

# A survey on algorithms of hole filling in 3D surface reconstruction

Xiaoyuan Guo<sup>1</sup> · Jun Xiao<sup>1</sup>  · Ying Wang<sup>1</sup>

Published online: 2 September 2016  
© Springer-Verlag Berlin Heidelberg 2016

**Abstract** The surface reconstruction of 3D objects has attracted more and more attention for its widespread application in many areas, such as computer science, cultural heritage restoration, medical facilities, entertainment. However, due to occlusion, reflectance, the scanning angle, raw data preprocessing, it is inevitable to lose some point data, which leads to holes in the reconstruction surface, making it undesirable for various applications. Therefore, methods for filling holes in the process of surface reconstruction are critical to the final results of reconstruction. This paper makes a survey of existing well-known hole-filling algorithms, classifies the algorithms into two main categories, analyzes and compares these algorithms from the viewpoints of theories and experimental results to make a clear introduction of their performance. At the end, the paper points out the possible development direction of hole filling in the future and hopes to be a good guide for other researchers.

**Keywords** Hole filling · Surface reconstruction · Point clouds · Triangle mesh

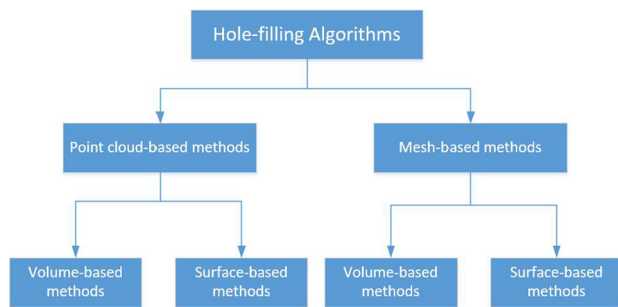
## 1 Introduction

In recent years, a great improvement of 3D laser scanning techniques contributes to the convenient acquirement of 3D point clouds [1]. The angle deviation of scanning, the low reflectance and reclusion of the object surface, the constraints on scanner placement, the occlusion and shade of

other objects lead to partial loss of collected data because of the difficulty of sampling convoluted regions. Additionally, the holes can be caused by missing parts of the original object. Reconstruction based on the scattered data results in incomplete surface with many holes, which would weaken the visual effects of the 3D models and the ability of post-processing for the model. Therefore, hole filling is extremely significant to acquire high-quality 3D model reconstruction and completed geometric information. Many scholars have made relevant research works on filling holes to recover the lost information. And they also proposed several hole-filling methods which generate new points to interpolate the hole region. These methods mainly take use of the information of neighborhood points, the topology, the change of curvature and the distribution of texture of holes. There are mainly two steps to fill holes in 3D reconstruction: extraction of the hole boundaries and hole filling. Many algorithms and techniques have been developed for hole filling and it is a tough task to discuss all of them, so we primarily discuss about the typical algorithms of different classes. Figure 1 shows our classification of current hole filling algorithms. These approaches have proceeded along two main directions: hole filling based on unorganized points, and hole filling based on the mesh. Both of them can be further subdivided into volume-based methods and surface-based methods; according to the data representation to be dealt with, we will discuss corresponding representatives of each category later. The rest of this paper is structured as the following: we first introduce the general process of hole filling (Sect. 2) and analyze the prevailing hole filling methods classified into two main categories specifically (Sects. 3, 4), and then we discuss the improved MeshFix algorithm (Sect. 5), after that we test mainstream algorithms to fill holes using the test data downloaded from <ftp://ftp.cyberware.com> and make a comparison of their performance (Sect. 6); finally, we point out

✉ Jun Xiao  
xiaojun@ucas.ac.cn

<sup>1</sup> School of Engineering Science, University of the Chinese Academy and Sciences, Beijing, China



**Fig. 1** The classification of hole filling methods

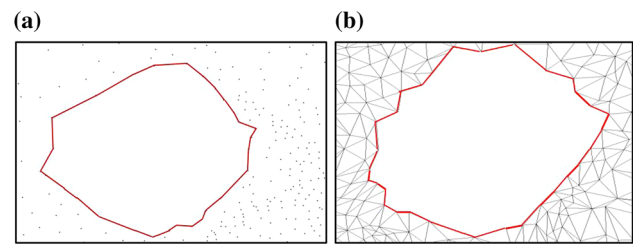
the future development of hole filling in 3D surface reconstruction (Sect. 7).

