Computer-Aided Design 45 (2013) 1651–1664

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

A triangulation-based hole patching method using differential evolution

Wei-Cheng Xie, Xiu-Fen Zou *

School of Mathematics and Statistics, Wuhan University, Wuhan, 430072, China

HIGHLIGHTS

- A new hole patching method is proposed to repair the defective model.
- The information on both sides of the boundary around the considered hole is used.
- The points in the hole region are predicted by differential evolution.
- The operations of mesh optimization are used to improve the quality of the mesh.

ARTICLE INFO

Article history:
Received 1 March 2013
Accepted 9 August 2013

Keywords:
Hole patching
Triangulation model
Differential evolution
Point correspondence
Mesh optimization

ABSTRACT

In this work, a new hole patching method (namely as, HPDE) is proposed to repair the damaged or ill-scanned three dimensional objects in real engineering applications. Our method differentiates from other related algorithms mainly on the following three aspects. Firstly, our algorithm sufficiently utilizes the point information around the considered hole for each prediction by constructing point correspondences on both sides of the boundary curve of the hole; secondly, the missing points in the hole region are predicted by the algorithm of differential evolution (DE), which is used to obtain the topological and geometrical structures of the mesh in the hole region; thirdly, operations of mesh optimization are adopted for improving the quality of the obtained triangulation mesh. Numerical results on kinds of holes with complex shape and large curvature, and a comparison with two recently proposed algorithms verify the effectiveness of the algorithm, further experiments on the noisy data points illustrate the robustness of the algorithm against noise.

1. Introduction

Methods for constructing a mathematical model from a given set of three dimensional (3D) points have been applied into a wide variety of fields, such as engineering design, virtual reality, movie making and data visualization. Surface reconstruction has drawn much attention from last decades of years and a lot of methods are reported [1–4]. It is more often that the provided points are represented with piecewise triangles (namely as triangulation model), such as the Marching Cubes method in [5], the crust algorithm in [6], direct advancing front method and modified decimation method in [7], and a divide and conquer Delaunay triangulation in [8].

However, the constructed model may not be directly applicable for the real application. In one case, the provided triangulation model is damaged or the points in the considered model are ill-scanned. In another case, the provided 3D points can also be the measured parameter values from the simulation experiments, and it may come up that the data referring the parameter values in the abnormal conditions is difficult to be obtained by the simulations, which may be the result from the difficulty of these experiments or the high cost and time-consuming of the simulation. For these two cases, there still exist concerned regions (hole regions) where the point information is unknown and needs to be predicted based on the constructed model, thus, the hole patching method is resorted to.

In this work, a new hole patching method (HPDE) based on defective triangulation model is devised to patch challenging holes with large curvature. Based on the boundary points and boundary directional vectors, the proposed HPDE utilizes algorithms of differential evolution (DE) and constrained triangulation sequently to obtain the topological and geometrical structures of the triangulation mesh in the hole region, which is further resorted to the algorithms of smoothing and mesh optimization to improve the mesh quality. The proposed HPDE differs from other hole patching methods mainly on the following aspects. (1) The proposed HPDE combines the frequently used algorithms of DE, constrained triangulation and mesh optimization in the application of hole patching.

* Corresponding author. Tel.: +86 27 68772958; fax: +86 27 68752256.
E-mail addresses: zouxiufen@yahoo.com, xfzou@whu.edu.cn (X.-F. Zou).

0010-4485/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cad.2013.08.003

(2) Different from other algorithms obtaining the topological structure in advance, our method obtains the topological and geometrical structures of the mesh in the hole region at the same time. (3) HPDE sufficiently utilizes the directional vectors at each two corresponding base points on the boundary curve around the considered hole for the prediction, which is different from other algorithms either using only one side of boundary information or not using the directional vectors at the boundary points.

This paper is organized as follows. Some related work of hole patching are introduced in Section 2. Section 3 presents the description of the classical DE algorithm. The proposed HPDE is described in Section 4. Section 5 presents all the numerical results on some kinds of test models. Finally, some discussions and conclusions are given in Sections 6 and 7, respectively.

2. Related work

Method of hole patching is to fill in a considered region by expanding the constructed model around the region. For the hole patching, the surrounding region of the hole region is filled with the point information of the constructed model, then hole regions are patched with the methods which are similar to interpolation by incorporating this information. A few methods referring to the hole patching have been sprang up in the past few years. Based on the representation of the constructed model, these methods are summarized into mainly two categories, one is volume-based, the other is surface-based.

Generally, the volume-based methods patch the holes by first assigning signs to a set of voxels with a signed distance function, then the point information in hole region is completed in the volume representation, which is used to approximately obtain the missing interface of the geometry by some surface-extracting methods such as the marching cubes method. The model with holes was converted to the volumetric representation, on which form the model was naturally repaired by parity count method or ray stabbing method, then the repaired model was converted back to a waterproof polygonal model [9]. The surface model was first converted to the model with volume representation, then a diffusion was employed to extend the surface to form a watertight model [10]. This method can deal with geometrically and topologically complex holes. A novel hole-filling algorithm was proposed for volumetric objects where not only the outer surface of 3D objects but also the solid volumetric objects were closed filled [11]. The main disadvantage of these volume-based methods is that they may miss some important features of the original model when converting to and from a volume, although these methods can often produce a watertight model in a robust way.

The surface-based methods deal with the data points directly and they often utilize the information in a local region around the considered hole. For these surface-based methods, the topological structure of the triangulation mesh in the hole region is first obtained, then the geometrical structure is obtained by some other algorithms. The defects and holes were proposed to be repaired by a 3D triangulation method which is to minimize the total sum of edge lengths by dynamic programming [12]. The constrained triangulation of the unfolded 3D points on a plane was embedded into the triangular mesh by minimizing an energy surface [13]. The radial basis function interpolator using neighboring edges was employed to fill the holes [14]. The topological structure of the prediction points in the hole region was obtained by the advancing front method, and the geometric structure of these points was determined by solving the Poisson equation [15]. The surface-based approach and a two-step volume-based method-heat diffusion and the Poisson surface reconstruction were incorporated to repair holes with geometric and topological complexities [16]. It was proposed that complex polygonal holes were filled in a piecewise manner so as to obtain the entire hole triangulation with the piecewise planar triangulation [17]. The patching mesh in hole region was obtained by iteratively refining and smoothing the preciously optimized mesh according to the lengths of the adjacent triangles, based on an initial triangulation mesh of the boundary points of the considered hole region [18,19]. A grey prediction model was proposed to predict and adjust the coordinates of the newly added points by obtaining two controlled variables—the normal vector and the included angle size [20], this method tends to maintain the variation trend of the boundary points by utilizing the point information adjacent to the prediction points. While the used point information is only on one side of the boundary, the variation trend from one side to another side in the hole region cannot be detected, which may result in losing fidelity for holes where the points vary uniformly from one side to the other. The hole regions were patched with non-uniform rational B-spline surface by utilizing several layers of adjacent triangles [21], this method first obtains several interpolation spline curves using some adjacent points around the hole region, then some discrete points in the hole region are extracted from these interpolation curves, which are used for a B-spline surface interpolation to obtain the geometrical structure of the triangulation mesh. Because of the characteristic of the B-spline function that the shape taken by the spline is with minimum elastic energy, the patched surface in the hole region is apt to be flat which may also result in losing fidelity.

These reported methods adopt their own strategies to predict the points in the hole region, and achieve competitive results on their concerned test models. However, how to sufficiently utilize the information around the considered hole and how to effectively handle challenging holes are still hot topics in this field.

3. DE algorithm

As a stochastic algorithm which is first proposed in [22,23], DE and its variants are frequently adopted in various kinds of real engineering applications [24]. Different from classical gradient-based optimization algorithms, DE adopts one greedy strategy to proceed to global optimum from multiple positions of the searching region based on a random generated population. Without utilizing the gradient information of the objective function, DE is often the real alternative for a variety of non-differentiable and non-convex problems which are difficult for gradient-based algorithms.

Without loss of generality, assume the following minimization problem is considered:

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$, and $x_j \in [l_j, u_j], j = 1, \ldots, n$. For stochastic algorithms which are population based, denote population as $\mathbf{X} \triangleq \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, where N is the number of individuals. Denote $\mathbf{X}^g \triangleq \{\mathbf{X}_1^g, \ldots, \mathbf{X}_N^g\}$ as the population at the generation of g.

DE algorithm employs the diversity information implied in the population to generate the mutation individual, the most frequently used version of DE algorithm is *DE*/*rand*/1/*bin*, where *rand* denotes choosing random vectors for mutation, 1 denotes employing one difference term for the mutation procedure, and *bin* denotes generating a trial individual by accepting the parameter values from the mutation individual one at a time. This version is detailed described in Algorithm 1 and our proposed algorithm is devised based on this version.

