the new vertices is less than 5000, these equations can be solved in one second on our personal computer.

7 Conclusions

In this paper a novel algorithm for filling holes in triangular mesh is proposed. The advancing front mesh technique is used to cover the hole with newly created triangles, which guarantees the robustness of the approach. The triangles of the initial patch mesh are modified by estimating their desirable normals instead of relocating them directly. In this way, the algorithm becomes more accurate. Finally these triangles are re-positioned by solving the Poisson equation according to the desirable normals and the boundary vertices of the hole.

This approach is of good robustness for: (1) the advancing front mesh technique can close an arbitrary hole; (2) solving the harmonic and Poisson equation is a robust process because they are all symmetric and positive-defined linear systems and uniquely solvable.

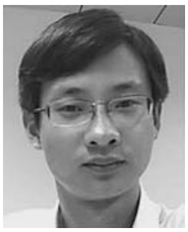
The normal estimating method plays an important role in our algorithm, directly determining the quality of the reconstructed patch mesh. How to approximate the normals more accurately and intuitively is our future work.

Acknowledgement This work is supported by the National Natural Science Foundation (No. 60574061, No. 60503057).

References

1. Carr, J., Beatson, R., Cherrie, J., Mitchell, T., Fright, W., McCallum, B.: Reconstruction and representation of 3D objects with radial basis functions. In: Processing of SIGGRAPH, Los Angeles, CA, 12–17 August 2001, pp. 67–76. ACM Press, New York (2001)
2. Chen, C.Y., Cheng, K.Y., Liao, H.Y.M.: A sharpness dependent approach to 3D polygon mesh hole filling. In: Proceedings of Eurographic, Trinity College, Dublin, Ireland, 29 August–2 September 2005, pp. 13–16. Blackwell Press (2005)
3. Chen, C.Y., Cheng, K.Y.: A sharpness dependent filter for mesh smoothing. Comput. Aided Geom. Des. **22**(5), 376–391 (2005)
4. Chen, C.Y., Cheng, K.Y., Liao, H.Y.M.: Fairing of polygon meshes via Bayesian discriminate analysis. The 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen-Bory, Czech Republic, pp. 175–182. UNION Agency-Science Press (2004)
5. Chui, C., Lai, M.-J.: Filling polygonal holes using C^1 cubic triangular spline patches. Comput. Aided Geom. Des. **17**(3), 297–307 (2000)
6. Curless, B., Levoy, M.: A volumetric method for building complex model from range image. In: Processing of SIGGRAPH, New Orleans, USA, 4–9 August 1996, pp. 303–312. ACM Press, New York (1996)
7. Davis, J., Marschner, S.R., Garr, M., Levoy, M.: Filling holes in complex surface using volumetric diffusion. In: Processing of 3D Data Processing Visualization and Transmission, Padova, Italy, 19–21 June 2002, pp. 428–433. IEEE Computer Society Press, New York (2002)
8. Dey, T.K.: Delaunay triangulation in three dimensions with finite precision arithmetic. Comput. Aided Geom. Des. **9**(6), 457–470 (1992)
9. Eck, M., DeRose, T.D., Kuchamp, T., Hoope, H., Lounsbery, M., Stuetzle, W.: Multi-resolution analysis of arbitrary meshes. In: Processing of SIGGRAPH, Los Angeles, CA, USA, 6–11 August 1995, pp. 173–182. ACM Press, New York (1995)
10. George, L.P., Seveno, E.: The advancing-front mesh generation method revisited. Int. J. Numer. Methods Eng. **37**(7), 3605–3619 (1994)
11. Girod, B., Greiner, G., Niemann, H.: Principles of 3D Image Analysis and Synthesis. Kluwer, Boston (2000)
12. Ju, T.: Robust repair of polygonal models. In: Processing of SIGGRAPH, Los Angeles, CA, USA, 8–12 August 2004, pp. 888–895. ACM Press, New York (2004)
13. Jun, Y.: A piecewise hole-filling algorithm in reverse engineering. Comput. Aided Des. **22**(8), 263–270 (2005)
14. Joshua, P., Szymon, R.: Atomic volumes for mesh completion. In: Proceedings Eurographics Symposium on Geometry Processing, Dublin, Ireland. 29 August–2 September 2005, pp. 33–41. Blackwell Press (2005)
15. Liepa, P.: Filling hole in meshes. In: Proceedings Eurographics Symposium on Geometric Processing, Granada, Spain, 1–6 September 2003, pp. 200–207. Blackwell Press (2003)
16. Levin, A.: Filling a N-sided hole using combined subdivision scheme. http://www.math.tau.ac.il/~levin/adi/pdf/nsided.pdf. Cited 1999
17. O'Rourke, J.: Computational Geometry in C. Cambridge University Press, New York (1999)
18. Perez, P., Gangnet, P., Blake, A.: Poisson image editing. In: Processing of SIGGRAPH, San Diego, California, USA, 27–31 July 2003, pp. 313–318. ACM Press, New York (2003)
19. Pfeifle, R., Seidel, H.P.: Triangular B-Spline for blending and filling of polygonal holes. In: Processing of Graphics Interface, Halifax, Nova Scotia, Canada, 22–24 May 1996, pp. 186–193. ACM Press (1996)
20. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Comput. Aided Des. **25**(4), 225–232 (1993)

21. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gorter, S.J., Hoope, H.: Fast exact and approximate geodesics on meshes. In: Processing of SIGGRAPH, Los Angeles, California, USA, 31 July–4 August 2005, pp. 553–560. ACM Press, New York (2005)
22. Yu, Y., Zhou, K., Shi, X., Bao, H. Guo, B., Shum, H.: Mesh editing with Poisson-based gradient field manipulation. In: Processing of SIGGRAPH, Los Angeles, California, USA, 8–12 August 2004, pp. 644–651. ACM Press, New York (2004)



WEI ZHAO is a PhD candidate of the State Key Lab of CAD&CG, Zhejiang University. He received his BS degree (2003) in Computer Science from the Shan Dong University of Science and Technology. His research interests include freeform feature technique, computer graphics, and geometry modeling, etc.

DR. SHUMING GAO is a professor of the State Key Lab of CAD&CG, Zhejiang University. He received his PhD degree from the Applied Mathematics Department of Zhejiang University in 1990, and was a visiting scholar and a visiting professor in the Design Automation Lab of Arizona State University, respectively, in 1996 and 2001. His research interests include product modeling, CAX integration, collaborative design, virtual reality in design and manufacturing, MEMS CAD, etc.

DR. HONGWEI LIN is an associate professor in State Key Laboratory of CAD&CG, Zhejiang University, China. He received his BSc from the Department of Applied Mathematics at Zhejiang University in 1996, and PhD from the Department of Mathematics at Zhejiang University in 2004. He worked as a communication engineer from 1996 to 1999. His current research interests are in computer-aided geometric design, computer graphics, and image process.