## 2 Preview

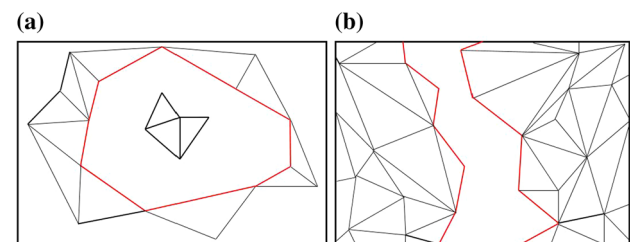
The raw point data acquired are usually unorganized, disordered and noisy [4], so it is necessary to preprocess the raw input to clean the noise, remove outliers, homogenize raw points, organize disordered points, and correct point normals to build a foundation of 3D surface reconstruction. After preprocessing, the scattered point data can be repaired by generating interpolation points in the hole region directly or be repaired by filling polygonal holes after triangulating the unorganized points. Thus, due to the differences of data representation, current hole filling algorithms can be divided into two categories: point cloud-based methods and mesh-based methods. Point cloud-based methods are first studied and there are indeed some valuable theories in them which may enlighten the study of hole filling for future researchers. With the rapid development of Computer Graphics, mesh-based methods have attained more and more attention and played a fundamental role in surface reconstruction. Despite the existence of distinct kinds of meshes, triangle-mesh is the most popular one to represent reconstructed surfaces. Therefore, this paper mainly focuses on triangle-meshed incomplete surface and when referring to mesh-based methods it means triangle-meshed methods.

### 2.1 Typical holes

Holes exist in raw model inputs of both point cloud representation and triangle mesh data representation, thus it is essential to introduce typical holes to make a clear clarification of the problem we intend to settle. For different reasons that lead to the occurrence of various kinds of holes, we can partition them into simple holes and complex holes and complex holes include gaps, islands, the holes which have high curvature and incomplete topology and the holes which have degenerate elements. Simple holes usually mean that holes are disk like and flat compared with its neighbors. Figure



**Fig. 2** a A simple hole in point clouds, b a flat polygonal hole in meshes



**Fig. 3** a An island in meshes, b a gap in meshes

2a shows a typical hole in point clouds, as can be seen, the region enclosed by red lines is blank, missing point data in the region, thus forming a hole loop. Figure 2b shows a flat simple polygonal hole in meshes. In Fig. 3, islands and gaps are shown in (a) and (b); they are more complex to repair than simple holes. Rudolf et al. [2] discuss possible errors in triangular hull geometry representation in the process of fixing mesh.

## 3 Hole filling algorithms based on point clouds

The raw data of scanned objects are stored in the form of points, thus holes in point clouds can be filled directly before these points are triangulated into meshes. Hole filling algorithms based on point clouds can be implemented in two steps, first hole boundary should be detected and then holes are filled in points by generating new points.

### 3.1 Boundary detection of hole

Compared to mesh-based representations, the lack of explicit connectivity information simplifies the definition but overburdens many tasks involved in geometric modeling. Before any attempts to fill holes in point clouds algorithmically, it is obviously required to identify hole boundaries in points. In [3], the boundary probability is computed for every point by applying the angle criterion and then points in the point set are classified into boundary or interior points. At last, by exploiting the coherence of points, a boundary loop is extracted which is a simple hole boundary. For the reason that the point cloud is usually unstructured, it will be helpful to use a kD-tree, octree or BSP-tree to speed up the search for

neighborhoods of a point. Other criteria to determine whether a point is a boundary point or not are half-disc criterion, shape criterion and a few improvements based on these criteria. Besides, features of a point set itself can be extracted to contribute to more piecewise hole filling details for the reconstruction surface. Weber et al. [5] have worked on sharp features detection in point clouds to rebuild complex geometry information in the pipeline of surface reconstruction. Tran et al. [6] extract basic reliable primitives such as planes, spheres, cylinders from unorganized point clouds which have a significant implication to surface reconstruction.

### 3.2 Filling holes

Hole filling methods in point clouds can be subdivided into volume-based methods and surface-based methods for the different data used to fill holes. Unstructured and disordered input points are often inconvenient to be processed directly, so some researchers consider embedding the point set into space grids to find out those irregular volume grids and fill them with new generated points, that is how volume-based methods work.