There are other mutation and crossover strategies which compose other versions of DE [24]. In addition to *rand*/1 mutation strategy, *best*/1 and *rand*/2 are also frequently used, which are listed as follows:

1. *best*/1:

$$\mathbf{V}_i = \mathbf{X}_*^g + F \cdot (\mathbf{X}_{r_2^{(i)}}^g - \mathbf{X}_{r_3^{(i)}}^g) \tag{6}$$

Algorithm 1 DE algorithm [23]

1: Population initialization: $\mathbf{X}_{i,j}^0 = l_j + rand \cdot (u_j - l_j), i = 1, \ldots, N, j = 1, \ldots, n$, *rand* is an uniform random number in [0, 1] for every $\{i, j\}$. $g = 0$.
2: **while** the termination condition is not satisfied **do**
3: **for** $i = 1, \ldots, N$ **do**
4: Randomly choose mutually different indexes $r_1, r_2, r_3 \triangleq r_1^{(i)}, r_2^{(i)}, r_3^{(i)} \in \{1, \ldots, N\}$, randomly choose rn_i from $\{1, \ldots, n\}$.
5: **Mutation:** The mutation individual $\mathbf{V}_i$ is generated as

$$\mathbf{V}_i = \mathbf{X}^g_{r_1^{(i)}} + F \cdot (\mathbf{X}^g_{r_2^{(i)}} - \mathbf{X}^g_{r_3^{(i)}}) \tag{2}$$

where $F \in [0, 1]$ is the mutation probability.
6: **Crossover:** The trial individual $\mathbf{U}_i$ is created as

$$\mathbf{U}_{i,j} = \begin{cases} \mathbf{V}_{i,j}, & \text{if } (rand \le CR) \text{ or } (j = rn_i), \\ \mathbf{X}^g_{i,j}, & \text{otherwise} \end{cases} \tag{3}$$

where *CR* records the crossover probability and *rand* is a new generated uniform random number in [0, 1] for every $\{i, j\}$.
7: **Selection:** The i-th individual $\mathbf{X}_i^{g+1}$ of the generation $g + 1$ is chosen as

$$\mathbf{X}_i^{g+1} = \begin{cases} \mathbf{U}_i, & \text{if } f(\mathbf{U}_i) \le f(\mathbf{X}_i^g), \\ \mathbf{X}_i^g, & \text{otherwise.} \end{cases} \tag{4}$$

8: Constrain the variables to the given regions

$$\mathbf{X}_{i,j}^{g+1} = \begin{cases} \min\{2 \cdot l_j - \mathbf{X}_{i,j}^{g+1}, u_j\} & \text{if } \mathbf{X}_{i,j}^{g+1} < l_j, \\ \max\{2 \cdot u_j - \mathbf{X}_{i,j}^{g+1}, l_j\} & \text{if } \mathbf{X}_{i,j}^{g+1} > u_j, \\ \mathbf{X}_{i,j}^{g+1} & \text{otherwise.} \end{cases} \tag{5}$$

9: Renew the objective function value of each new generated individual. $g \leftarrow g + 1$.
10: **end for**
11: **end while**

2. *rand*/2:

$$\mathbf{V}_i = \mathbf{X}^g_{r_1^{(i)}} + F \cdot (\mathbf{X}^g_{r_2^{(i)}} - \mathbf{X}^g_{r_3^{(i)}}) + F \cdot (\mathbf{X}^g_{r_4^{(i)}} - \mathbf{X}^g_{r_5^{(i)}}) \tag{7}$$

where $\mathbf{X}^g_*$ denotes the best individual in the generation g. For the crossover operator, there is another strategy which is often used in the literature, namely as *exp* crossover, which adopts some consecutive components from the mutation individual with a predetermined probability. The performance of DE algorithm on complex problems is sensitive to the parameters *F* and *CR* in Algorithm 1, consequently, these parameters are often set to be self-adaptive. From another aspect to enhance the performance of this algorithm, gradient-based algorithms are frequently incorporated into it [25].

For employing DE algorithm, only several terms need to be concerned and input:

- The objective function $f(\cdot)$.
- The lower and upper bounds $\{l_i, u_i, i = 1, \ldots, N\}$ of optimization variables.
- The stopping conditions (maximum iteration generations or time).
- The parameter values (F, CR, N) of DE algorithm.

4. The proposed method HPDE

In this section, we first generally introduce the proposed HPDE in Algorithm 2 and then explain the seven steps of this algorithm concretely in seven sub-sections.

Algorithm 2 The description of the proposed HPDE

1: **Step 1:** Model the provided points with triangulation and preprocess.
2: **Step 2:** Extract the boundary points and triangles, and obtain a set of dense base points $\{BP_1, \ldots, BP_n\}$ on the boundary curve. Segment the base points if necessary.
3: **Step 3:** Find and modify pairs of correspondence of the base points.

$\{\cdots; BP_{i-1}, BP_{n-i+2}; BP_i, BP_{n-i+1}; BP_{i+1}, PB_{n-i}; \cdots\}$

4: **for** each pair of correspondence points BP_i, BP_{n-i+1} **do**
5: **Step 4:** Construct a section plane across the correspondence points, find their intersection points $\{BP_i, AP_i, AP_{n-i+1}, BP_{n-i+1}\}$ with the edges of the adjacent triangles.
6: **Step 5:** Predict the points $\{PP_1, \ldots, PP_m\}$ between the correspondence points on the section plane by DE algorithm.
7: **end for**
8: **Step 6:** Conduct 2D constrained triangulation on the projections of the boundary points and all the prediction points on a rotated 3D plane.
9: **Step 7:** Smooth and optimize the obtained constrained triangulation.

4.1. Preprocess

This overall algorithm is based on the damaged triangulation model of a 3D object. When it is provided with a set of 3D points, triangulation model needs to be constructed in this step. This step also includes operation of smoothing procedure when the provided points are noisy. In this work, an easy-to-implement form of smoothing method in [26] is adopted, which is reported to be able to avoid surface shrinkage by conducting a Gaussian filter step and an un-shrinking smoothing step consecutively.

4.2. Extract and segment boundary

The boundary curve around a hole is a closed path which is composed of a set of edges of the adjacent triangles, where there are at least one edge which does not have adjacent triangles for these triangles. A hole identification method was proposed in [14] by studying the torsion of the contour curve, which can distinguish the natural holes from those man-made holes.

Besides of the boundary curve, a ring of adjacent triangles around the considered hole are obtained (see Fig. 4), and a set of dense base points $\{BP_1, \ldots, BP_n\}$ with the same arc length are also extracted on the boundary curve. The number of base points n is a parameter in the algorithm (Assume the boundary points and triangles are sorted in advance), which is adjusted according to the average length of the edges of the adjacent triangles, that is, $n = \lfloor \frac{BudLen}{0.8 \cdot EdgeLen} \rfloor$, where *BudLen* is the length of the boundary of the considered hole, *EdgeLen* is the average of the edge lengths of the surrounded triangles, $\lfloor a \rfloor$ is the floor of the number a.

Because the following triangulation is conducted on the projected prediction points on a 3D plane, overlapping or self-intersection may occur when these prediction points are high curled, in which case, the points need to be divided and processed with respect to (w.r.t.) two planes separately. In this step, we judge whether the boundary points and the following prediction points need to be segmented, which is stated in the following.

The approximately optimal segmentation of the boundary base points is determined by a few tries through the principal component analysis (PCA) method in the following. It is implied in [1] that the size of the smallest eigenvalue of PCA with a set of 3D points determines the degree of approximation of these points to a plane, and the smaller the smallest eigenvalue compared to the

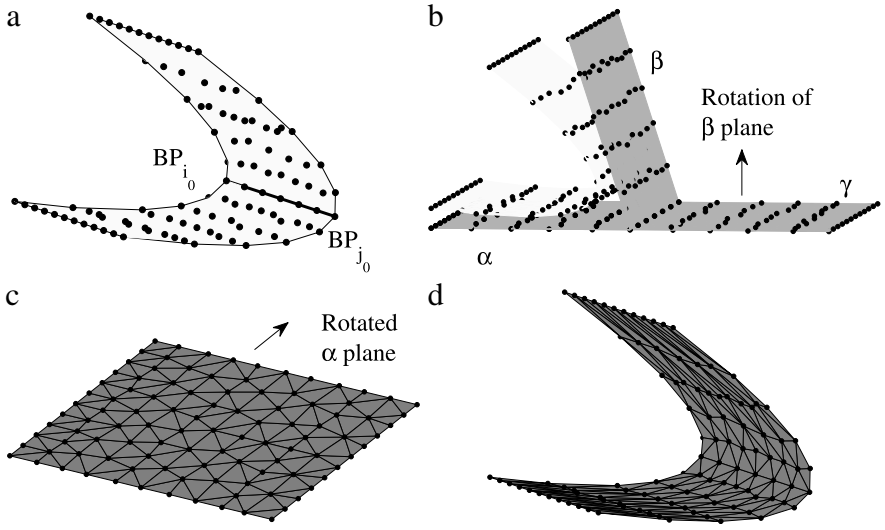

Fig. 1. Illustration of the projection of points on two planes and the constrained triangulation. Fig. (a) depicts the prediction points in the hole. In Fig. (b), these prediction points are first projected onto two planes α and β, then onto an unique plane $\alpha(\gamma)$. Result of 2D constrained triangulation on the rotated plane α is shown in Fig. (c). The local triangulation is applied to the initial prediction points in (d).

largest eigenvalue, the more approximated to a plane these points are. Thus, the larger of the two smallest eigenvalues through the PCA method is minimized to obtain the optimal segmentation by the following optimization problem.

$$\min_{1\le i<j\le n} \max\{\lambda_{3,PS_1}, \lambda_{3,PS_2}\}, \tag{8}$$

where $PS_1 = \{BP_j, \ldots, BP_n, BP_1, \ldots, BP_{i-1}\}$, $PS_2 = \{BP_i, \ldots, BP_{j-1}\}$, $\lambda_{3,PS_1}, \lambda_{3,PS_2}$ are the smallest eigenvalues of the two segmented sets of points PS_1, PS_2, respectively, by PCA method. When the dihedral angle of these two fitting planes is smaller than a critical angle A_c ($A_c = \pi/2$ is chosen for all the test problems which is discussed in Section 4.8), the boundary base points are segmented, otherwise, the boundary points remain intact. The boundary of the provided points in Fig. 1(a) is segmented, which determines two planes α and β in Fig. 1(b).

4.3. Find the correspondences

The prediction of the proposed HPDE is conducted between each pair of base points, the correspondences of all the base points on the boundary are found and modified in this step. If the boundary base points are segmented, the correspondences of the indexes of the base points are mandatorily determined as follows:

$$\{\ldots, (i_0-1, j_0+1), (i_0, j_0), (i_0+1, j_0-1), \ldots\} \tag{9}$$

where i_0, j_0 are the optimal segmentation points obtained by solving the problem in Eq. (8), all the indexes are incremented and decremented modulo n.