*Volume-based methods* Volume-based methods of hole filling from point clouds usually decompose the point space into atomic volumes using an octree grid in general. For each volume grid, it is essential to determine the individual volume as inside or outside the point surface and generate signs for the grid which needs to be processed. Besides using an octree grid to produce a hierarchical volume grid, there is an alternative to construct a uniform grid to embed the point space wholly. Franchini et al. [7] segment input point objects into 3D uniform grids to work on large unorganized point sets. The uniform grid approach is simple and straightforward to perform computation of the grid data, yet it requires that the input is uniformly sampled. Different scholars take different measures to fill each irregular volume grid to complete the lost information of the point surface. Xie et al. [4] first organize the input point set by an octree and determine the inside/outside orientation of each point cluster through a robust voting algorithm, which blends the implicit quadric surface of each octree cell to reconstruct the noisy and defective data sets. Based on positive definite radial basis functions, [8] presents a local approach to reconstruct surface from point clouds, progressively filling holes. It builds a uniform grid and employs a multilayer technique to expand the hole boundary information and sets the local nodal function parameters to reconstruct the hole surface. However, this method requires large amount of computation and is sensitive to parameter configurations. The method proposed by [9] discretizes the continuous unsigned field into a volumetric grid to identify the most similar patch candidate to repair holes for both shape and appearance of incomplete points.

Context-based methods can complete both the geometry and texture information only when the input surface itself exists the similar information of the hole area which expects to be recovered. Sharf et al. [10] fill holes with a smooth patch copied from valid regions of a given surface. It partitions the point clouds via constructing an octree, works top down to build local implicit approximation of the shape from the points inside a cell, classifies these cells into valid or invalid cells, and computes patch by copying part of the surface around the filled region by an iterative closest point procedure.

*Surface-based methods* Surface-based methods primarily work with various properties (boundary curvature, topology et al.) of the surface. Wang and Oliveira [11] propose a hole filling algorithm that can recover both geometry and shading information on surface from point clouds based on MLS algorithm (moving least squares). Hoppe et al. [12] use the signed distance field to reconstruct the surface from point clouds. Chalmoviansky and Jttler [13] sample auxiliary points in the neighborhood of holes and reproduce surfaces algebraically, such as planes, cylinders, and spheres to fill holes by approximating the geodesic offsets of the boundary. However, this method can only handle the case of a single hole to patch and is not ideal to patch more complex hole regions. Wu and Chen [14] present a method to fill holes from incomplete point clouds based on boundary extension and boundary convergence. The hole filling procedure is conducted by moving boundary points along the extension direction into the missed regions. Doria and Radke [15] introduce a hole filling technique in LiDAR data sets by inpainting depth gradients to fix the large holes in point clouds of both texture and structure; it reconstructs the 3D scene structure via inpainting methods. Other methods like [16] take use of neighbor information of the hole boundary to fill holes. Though hole filling from incomplete point sets has been discussed earlier than filling polygonal holes in mesh, methods of this kind are less intuitive than the process of repairing holes in mesh surface and not extensible enough to be applied in various surface reconstruction requirements. Therefore, the aforementioned methods are introduced to inspire more productive ideas beneficial to the solution of this issue.

## 4 Hole filling algorithms based on the mesh

Different from the above approaches, the input of mesh-based hole filling methods is a mesh consisting of a vertices set and a triangle set. The methods reproduce the original geometry preferably compared with the hole filling methods from incomplete points directly and they mainly focus on the repairing mesh defects of the input. Similarly, the methods based on mesh are separated into two kinds of categories: volume-based methods and surface-based methods for the

different processes of the data used in the phase of filling holes. Normally, the simple triangulation algorithms work well to triangulate the polygonal hole region, especially for the flat and disk-like holes without intense changes to the hole topology and geometry. For holes that are too complex to fill using triangulation algorithms, researchers have proposed valuable algorithms to solve the hole filling issue of the input mesh, remeshing the hole region and improving the mesh quality smoothly. Volume-based methods fill the polygonal hole via isotropic diffusion of volumetric data; in this way, the topology of the input objects can be recovered more precisely than the surface-based methods, which mostly learn the neighboring surface information of the hole to be filled with new triangles.

#### 4.1 Boundary detection of the hole

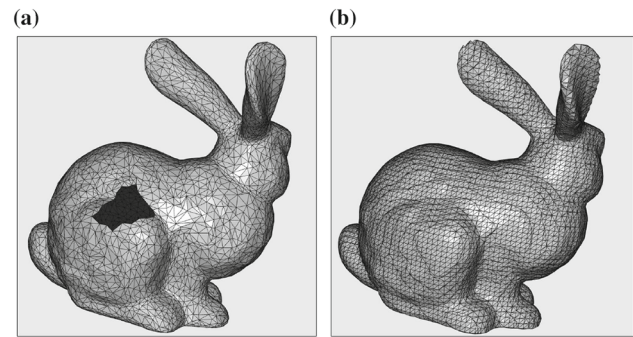
A triangle mesh is composed of a set of vertices and a set of triangles. Normally, an edge is shared by two triangles, called adjacent triangles of the edge. A boundary edge is defined as an edge that is only adjacent to a single triangle. Correspondingly, the boundary edge loop is the representation of a closed hole boundary which can be extracted in the input mesh automatically once a boundary edge is found by tracing its adjacent edges. Jun [17] explains the basic preliminaries of triangular meshes which contain holes. Qiang et al. [18] extract the polygonal hole boundary with Mean Shift approach to find out the vertices that have similar geometry property with the hole region in the neighborhood of the boundary. Clearly, most of the hole filling methods aim to fix simple holes whose boundaries are usually composed of simple closed loops of boundary edges. Besides simple holes, there are many complex holes within islands and gaps that cannot be detected easily by the above-mentioned approach; a detailed description about the defects in the mesh covering complex holes was presented in [19].

#### 4.2 Filling holes

Surface-based methods usually work well on simple holes and fail to recover fine geometry details of large and complex holes. In contrast, volume-based methods can deal with this issue better.

*Volume-based methods* This kind of methods firstly converts input triangular meshes into discrete volume representation, generates inside/outside partitions on the volume and then takes different approaches to finish hole filling works. The methods focus on every single cell and repair those irregular grid cells to reach the final complete triangular mesh surface.

Davis et al. [20] present a typical technique of filling holes by employing the volumetric diffusion procedure to complete the hole surface. It aims to cases where in detail holes are too

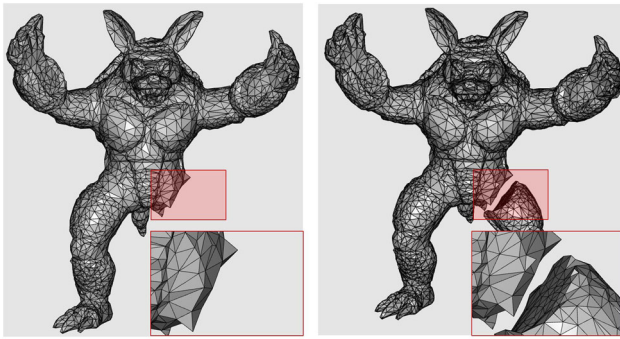


**Fig. 4** The bunny model to be repaired. **a** the original bunny mesh with a simple hole consists of 4964 vertices and 9900 triangle faces, while in **b** the filled result mesh of the algorithm [21] consists of 11,314 vertices and 22,624 triangle faces

geometrically and topologically complex to solve by general surface-based methods. Ju [21] partitions the space into disjoint internal and external volume using an octree grid to repair polygonal model surface by contouring. Though it guarantees a closed output model, it alters the input mesh which may lose sharp features and fine details of the models, as can be seen from the results in Fig. 4; this approach alters the organization of the original input mesh obviously. Guo et al. [22] introduce an oriented voxel global diffusion method to fill holes in complex surfaces. In particular, the diffusion direction of the oriented distance field is controlled accurately inward the hole to restore the sharp features of the incomplete input. While this algorithm is able to fix the incomplete surface smoothly and handle complex holes which contain islands, it consumes large quantity of memory to store these oriented volumetric units such that it lacks the ability to tackle large size input models. Centin et al. [23] employ a Poisson Reconstruction step to obtain an implicit function and then deploy an interpolation method to reconstruct missing parts of the input mesh. A fast and continuous surface-oracle computation exploited to efficiently guide the restricted Delaunay triangulation is one of the crucial elements to perform hole filling works. The biggest difference between this method and others is that it guarantees the full preservation of the input mesh. Figure 10 confirms this point. The main strength of volume-based hole filling methods in meshes is that they are robust to resolve geometric defects and produce a manifold output mesh surface without self-intersections. The major weakness is that in the diffusion phase, geometric details are lost inevitably, even in the remainder areas far from the hole.