When the base points need not to be segmented, to obtain good correspondences, the sum of all the clamping distances between the corresponding points should be minimized to reduce the uncertainty in the hole region and to sufficiently utilize the adjacent information around the hole. The minimization of the clamping distances is approximated by optimizing the problem in Eq. (10) w.r.t. a starting position $sp \in \{1, \ldots, n/2\}$.

$$\begin{aligned} &\min_{sp} Dist = \sum_{i=1}^{n/2} \|BP_{Seq(i)} - BP_{Seq(n-i+1)}\|, \\ &Seq = \{sp, \ldots, n, 1, \ldots, sp-1\}, \end{aligned} \tag{10}$$

where $\|\cdot\|$ is the Euclidean norm which is the form of the norm in the following text unless explicitly stated otherwise, $Seq(i)$ is the ith element of Seq.

The preceding obtained correspondences need to be slightly modified to exclude the case that the intersection angle of the boundary and the line linking two corresponding points is close to zero. The modification is conducted by directly skipping the base points which render the intersection angle being smaller than a critical angle I_c (I_c is set to be $\pi/15$ in this work which is discussed in Section 4.8). In Fig. 2(a), the vector $BP_{n-i+1}BP_i$ intersects the vector Q_tQ_{t+1} with a small angle, the point BP_i is modified to that in Fig. 2(b).

4.4. Construct section planes

In this step, a section plane is constructed for each pair of corresponding points, where the section plane is assumed to cross the line linking two corresponding points and only the normal direction of the plane needs to be determined. Denote three adjacent pairs of corresponding base points as $\{BP_{i-1}, BP_{n-i+2}; BP_i, BP_{n-i+1}; BP_{i+1}, BP_{n-i}\}$ (where the base points $\{BP_1, \ldots, BP_n\}$ have been sorted such that the first point is on the position of the optimal starting point sp^* in Eq. (10) and the rotated point set is still denoted as $\{BP_1, \ldots, BP_n\}$), the normal direction $\overline{Nv}$ of the section plane across the points BP_i, BP_{n-i+1} is computed as in Eq. (11) (see Fig. 2(c)).

$$\begin{aligned} &\overline{Nv} = \overline{lv} \times \overline{n}, \\ &\overline{lv} = BP_i - BP_{n-i+1}, \\ &\overline{n} = \overline{lv} \times \left(\frac{BP_{i-1} + BP_{n-i+2}}{2} - \frac{BP_{i+1} + BP_{n-i}}{2} \right), \end{aligned} \tag{11}$$

where $\overline{lv} \times \overline{n}$ is the outer product of the vectors $\overline{lv}$ and $\overline{n}$.

Another two intersection points (besides of the points BP_i, BP_{n-i+1}) of the constructed section plane and the adjacent ring triangles are found, which are denoted as AP_i, AP_{n-i+1}.

4.5. Predict the points

By using the four intersection points $\{BP_i, AP_i, AP_{n-i+1}, BP_{n-i+1}\}$ on the edges of the boundary ring triangles, several points are predicted on the section plane in this step. The prediction points are deemed to be appropriate if the following conditions are satisfied:

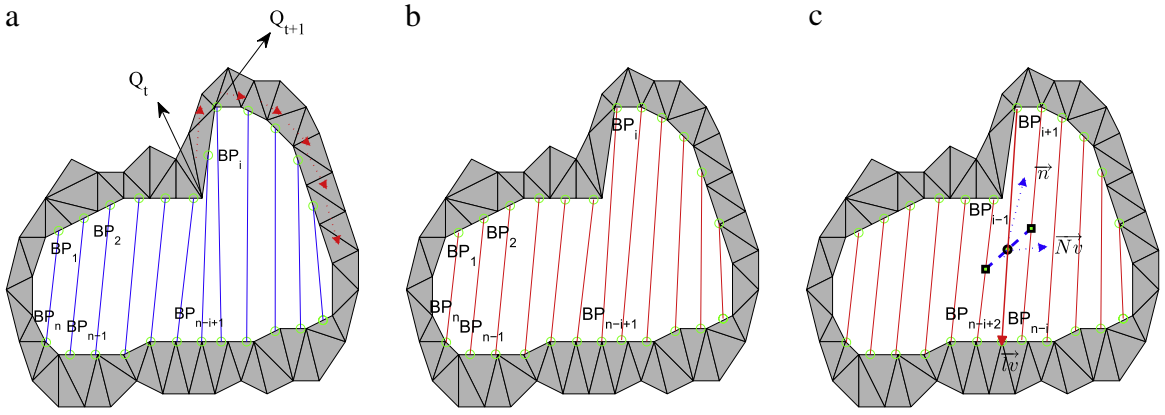

Fig. 2. Modification of the correspondence and derivation of the normal direction. In (a), the vector Q_tQ_{t+1} intersects the vector $BP_{n-i+1}BP_i$ with a small angle. The point BP_i and the following points have been moved to new positions in (b). Fig. (c) depicts the procedure of obtaining the normal direction.

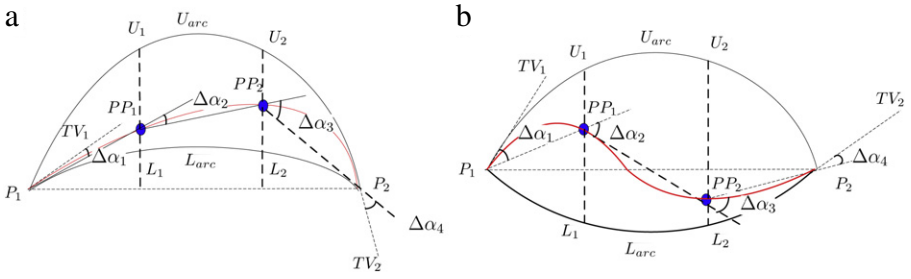

Fig. 3. Two cases of the prediction of two points on a rotated 3D section plane.

- The distances between each two consecutive points should be equal.
- The intersection angles of each pair of consecutive connection lines of the prediction points should be equal.

The following illustration is concentrated on obtaining the prediction points to satisfy the preceding conditions as much as possible. Denote two correspondence points as $\{P_1, P_2\} \triangleq \{BP_i, BP_{n-i+1}\}$, and two endpoint tangent vectors as $\{TV_1, TV_2\} \triangleq \{P_1 - AP_i, AP_{n-i+1} - P_2\}$ (see Fig. 3), where $\{AP_i, AP_{n-i+1}\}$ are another two intersection points of the ith section plane and the adjacent triangles besides of two base points $\{BP_i, BP_{n-i+1}\}$. Linearly rotating the section plane along with P_1, P_2, TV_1, TV_2 to a 3D plane parallel to xy-plane, the rotated points and vectors are still denoted as P_1, P_2 and TV_1, TV_2. The two vectors TV_1, TV_2 determine two circular arcs L_{arc}, U_{arc} whose tangent vectors at the points P_1, P_2 are TV_1, TV_2, respectively.

To obtain prediction points with uniform distances, the first coordinate (x-coordinate) values are computed by averaging two sets of x-coordinate values of m points with uniform distances on the two arcs L_{arc}, U_{arc}. That is,

$$PP_{i,1} = \frac{1}{2}(P^L_{i,1} + P^U_{i,1}), \quad i = 1, \ldots, m \tag{12}$$

where $\{P^L_i, P^U_i, i = 1, \ldots, m\}$ are two sets of points on the arcs L_{arc}, U_{arc}, respectively, $\{P^L_{i,1}, P^U_{i,1}\}$ are the corresponding x-coordinate values. These points satisfy the following conditions

$$\begin{cases} \|P^U_i - P^U_{i-1}\| = \|P^U_{i+1} - P^U_i\|, \\ \|P^L_i - P^L_{i-1}\| = \|P^L_{i+1} - P^L_i\|, \quad i = 1, \ldots, m \\ P^L_0 = P_1, \quad P^L_{m+1} = P_2; \quad P^U_0 = P_1, \quad P^U_{m+1} = P_2. \end{cases} \tag{13}$$

To obtain prediction points with uniform intersection angles, the second coordinate (y-coordinate) values $\{PP_{i,2}, i = 1, \ldots, m\}$ are determined by optimizing the following optimization problem

$$\min_{\{PP_{i,2}, i=1,\ldots,m\}} \max \ \Delta\alpha_i = \min f(\{PP_{i,2}\}),$$
$$\text{subject to } l_i = L_{i,2} \le PP_{i,2} \le U_{i,2} = u_i, \ i = 1, \ldots, m \tag{14}$$

where $\Delta\alpha_i = \langle PP_i - PP_{i-1}, PP_{i+1} - PP_i\rangle$ is the angle of the vectors $PP_i - PP_{i-1}, PP_{i+1} - PP_i$ (We denote $PP_0 = P_1, PP_m = P_2$), L_i, U_i are the intersection points of the line $x = PP_{i,1}$ and the arcs L_{arc}, U_{arc}, respectively (see Fig. 3). The number of prediction points m is carefully chosen such that the distances between the prediction points are approximately equal to the average length of the edges of the corresponding adjacent triangles, which is computed as follows:

$$\begin{cases} m = \left\lfloor \dfrac{Len}{MaxLen} \cdot M \right\rfloor + 1, \\ M = \left\lfloor \dfrac{MaxLen}{EdgeLen} \right\rfloor + 1 \end{cases} \tag{15}$$

where Len is the length of the considered two correspondence base points, $MaxLen$ is the maximum of all the lengths, $EdgeLen$ is the average of the edge lengths of the surrounded triangles.

The optimization problem defined in Eq. (14) is directly solved by the DE algorithm described in Algorithm 1. Fig. 3(a) presents $m = 2$ prediction points $\{PP_1, PP_2\}$ in the clamping interval of the points P_1, P_2 on a 3D plane when the two endpoint vectors TV_1, TV_2 have opposite directions. Fig. 4(a) demonstrates these prediction points $\{PP_1, PP_2\}$ in the original 3D space. Figs. 3(b) and 4(b) show the prediction results in the case that the two endpoint vectors TV_1, TV_2 have the same direction.

4.6. Triangulate the predicted points

In this step, the 2D constrained triangulation method [27] is conducted on the projections of the boundary points and the

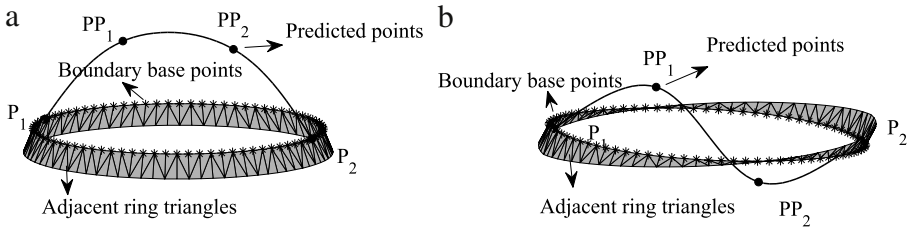

Fig. 4. Two cases of the prediction of two points in original 3D space.

prediction points. When the base points are segmented which is determined in the second step, the prediction points and the base points are projected onto two adjoining planes, which are then unfolded to an unique plane in 3D. The prediction points in Fig. 1(a) are projected onto two planes α and β in Fig. 1(b), which are unfolded onto an unique plane α. When the boundary points are not divided, all the points are directly projected to the fitting plane determined by the PCA method. The plane α along with the corresponding projected points are further rotated onto a plane parallel to the *xy*-plane in 3D space (see Fig. 1(c)). Finally, the constrained triangulation method in 2D space is conducted on all the projection points, the obtained topology of the triangulation mesh is applied to the original 3D prediction points (see Fig. 1(d)).

4.7. Smooth and optimize the local mesh

The last step of the method is to smooth and optimize the obtained triangulation of the predicted points and the boundary points. The obtained triangulation is always smoothed for $NumSmooth = 60$ times for all the test problems in this work. All the smoothed predicted points are denoted as $\{SP_i, i = 1, \ldots, N\}$.

To further improve the triangulation mesh, mesh optimization methods [28,29] including triangle simplification and edge swapping are employed. The simplification operation is to decimate the points being too close to other points, the swapping operation is to decrease those long and narrow triangles.

In Fig. 5, from (a) to (b), the triangles including both points P_1 and P_2 are decimated, the new generated point P_0 adopts the midpoint of P_1 and P_2 as $\frac{1}{2}(P_1 + P_2)$, and the decimation procedure is performed on all the edges with lengths being less than a predetermined length. Fig. 5(c) and (d) illustrate the swapping procedure of the local mesh, this operation is executed as long as the minimum of all the angles included in triangles $\triangle P_1Q_1Q_2$ and $\triangle P_2Q_1Q_2$ is smaller than that in triangles $\triangle P_1P_2Q_1$ and $\triangle P_1P_2Q_2$.

4.8. Execute the algorithm

The time complexity of the algorithm HPDE from step 2 to step 7 are $O(n^2)$, $O(n^2)$, $O(n)$, $O(n \cdot M \cdot NumIter \cdot NumPop)$, $O(n \cdot M \cdot \log(n \cdot M))$, $n \cdot M \cdot (NumSimplify + NumSwap)$, respectively, where n is the number of base points on boundary curve, M is the maximum number of prediction points in the clamping region of each two corresponding points, *NumIter*, *NumPop* are the numbers of iteration and population in the employed DE optimization which are set to be constants, and *NumSimplify*, *NumSwap* are the numbers of operations of simplification and swapping in step 7. Thus, the overall time complexity of the proposed HPDE is $O\{n \cdot M \cdot [NumIter \cdot NumPop + \log(n \cdot M) + NumSimplify + NumSwap] + n^2\}$. In our trials, *NumSimplify* and *NumSwap* are usually the rates of the number of all prediction points and the DE optimization always obtains a convergence solution after only several iterations, that is, $NumSimplify = O(M \cdot n) = NumSwap$ and *NumIter*, *NumPop* are small constants. Consequently, the approximate time complexity of the proposed HPDE on one hole is $O(n^2 \cdot M^2)$.

To avoid searching in the entire points and triangles when finding the holes and adjacent points and triangles, a special data structure is employed to alleviate the burden of the time cost. Each coordinate is partitioned into many small equidistant intervals according to the minimum and maximum values of the coordinate of all the provided points, then the equidistant interval that the coordinate of each point lies in is stored. Based on the stored information, the searching of the adjacent triangles is confined in a small region which is composed of a small proportion of points and triangles, because the points lie in a local region of the considered point share the adjacent coordinate intervals.

To speed up the optimization of DE algorithm, the initial population for optimizing the problem (14) is carefully chosen. The N individuals of the initial population are randomly initialized around the vector $(y_1, \ldots, y_m)$ calculated in the following

$$y_i = L_{i,2} + (U_{i,2} - L_{i,2}) \cdot \frac{i}{m+1}, \quad i = 1, \ldots, m \tag{16}$$

where $L_{i,2}$ is the second coordinate value of the intersection point L_i (see Fig. 3).

In executing the mesh optimization in Section 4.7, the adjacent points, edges and triangles are stored and renewed in each operation, which is adopted as the data structure of the proposed HPDE. The following economical form is adopted for selecting edges to decimate.

1. Sort the lengths of all the edges in ascending order, the sorted edges are $\{e_1, \ldots, e_R\}$. $Id = 1$. $NS = \emptyset$.
2. While the length of the shortest edge is smaller than a rate of the average length of the edges in the adjacent triangles around the hole.
   If *NS* is empty
   Choose the edge e_{Id} for decimation.
   Else
   Choose the shortest edge e_t in the union set of e_{Id} and *NS* for decimation.
   End if
   Renew the adjacent information around the decimated edge and find the shortest edge e_s in the renewed edges. $NS \leftarrow NS - e_t$, $NS \leftarrow NS \bigcup e_s$. $Id \leftarrow Id + 1$;
3. End while iteration.

The selection of the edges for swapping is similar. An intermediate variable $Rate = 1 - \frac{NumSimplify}{M \cdot n/2}$ is employed to denote the rate of the retained triangles after decimation.

In HPDE, several sets of parameters are introduced, which are the parameters N, F, CR in DE algorithm, two critical angles A_c, I_c for modifying projection and correspondence, two numbers m, M for determining the number of prediction points, and two variables *NumSmooth*, *Rate* for improving the obtained triangulation. In all the tests, the adopted setting $N = 3 \cdot m$, $F = 0.8$, $CR = 0.05$ works well for all the test models that DE obtains a convergence solution fast because the initial population is carefully chosen. The parameter A_c is for determining whether the hole region needs to be segmented before processing, the segmentation procedure

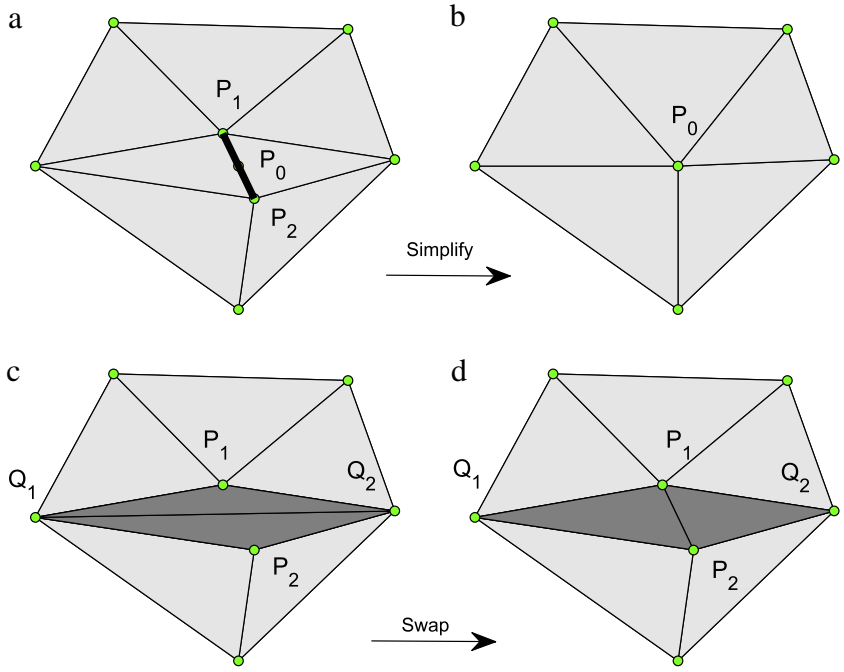

Fig. 5. The operations of triangle simplification and edge swapping.

is adopted to avoid the case that the hole with high curvature is projected onto a plane yielding a wrong topology structure. The setting of the parameter A_c in the region $[\pi/4, \pi/2]$ achieves this purpose in all our trials. The parameter I_c is brought in modifying the correspondences of the base points, which is for avoiding the case that two intersection points of the section plane and the adjacent triangles on one side of the boundary are not inside the same triangle yielding wrong prediction. An angle in the region $[\pi/30, \pi/10]$ is proposed for the setting of this parameter. The parameters n, M, *NumSimplify* are automatically determined by the lengths of edges around the considered hole. The effect of the parameter *NumSmooth* on the performance of the algorithm is studied in Section 5.

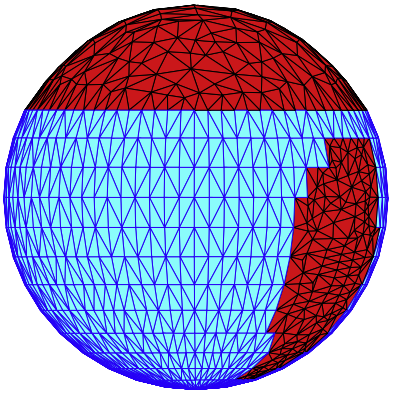

Fig. 6. Hole patching of two large holes on the sphere model.