*Surface-based methods* Liepa [24] proposes a surface oriented method to smoothly fill holes such that the vertex densities around the holes are interpolated. Jun [17] splits a complex hole into several simple holes and fills each divided simple hole with planar triangulation method consecutively until the entire complex hole is closed. The method of [25]





**Fig. 5** The hole filling results between the original Meshfix [25] (*left*) and modified MeshFix algorithm (*right*) by us. The *left* remains the single biggest component of the input model, while the *right* makes a correct reminder of the whole input

concentrates on converting a low-quality digitized polygon mesh to a single manifold and watertight triangle mesh without degenerating or intersecting elements. It assumes that the resulting mesh should be a single connected manifold bounding a polyhedron and the sampling density should not vary significantly from one part of the mesh to another. Its inherited characteristics do not guarantee that the filled surface will converge eventually, which is the main drawbacks of this method. To improve the performance of the algorithm to process more complex holes such as gaps, we utilize the coherence of the model topology to make a correct decision when deciding which small disconnected components should be discarded and thus develop a modified MeshFix algorithm which will be discussed later. MeshFix is developed by Marco Attene and the latest version is available on <http://meshrepair.org/>. Figure 5 shows the hole filling results of the original input mesh Armadillo, whose holes are manually generated. As we can see, according to the coherence of the whole model topology, the smaller component should be remained rather than discarded directly. Qiang et al. [18] create new triangles by the constraint Delaunay triangulation method to fill holes. Similarly [26], generate a minimal triangulation to the hole. To assort with the density of neighbor meshes, the new triangles are subdivided according to their edge length. The defect of the method is obvious for the filling results depending heavily on the geometry of the hole neighbors. Harary et al. [27] introduce a hole filling algorithm for a given triangle mesh to synthesize missing geometry by learning the knowledge of the similar geometry in the remainder mesh. Different from other context-based approaches, it completes the hole field coherently to the local neighborhood while maintaining a proper triangulation of the mesh surface. Surface reconstruction via an implicit surface representation based on radial basis functions (RBF) [28] and moving least squares [29] have been used widely to deal with large scattered data sets. Without the requisite of prior knowledge of the object topology, the methods construct the

implicit signed distance function constrained by the exterior points near the hole to approximate the surface and interpolate new points into hole regions. Furthermore, exploiting the additional information such as global symmetry [27], an image [30] facilitates the hole filling process efficiently and precisely. The approach of [31] requires a priori knowledge of the input model and it mainly focuses on the hole filling problem of the CAD model, so this algorithm cannot be applied to other common inputs. To summarize, the benefits of context-based approaches are that they learn the characteristics of the given surface making them able to restore lost geometric features without affecting the region mesh far from the hole. The drawbacks of these approaches are their deficiency of robustness in the case of noisy inputs and difficulty in preventing geometric overlaps. Also if there are no similar geometric patches in the meshes, the repaired surface is likely to be unreasonable.

## 5 The procedure of the modified MeshFix algorithm

Since the original MeshFix algorithm first converts the input to a single combinatorial manifold, the fixed results of gaps are not desired, especially for the circumstance that the input model is split into several connected components whose sizes are similar that cannot simply remove all the components except the largest one. The underlying reason of the incorrect topology lies in the phase of topology reconstruction which we mainly focus on to improve the algorithm performance.

---

**Algorithm 1** Main steps of the topology reconstruction phase.

---

**Require:** An indexed face set  $F$   
**Ensure:** A single combinatorial manifold  $M$

- 1: Triangulate all the faces in  $F$
- 2: Initialize  $M$  with the resulting triangles
- 3: Compute the triangle-triangle adjacency relations
- 4: Remove singularities
- 5: Remove all the connected components but the largest one
- 6: Orient the mesh
- 7: **if** cuts were necessary **then**
- 8: warn the user and terminate
- 9: **else**
- 10: patch mesh holes with new triangles
- 11: **end if**