5. Numerical results

5.1. Test problems and criteria

5.1.1. Test problems and aspects for comparison

The overall algorithm is realized by the popular software MATLAB of version 2009b on a PC with a core processor operating at 2.8 GHZ with 4 GB of RAM. To check the effectiveness of the proposed method, several different kinds of holes are chosen for the numerical test, which are mainly divided into two categories—natural holes and manual holes. The natural holes are extracted from the Stanford bunny model, while the manual holes are derived from the models of an unitary sphere, the elephant, the gargoyle [30] and an unitary cube, which are divided into the kinds of large holes, irregular holes, fringe holes, holes on complex model and holes with high curvature or characteristic line. The numbers of points and triangles in these models are demonstrated in Table 1.

To test the performance of the method, mainly two aspects—the prediction error and runtime (RT) with unit in second (s) are considered. The prediction errors are computed w.r.t. the patching of manual holes, which includes the maximum prediction error (MPE) and the average prediction error (APE). These errors are computed as follows:

$$\begin{cases} MPE = \dfrac{1}{Mx} \max\limits_{1\le i\le N} E_i \\ APE = \dfrac{1}{K \cdot Mx} \sum\limits_{1\le i\le K} E_i \\ Mx = \max\limits_{1\le i\le K,\, 1\le j\le 3} |SP_{i,j}| \end{cases} \tag{17}$$

where K is the number of overall prediction points, E_i is the distance of the ith smoothed predicted point SP_i to the abandoned hole triangles, which is the distance of the point SP_i to the nearest projection point on the piecewise surface composed by the hole triangles. The results about the prediction errors, runtime and some information about the intermediate parameters and the holes are demonstrated in Table 1. The performance of these hole patching are demonstrated in Figs. 6–15.

A further comparison of our method and two recently proposed algorithms [20,21] (we code their algorithms) on holes of the sphere, elephant and horn is conducted, the numerical results of the comparison are presented in Table 2 and the performance of these algorithms on one sphere hole are depicted in Fig. 16. Besides of the comparison with recently proposed algorithms,

Table 1
Some information about the model, the holes and the results of hole filling. The abbreviations NP, NOT are the numbers of points and the overall triangles of the considered model, NHT is the number of the abandoned triangles in the hole region. RT is the abbreviation of the runtime, which is recorded in the unit of second (s). MPE and APE are the maximum and average prediction errors, respectively. The variables M and n are the maximum number of prediction points in each pair of corresponding points and the number of base points, respectively. Rate is the rate of the retained triangles after decimation.

Model	Statement	NP	NOT	NHT	MPE	APE	RT(s)	M	n	Rate
Sphere	2 large holes	1600	3040	1210	1.77E−02	7.35E−03	9.3	10, 7	106, 124	0.73, 0.77
Bunny	4 natural holes	35947	71892	–	–	–	21.8	8, 8, 7, 8	178, 166, 182, 156	0.81, 0.67, 0.69, 0.58
Elephant 1	4 fringe holes	24955	49914	878	8.10E−03	3.01E−03	17.1	9, 8, 9, 7	88, 92, 98, 84	0.67, 0.65, 0.72, 0.60
Elephant 2	4 irregular holes	24955	49914	740	1.01E−02	3.23E−03	15.8	6, 5, 6, 5	142, 138, 150, 146	0.65, 0.66, 0.72, 0.54
Gargoyle	6 holes on complex model	25038	50084	385	8.87E−04	2.98E−04	20.0	8, 8, 7, 6, 7, 7	136, 122, 128, 130, 126, 122	0.88, 0.81, 0.72, 0.66, 0.73, 0.68
Horn	1 hole with high curvature	232	476	43	7.80E−02	3.72E−02	3.6	6	56	0.48
Cube	3 holes with characteristic	1744	5820	1058	1.12E−02	6.37E−03	12.1	14, 14, 8	82, 84, 146	0.62, 0.59, 0.69

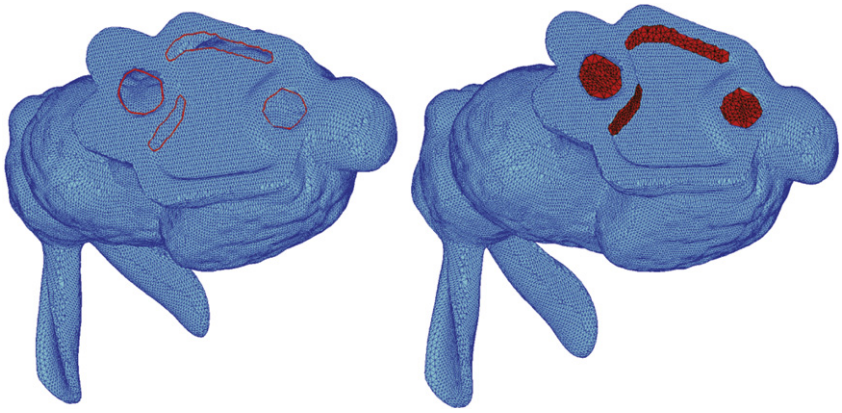

Fig. 7. Hole detection and patching for the bunny model. The left depicts the boundaries of all the holes, the right presents the results about hole patching.

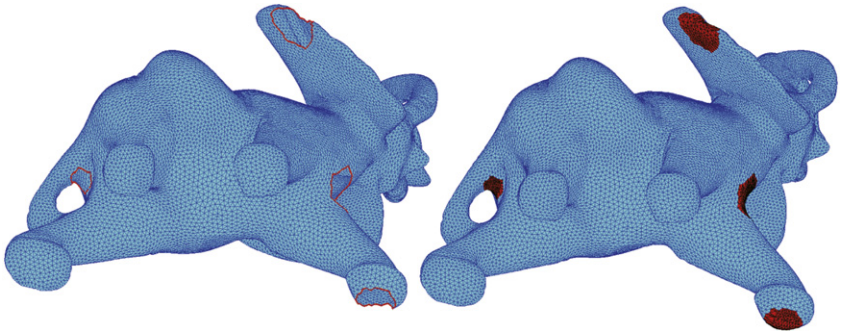

Fig. 8. Patching results of the fringe holes on the elephant model.

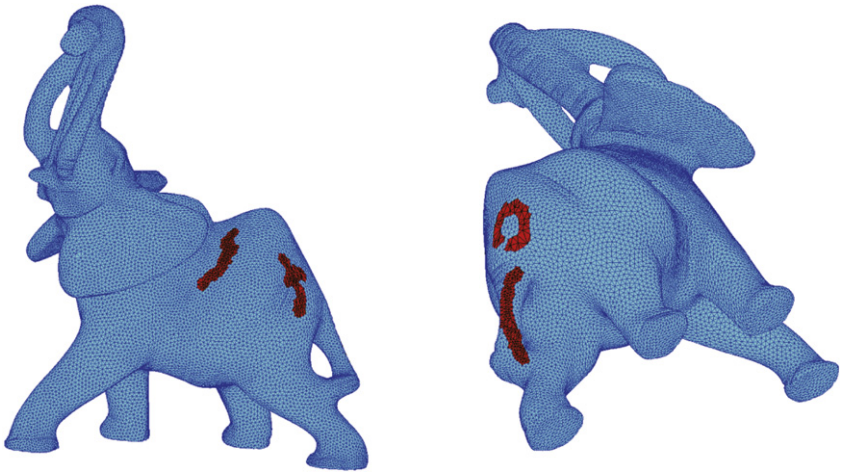

Fig. 9. Patching results of irregular holes on the elephant model.

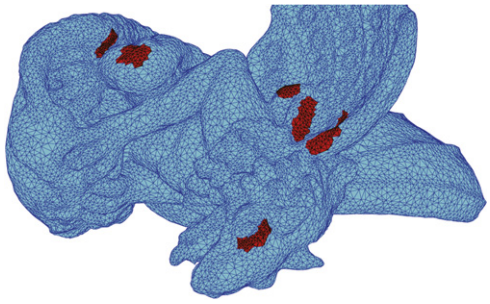

Fig. 10. Results of hole patching for the holes on the gargoyle model.

another two softwares MeshLab and ReMESH are chosen for the comparison. The patching results on one hole of the sphere model are demonstrated in Fig. 17.

5.1.2. Holes with characteristic lines

Furthermore, holes with characteristic lines are patched with the proposed HPDE, the results are demonstrated in Fig. 15. Two models are chosen for this test, one is an unitary cube, the other is an inclined prism generated by translating the up face of the unitary cube a distance of 0.25 in the x axis direction. The manual holes always contain a border line of the considered models.

5.1.3. Robustness and sensitivity

To test the robustness of the method against noise, noises of uniform distribution $[-\sigma, \sigma]$ with different intensities are added into the original data points, where σ is a scale factor that reflects the intensity of the noise, we consider the values of 1%, 3% and 5% in our trials. The perturbed points are first triangulated by the crust algorithm in [6], then the proposed HPDE is employed to patch the abandoned hole region after which the adjacent points around the hole are smoothed by the approach described in the first step of the proposed HPDE. Results of HPDE w.r.t. the holes on the noisy sphere models are depicted in Fig. 18.

To test the sensitive of the smoothing operation to the proposed HPDE, the effect of the parameter *NumSimplify* on the performance of the HPDE is studied in this part. Although the parameters M, n are automatically determined by the average length of the edges around the hole, these parameters determine the number of overall prediction points $M \cdot n/2$ which consequently determine the accuracy and the time complexity of the prediction. The effect of the number $M \cdot n/2$ on the prediction accuracy is also studied, the results of the means and the variances (DE algorithm is stochastic-based) of the prediction accuracy are presented in Fig. 19.