---

We optimize the step of 5 by adding a step to learn the relationships between these connected components to make a judgement about whether it is right to discard the irrelevant components in the hole filling procedure. To achieve this, we first compute the number of vertices and triangles of every single part and make a direct comparison of the number computed to find out the smallest piece component. Then, according to the change of the curvature and of the orienta-

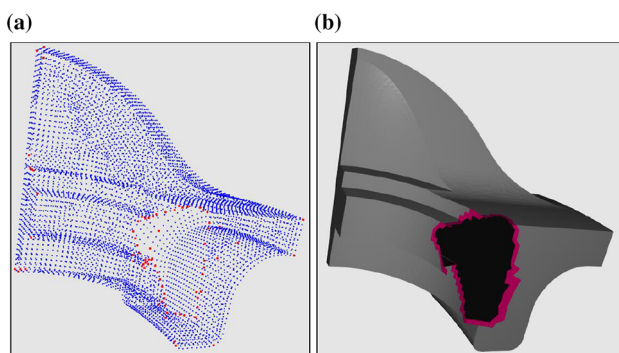
tion of the face normals, we can decide the uniformity of the several components to keep the coherent components. If the change is too obvious to be minimized by adding some triangle faces between them to keep harmony, the incorporate component should be moved away and vice versa.

## 6 Experimental results and comparisons

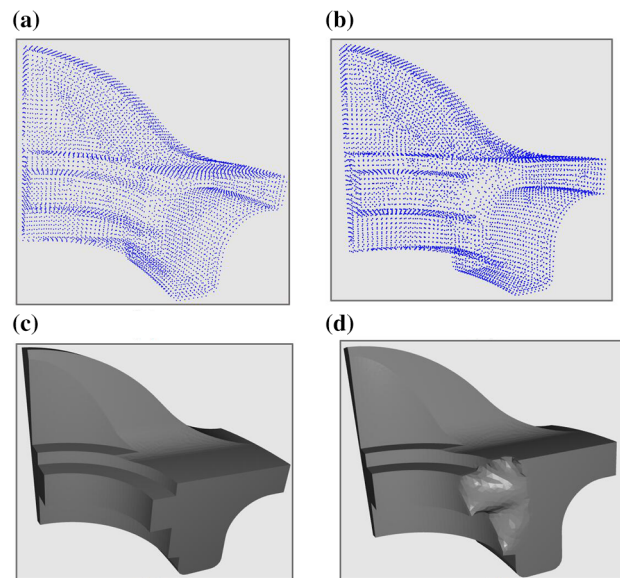
Due to the differences between point cloud-based methods and mesh-based methods to fill holes, it is critical to make a thorough evaluation of them to clarify the characteristics of the distinct hole-filling strategies.

### 6.1 Comparison of point cloud-based methods and mesh-based methods

As has been mentioned before, both the point cloud-based methods and the mesh-based methods should first detect the hole boundaries and then fix the missing regions accordingly. Boundary detection: Experiments have been made to extract boundaries of the bunny model based on these two methods as illustrated in Fig. 6. In point clouds, it is non-trivial to search for neighbor points and to mark boundary points, because the density varies in the distribution of points and a threshold is required to identify a boundary point. Though many criteria can be applied to extract the boundary, they are all sensitive to the threshold and the density, causing redundant boundary points or unwanted points added in the neighboring region. As can be seen in Fig. 6 (a), there are some red points in the sharp corners of the fan disk model that represent wrong judgements of the boundary points, (b) represents the hole boundary in the mesh and the red triangles are the neighbors of the boundary edges; clearly, the boundaries extracted in the mesh are more accurate. This is primarily because of the fact that it is more convenient and straightforward to detect hole boundary by judging whether an edge has two adjacent triangle faces in mesh. In contrast, boundary detection of the point clouds is sensitive to the density of the point set and



**Fig. 6** **a** Boundary detection of the hole in point clouds, the red points are boundary points. **b** Boundary detection of the hole in the mesh



**Fig. 7** **a** The original point set of fan disk model. **b** The hole filling result of fan disk model in point cloud. **c** The original fan disk model mesh. **d** The hole filling result of the hole in the mesh

the boundary threshold, thus costing much time to achieve the best algorithm effects.

*Hole filling* In Fig. 7, hole filling results are presented in both the point cloud representation and in the mesh representation. Figure 7a, c shows the original model input data. Comparatively, hole filling in point clouds is more slowly than the filling procedure of the mesh repair work because of the frequency of computing and querying the neighbor points information of the boundary points and other mistaken points, which increases the cost of time significantly. However, the two kinds of methods fail to recover the original sharp corner of the model.

### 6.2 Comparison of several mesh-based methods

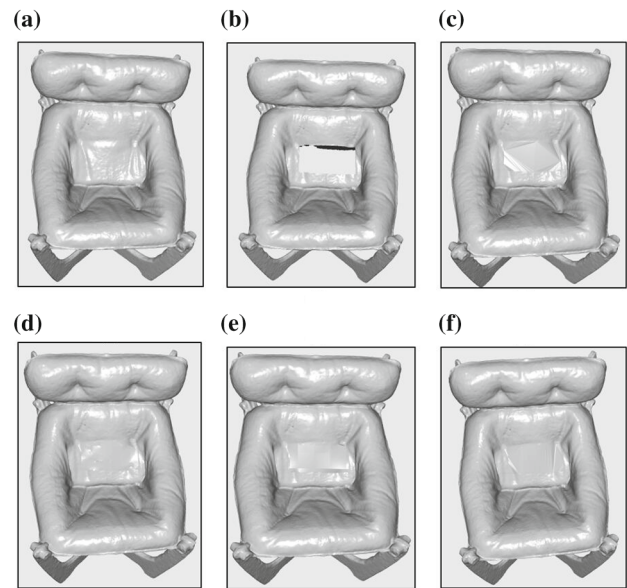
Most of the hole filling methods make an assumption that the holes are relatively small in size and flat in shape; disk-shaped holes are the very common simple holes to be filled. In contrast, complex holes such as high curvature hole regions, gaps between two disconnected components of the mesh, and islands in the middle of holes are more sophisticated to handle. The repairing algorithms we mainly discussed are [20, 21, 23–25]. Quantitative data analysis is mentioned first, and then the final effects of fixing test inputs are shown in the following figures to make a comparison qualitatively among the hole filling approaches. The performance of hole filling methods can be estimated intuitively to observe the density and shape difference between the filled regions and their neighbor meshes.

**Table 1** The test results of the five algorithms

Models	Armadillo		Buddha		Hand		Venus		Chair	
	Vertices	Faces	Vertices	Faces	Vertices	Faces	Vertices	Faces	Vertices	Faces
Original models	5000	9996	20,000	40412	10,000	19,881	134,345	268,686	1,532,312	3,064,462
Models with holes	4908	9695	19,861	40,002	9984	19,812	134,289	265,097	1,473,693	2,945,682
JuTao [21]	131,158	262,312	426,828	853,676	106,128	212,256	742,714	1,485,424	669,450	1,338,912
MeshFix [25]	4175	8342	19,367	38,838	10,773	21,542	133,070	26,613	1,517,126	3,034,264
Liepa [24]	4908	9804	19,861	40,133	9810	19,589	–	–	1,473,693	2,947,342
Ramesh [23]	5003	10,002	19,496	39,024	11,041	22,078	142,571	284,600	1,524,661	3,049,334
VolFill [20]	886,763	1,771,540	736,810	1,471,682	691,585	1,380,708	8549	17,094	–	–

To compare the results, we follow the common paradigm in [17, 21, 22, 26] to count the number of vertices and faces of the model before and after fixing as an important index to measure the performance of listed algorithms. Table 1 presents several representative experimental results of hole filling for several 3D cases in the form of vertices number and face number to explain the main functionalities of the hole filling methods. The model is tested on the five algorithms and it is clear to compare the filled results quantitatively. The content of the box which is a line stands for the inability to fill the holes of the input mesh. In the table, the first line is the model names, and vertices and faces represent for the vertices of the mesh and the faces consisted in the corresponding mesh, respectively. As we can see clearly, compared with the original model meshes, numbers of vertices and faces of the repaired mesh change a lot. It means that the volume-based methods [20, 21] alter the position organization of original mesh points completely though their repair results reserve better topology information than other approaches. Because [24] fails to handle the input data which is large in size and contains many holes, so we leave it blank in the table. Also the algorithm of [20] fails to fix the hole which is relatively big in size and the input data exceed its ability to convert the mesh into volume, which can be proved in the last box with no data. To measure and compare the performance of different algorithms on real-world data, an office chair is scanned and preprocessed with software KSCAN3D. As shown in Fig. 8, the amount of data is considerably greater compared to previous models. During test, the runtime of nearly all the algorithms increased. Figure 8c does not add new points to fill the blank region, instead it constructs the connection relationships of the boundary points and fills the region with new generated triangle faces. Therefore, for the hole boundary long enough the algorithm will have an incompatible visual effect with the original mesh. The repaired result of f is congruous with the whole mesh organization. In this experiment, [20] fails to complete this hole filling work.