5.2. Illustrations of the results

5.2.1. Results of an overall comparison

Results in Figs. 6–13 illustrate that the proposed HPDE achieves reasonable results for natural holes, large holes, fringe holes, irregular holes and holes on the complex model. Results of the prediction error and the runtime in Table 1 provide further proof of the statement, where the maximum prediction errors for these kinds of holes do not exceed 1.77E−02, while the average prediction errors reach 10^{-3} or 10^{-4} for all the test problems. The runtime for the patching is maintained at reasonable level which is approximately 4 s for each manual hole, which increases to 5 s for each natural holes because the searching of the adjacent triangles and points consumes more time than the former. The

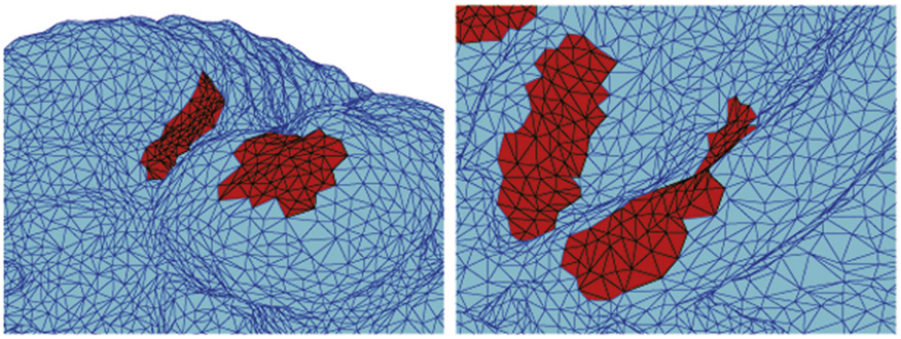

Fig. 11. Patching results on some of the holes of the gargoyle model from different views.

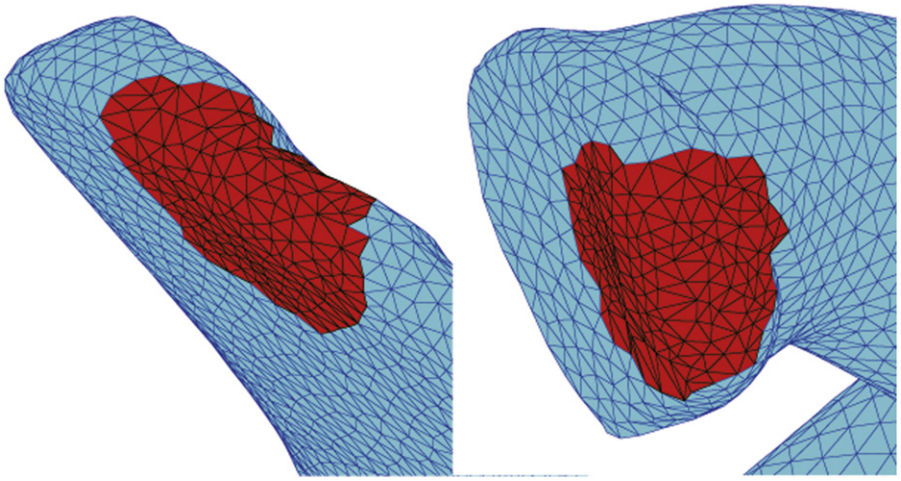

Fig. 12. Patching results of some of the fringe holes on the elephant model from different views.

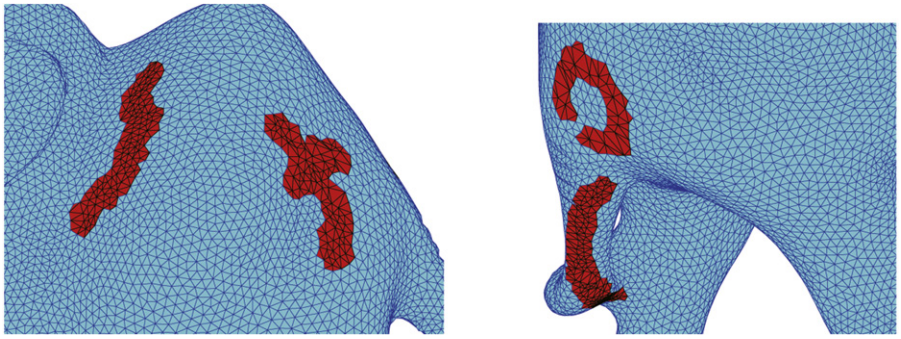

Fig. 13. Patching results of some of the irregular holes on the elephant model from different views.

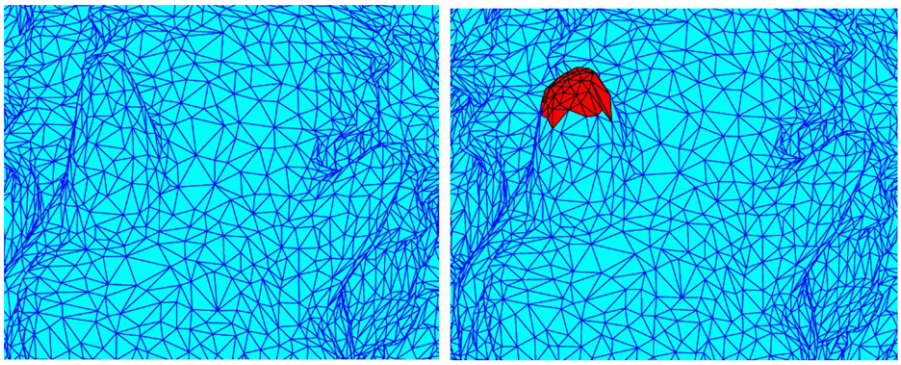

Fig. 14. The patching result of HPDE on the hole with high curvature. The left corresponds to the intact horn on the gargoyle model. The right demonstrates the patching result of one horn on the model.

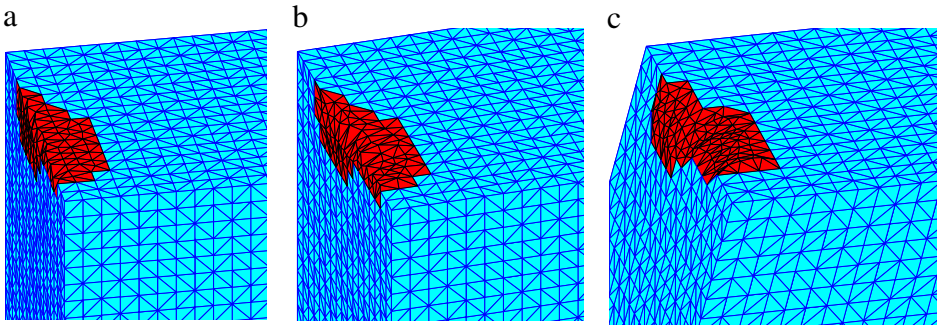

Fig. 15. The patching result of HPDE on the hole with characteristic line. The left corresponds to the patching result of a hole on the cube without smoothing the obtained triangulation. The center corresponds to the result of HPDE on the hole, where the smoothing operator is employed and the base points are segmented. The right corresponds to the patching result of a hole on an inclined cube, where the base points are not segmented.

Table 2
The comparison of the prediction error (APE) and the runtime (RT) of three algorithms on four test models.

Algorithm	Sphere		Elephant 1		Elephant 2		Horn	
	APE	RT	APE	RT	APE	RT	APE	RT
GREY	6.53E−02	2.3	8.76E−03	4.2	3.08E−03	3.9	1.08E−02	1.8
NURBS	8.72E−02	2.7	7.55E−03	6.1	2.94E−03	5.8	6.53E−02	2.1
HPDE	7.35E−03	9.3	3.01E−03	17.1	3.23E−03	15.8	3.72E−02	3.6

holes on the sphere model and the fringe of the elephant model are demonstrated with large curvatures, the proposed HPDE still achieves sound performance in Figs. 6, 8 and 12 and the average prediction errors also achieve 10^{-3} or 10^{-4}. Thus, the proposed HPDE is applicable for patching the holes with large curvature.

5.2.2. Results of high curvature and characteristic lines

From Fig. 3, it can be seen that the prediction points are bounded by the circular arcs determined by the two endpoint vectors. Because of the limitation of this prediction strategy, the proposed HPDE is not suitable to patch holes with very high curvature where the points exceed the bounded region of these circular arcs. The patching result of a hole with high curvature in Fig. 14 illustrates the statement, where the tip of the damaged horn on the gargoyle model is not finely recovered.

Referring to the holes with characteristic lines, an average prediction error of 6.37E−03 is achieved for the patching results in three different situations shown in Fig. 15. Correspond to Fig. 15(a)–(c), the average prediction errors on the holes of a cube and an inclined prism are 4.02E−03, 6.16E−03 and 8.93E−03 by

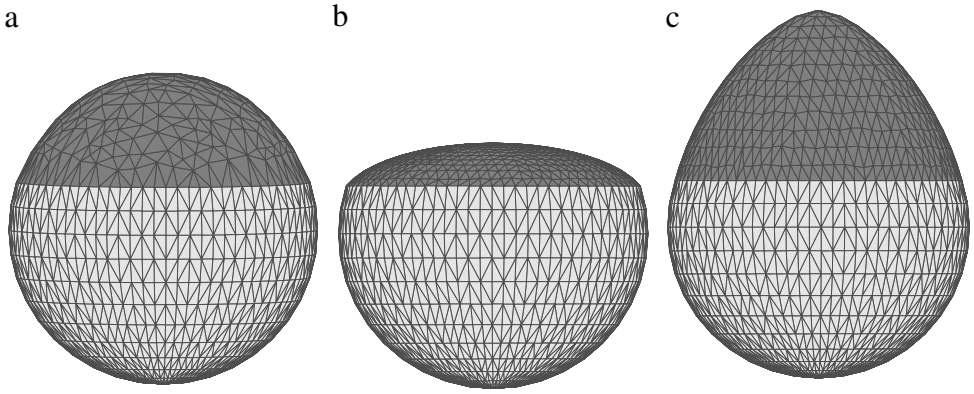

Fig. 16. The patching results of three algorithms on one hole of the sphere model, (a), (b) and (c) correspond to results of algorithms HPDE, NURBS and GREY, respectively.

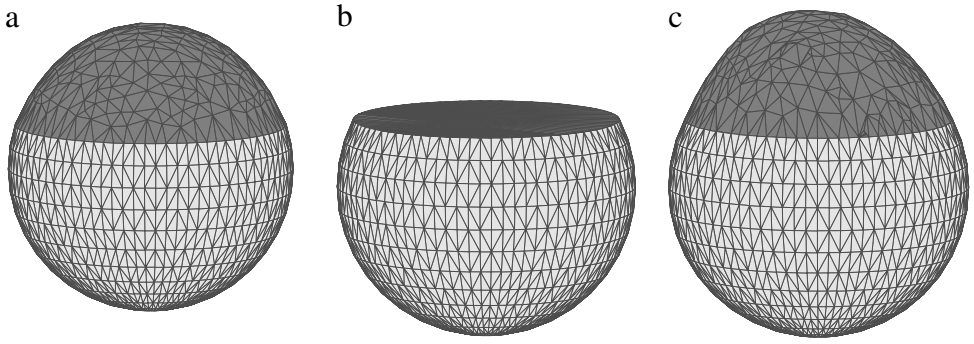

Fig. 17. The patching results of three algorithms on one hole of the sphere model, (a), (b) and (c) correspond to results of algorithm HPDE, and the algorithms in the softwares MeshLab and ReMESH, respectively.

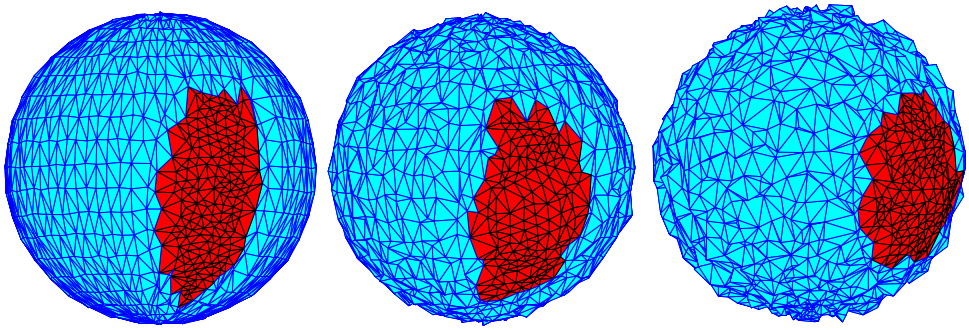

Fig. 18. Hole patching of the holes on the noisy sphere model. From left to right, figures correspond to the performance w.r.t. the intensities of noises of 1%, 3% and 5%, respectively.

the algorithms of HPDE without smoothing operator, HPDE and HPDE, respectively. For patching this kind of holes, the smoothing operator and the operation of segmenting the base points are critical.

For analyzing the function of the smoothing operator, it can be seen from Fig. 15(a) and (b) that the smoothing operator is not beneficial to patch this kind of holes, because the smoothing procedure polishes the characteristic line on the border line of a cube. In other words, the smoothing operator determines that the intersection angles between the prediction points in the hole are uniform and the patching surface is smooth. Thus, this operator is impossible to detect the characteristic lines in the hole which are often with sharp lines or cusp. To analyze the function of the segmentation procedure on the prediction, the differences between Fig. 15(b) and (c) are considered. Notice that the base points in Fig. 15(c) are not segmented because the dihedral angle of the two faces where the hole lies on is larger than $\frac{1}{2}\pi$ ($Ac = \frac{1}{2}\pi$ is adopted in HPDE), while the base points are segmented in Fig. 15(b). It is shown in Fig. 15(b) and (c) that segmenting the base points is beneficial to patch holes containing characteristic lines. By segmenting the base points, the overall prediction is divided into several parts, and the patching surface needed to predict in each part shares more characteristics with the respectively adjacent points and triangles, which yields more accurate prediction results in whole. However, the characteristic in the hole is still not finely recovered because of the smoothing operation (see Fig. 15(b)).

5.2.3. Comparison with methods of state of the art

The results of the comparison in Table 2 illustrate that our algorithm achieves competitive APE on the considered test problems of sphere and elephant1 although more runtime are needed. Concretely, for relatively flat holes (such as the holes on elephant2), the proposed HPDE reaches a prediction accuracy similar to the accuracies obtained by the other two algorithms. For holes with very high curvature (such as the hole on the horn model), HPDE

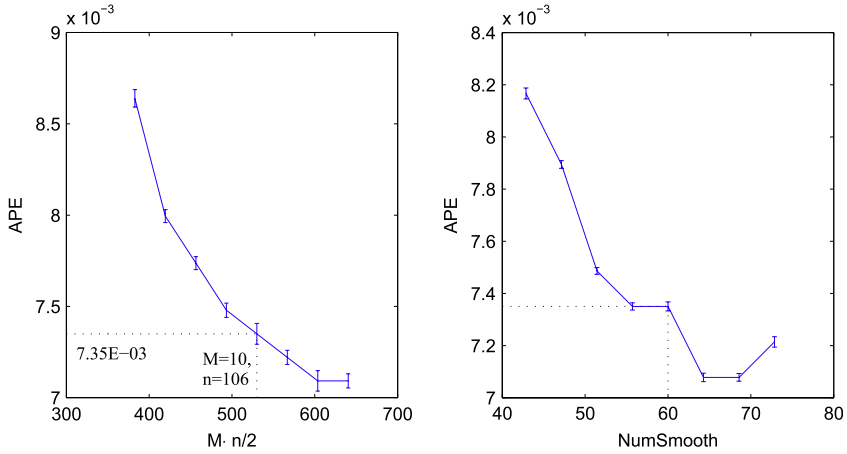

Fig. 19. The results of prediction error (APE) when the number of prediction points $M \cdot n/2$ and the number of smoothing operation *NumSmooth* vary.

achieves a prediction accuracy between the other two algorithms. While for holes with large curvature (such as the holes on the sphere and elephant1), HPDE obtains the highest prediction accuracies compared with the other two algorithms. Notice that the method [21] with non-uniform rational B-spline (NURBS) surface is apt to obtain a flat surface in the hole region because of the characteristic of spline interpolation as illustrated in Section 2, this algorithm obtains the best accuracy for relatively flat holes. While the algorithm [20] with grey prediction (GREY) is apt to maintain the boundary because the point information only on one side of the boundary is used in each point prediction, thus, this algorithm achieves the least prediction errors for holes with very high curvature. The proposed HPDE utilizes the point information on both sides of the boundary by constructing point correspondences, and the prediction points in the hole region are bounded between two circular arcs, therefore, it achieves the least prediction errors for those holes with moderately large curvature and balances the prediction results between holes with small and high curvatures.

Before comparing the time complexities of the three algorithms, assume the number of prediction points is N_p. The algorithm NURBS contains a local triangulation (with time complexity $O(N_p \cdot \log(N_p))$) and an operation of solving $O(N_p)$ sets of linear equations ($O(N_p \cdot N_{ctl})$, N_{ctl} is the time complexity of NURBS interpolation for obtaining the control points by solving a set of linear equations). Algorithm GREY contains a local triangulation and an operation of solving $O(N_p)$ sets of linear equations ($O(N_p \cdot N_{coeff})$, N_{coeff} is the time complexity of grey model for obtaining two coefficients by solving a set of linear equations). While for the proposed HPDE, besides of a constrained triangulation, it contains the DE optimization which is based on multi individuals and multi generations ($O(N_p \cdot NumIter \cdot NumPop)$, refer to Section 4.8), and it includes a procedure of mesh optimization ($O(N_p^2)$). Compared with HPDE, the dimensions of the linear equations in the other two algorithms are relatively small, that is, N_{ctl}(or N_{coeff}) $<$ $NumIter \cdot NumPop$. Moreover, although an economical form for selecting edges to decimate is adopted ($O(N_p^2) = \varepsilon \cdot N_p^2$, where ε is a small number), the procedure of mesh optimization is also time-consuming when N_p is a large number. Thus, the proposed HPDE needs more runtime on the test problems than the other two algorithms.

It can be seen from Table 2 that the proposed HPDE consumes more time than another two considered algorithms on the test problems. Coincide with the above analysis of the time complexities, the runtime costs are mostly concentrated on the DE optimization and the mesh optimization in the executions of HPDE, which account for proportions of 43% and 38% of the overall time costs on average.

Considering the comparison of the results of the proposed HPDE and two softwares, the hole patching algorithms (abbreviated as LMesh and RMesh, respectively) adopted in the softwares of MeshLab and ReMesh come from [18] (without mesh refinement and fairing) and [19], respectively. It can be seen from Fig. 17 that the algorithm LMesh without mesh refinement and fairing patches the holes with a triangulation of a flat plane. The algorithm RMesh patches the holes with a local triangulation matching the boundary of the considered hole, while the sizes of the triangles are not uniform enough and parts of the patching triangulation are not smoothed. It is demonstrated in Fig. 17 that the proposed HPDE achieves competitive result on this hole compared with the patching results obtained by these two softwares.

5.2.4. Results of robustness and sensitivity

Further experimental results about the patching of the holes on noisy model show that the average prediction errors w.r.t. the three abandoned holes are 0.016, 0.039 and 0.056, respectively, which increase with an approximate quantity of 0.01 compared with the intensities of the corresponding added noises. Incorporated with the performance of the hole patching in Fig. 18, it is concluded that HPDE is robust against the low-intensity of noise.

It is shown in Fig. 19 that APE varies in the region of [7.0E−03, 9.0E−03] when $M \cdot n/2$ varies in the region of [350, 650], which illustrates that initially selected number of prediction points $M \cdot n/2$ does not significantly affect the APE on the sphere hole. Thus, appropriate values of the parameters M, n can be flexibly selected to comprehensively consider of the accuracy, the runtime and the consistence of the edges in and around the hole region. Considering of the parameter *NumSimplify*, the prediction error varies in the region of [7.0E−03, 8.4E−03] when *NumSimplify* varies in the region of [40, 75], thus, the prediction accuracy is not significantly affected by the small perturbation of the parameter *NumSimplify*, which implies HPDE is not sensitive to the parameter *NumSimplify*.

6. Discussion

A new hole patching algorithm is proposed in this work, to cover the shortages of the previously related algorithms, the proposed algorithm predicts the points in holes by utilizing the point correspondences of two layers of base points around the holes, and it obtains uniformly distributed points in the hole by solving an optimization problem and employing a mesh simplification. Although good performance of our algorithm is achieved on kinds of holes of the considered models, some more work needs to be continued to further improve the proposed algorithm.

First, the proposed HPDE for hole filling is mainly concentrated on those topologically simple and geometrically complex holes, the prediction strategy needs to be modified for those more

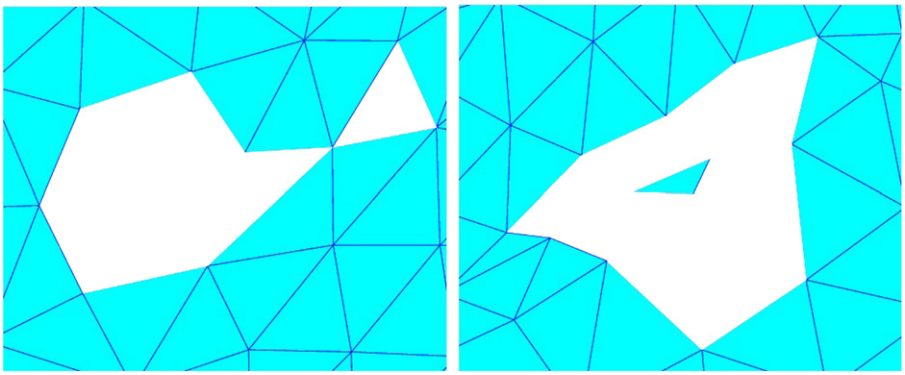

Fig. 20. Two kinds of unusual holes. The left hole is surrounded by not complete adjacent ring triangles. The right hole contains some fragmental information in the center.

topologically complex holes. Meanwhile, because the topology of the prediction points is obtained by the constrained triangulation on a rotated 3D plane, more intermediate 3D planes are needed for the projection of the predicted points when the holes are more complex, an efficient and automatic method to construct these planes needs to be proposed.

Second, the proposed HPDE is suitable to patch holes with large curvature, while it yields inaccurate results for holes with very high curvature because the prediction points are bounded between two circular arcs, for the latter kind of holes, the restricted arcs need to be modified correspondingly.

Third, another aspect to improve HPDE is to detect and utilize some feature lines around the considered hole region for the prediction, and the strategies for selecting the points with uniform distances and rotational angles in the hole region need to be correspondingly modified according to these feature lines. Moreover, the smoothing operator in the proposed HPDE should be adopted selectively, and the base points can be segmented into more parts and projected to more planes to maintain these characteristics.

Fourth, although competitive prediction accuracies of the proposed HPDE on the test problems are confirmed in Section 5, comparisons of time complexities and numerical results in Section 5.2.3 demonstrate that the proposed HPDE consumes more time than another two considered algorithms on the test problems, several measures will be considered to decrease the time cost. From one aspect, the number of iteration *NumIter* in DE algorithm can be reduced by adopting more powerful DE version [25], from another aspect, the number of prediction points in the hole region can be decreased or the number of decimated triangles will be reduced.

Finally, in the practical application, it may exist some more unusual holes, such as shown in Fig. 20. To deal with these cases with the proposed HPDE, some preprocessing operations (similar as in [20]) corresponding to the first case in Fig. 20 are needed and the prediction strategies corresponding to the second case in Fig. 20 should be slightly modified to make use of the fragmental information in the hole.

7. Conclusion

In this work, a new hole patching method HPDE is proposed, which completes the defective triangulation model with set by set prediction points through DE algorithm. Different from other hole patching methods, the proposed HPDE utilizes the point information on both sides of the boundary in each prediction, and it obtains the topological and geometrical structures of the patching mesh in the hole region at the same time, then two approaches of mesh optimization are resorted to improve the quality of local mesh by the locally constrained triangulation. Numerical results on triangulation models with different kinds of holes show the effectiveness of HPDE and that the proposed HPDE is suitable for patching the holes with large curvature. The results of the comparison of HPDE and another two related algorithms demonstrate that HPDE is rather competitive on the considered test models.

Because of the good performance of the proposed HPDE on the considered test models, we expect to apply the method into more fields and more actual problems. Our future work will also concentrate on alleviating and eliminating the defects of the proposed method stated in the discussion section.

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments and suggestions. The authors also thank Prof. Jun Zou of the Department of Mathematics of the Chinese University of Hong Kong for his useful discussions and suggestions. This work was supported by the Major Research Plan, National Natural Science Foundation of China (Grant No. 91230118), the National Natural Science Foundation of China (Grant No. 61173060) and the Key Program of National Natural Science Foundation of China under Grant No. 51039005.

References

[1] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points. In: Proceedings of the 19th annual conference on computer graphics and interactive techniques. Chicago (USA); 1992. p. 71–8.

[2] Krause FL, Fischer A, Gross N, Barhak J. Reconstruction of freeform objects with arbitrary topology using neural networks and subdivision techniques. CIRP Annals-Manufacturing Technology 2003;52(1):125–8.

[3] Zhao HK, Osher S, Fedkiw R. Fast surface reconstruction using the level set method. In: IEEE workshop on variational and level set methods in computer vision. Vancouver (Canada) 2001. p. 194–201.

[4] Xie WC, Zou XF, Yang JD, Yang JB. Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines. Computer-Aided Design 2012;44(11):1127–40.

[5] Lorensen WE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. In: Proceedings of the 14th annual conference on computer graphics and interactive techniques, California (USA); 1987. p. 163–9.

[6] Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th annual conference on computer graphics and interactive techniques. Orlando (USA); 1998. p. 415–21.

[7] Ito Y, Corey Shum P, Shih AM, Soni BK, Nakahashi K. Robust generation of high-quality unstructured meshes on realistic biomedical geometry. International Journal for Numerical Methods in Engineering 2006;65(6):943–73.

[8] Lo SH, Cignoni P, Montani C, Scopigno R. DeWall: a fast divide and conquer Delaunay triangulation algorithm in E^d. Computer-Aided Design 1998;30(5):333–41.

[9] Nooruddin FS, Turk G. Simplification and repair of polygonal models using volumetric techniques. IEEE Transactions on Visualization and Computer Graphics 2003;9(2):191–205.

[10] Davis J, Marschner SR, Garr M, Levoy M. Filling holes in complex surfaces using volumetric diffusion. In: Proceedings of 3DPVT. Padova (Italy); 2002. p. 428–41.

[11] Janaszewski M, Couprie M, Babout L. Hole filling in 3D volumetric objects. Pattern Recognition 2010;43(10):3548–59.
[12] Barequet G, Sharir M. Filling gaps in the boundary of a polyhedron. Computer Aided Geometric Design 1995;12(2):207–29.
[13] Brunton A, Wuhrer S, Shu C, Bose P, Demaine ED. Filling holes in triangular meshes by curve unfolding. In: IEEE international conference on shape modeling and applications. Beijing (China); 2009. p. 66–72.
[14] Branch J, Prieto F, Boulanger P. A hole-filling algorithm for triangular meshes using local Radial Basis Function. In: Proceedings of IMR. Birmingham (USA); 2006. p. 411–31.
[15] Zhao W, Gao S, Lin H. A robust hole-filling algorithm for triangular mesh. The Visual Computer 2007;23(12):987–97.
[16] Kumar A, Shih AM. Hybrid approach for repair of geometry with complex topology. In: Proceedings of the 20th international meshing roundtable. Paris (France); 2012. p. 387–403.
[17] Jun Y. A piecewise hole filling algorithm in reverse engineering. Computer-Aided Design 2005;37(2):263–70.
[18] Liepa P. Filling holes in meshes. In: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on geometry processing. Aachen (Germany); 2003.
[19] Attene M, Falcidieno B. ReMESH: an interactive environment to edit and repair triangle meshes. In: Proceedings of the IEEE international conference on shape modeling and applications. Matsushima (Japan); 2006. p. 41.
[20] Wang LC, Hung YC. Hole filling of triangular mesh segments using systematic grey prediction. Computer-Aided Design 2012;44(12):1182–9.
[21] Kumar A, Ito Y, Yu TY, Ross DH, Shih AM. A novel hole patching algorithm for discrete geometry using non-uniform rational B-spline. International Journal for Numerical Methods in Engineering 2011;87(13):1254–77.
[22] Storn R, Price K. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. In: Technical report TR-95-012, International Computer Science Institute. Berkeley; 1995.
[23] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 1997;11(4):341–59.
[24] Das S, Suganthan PN. Differential evolution: a survey of the state-of-the-art. IEEE Transaction on Evolutionary Computation 2011;15(1):4–31.
[25] Xie WC, Yu W, Zou XF. Diversity-maintained differential evolution embedded with gradient-based local search. Soft Computing 2012; http://link.springer.com/article/10.1007%2Fs00500-012-0962-x.
[26] Taubin G. A signal processing approach to fair surface design. In: Proceedings of the 22nd annual conference on computer graphics and interactive techniques. Los Angeles (USA); 1995. p. 351–358.
[27] Paul Chew L. Constrained Delaunay triangulations. Algorithmica 1989;4(1):97–108.
[28] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Mesh optimization. In: Proceedings of SIGGRAPH 93; 1993. p. 19–26.
[29] Kobbelt L, Campagna S, Seidel HP. A general framework for mesh decimation. In: Proceedings of graphics interface; 1998. p. 43–50.
[30] Available from: http://shapes.aim-at-shape.net/viewmodels.php.