The following figures provide some comparison of the repair results, giving a visual representation of the results which are



**Fig. 8** The original data of chair with artificially introduced holes. **a** The original chair scanned by KSCAN3D made of 1,532,312 vertices and 3,064,462 triangles. **b** The hole model of chair. **c** The hole filling result of Liepa [24]. **d** The hole filling result of MeshFix [25]. **e** The hole filling result of JuTao [21]. **f** The hole filling result of Ramesh [23]

arranged in order. We test the methods with different kinds of data containing various holes (simple and flat holes, gaps, islands, complex holes with high curvature and incomplete topology).

*Simple and flat holes* Figure 9 shows comparative results of the different hole filling methods, all of the methods fix the hole in the bunny model completely. However, the triangle meshes of (c) and (e) are inappropriate in size and the curvature is recovered stiffly. Relatively speaking, (d) and (f) handle the circumstance well. Remarkably, [20] fix small holes perfectly with the excellent topology and geometry reconstruction information, which may leave the large holes unclosed.

*Gaps* A gap exists between the body and leg of the Armadillo model as shown in Fig. 10, (g) is unable to heal the hole and



(c) loses the leg of the input model. The effect of (f) is the best one who bridges the gap correctly while others fail to finish the bridge work.

**Islands** Figure 11 shows the repaired results of Venus, (e) is the result of [24], and it fails to finish the whole hole repair work because of the large input size. There are clear volume grids that can be seen in (g) while the algorithm outputs correct topology information. It should be mentioned that the grid size is decided by the parameter predefined.

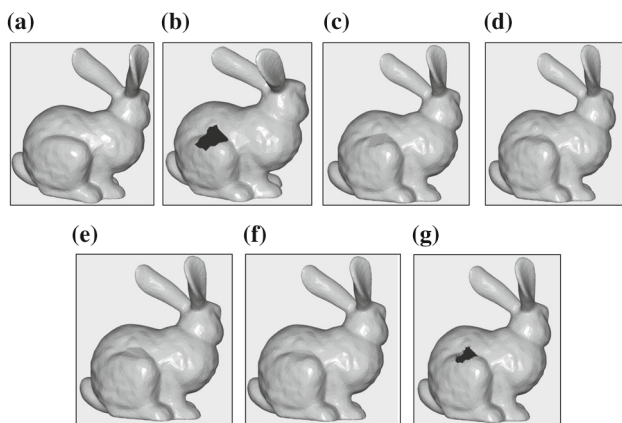
**Complex holes with high curvature and incomplete topology.** Figure 12 shows the behavior of the five algorithms to fill holes of the Buddha model; in the input mesh, there exist two big holes with the first one at the creases of its clothes and the other one at the bottom of the disc platform. The first one is a complex hole whose curvature varies much and, thus, when recovering the lost information of the hole it is important to reconstruct the curve of the creases of the clothes.

The first hole is difficulty to be filled because the curvature around the hole region varies dramatically while the other

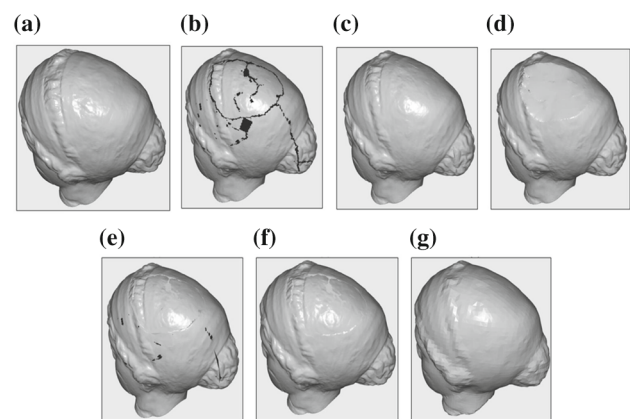
one focuses on the reconstruction of the topology of the disk platform rather than triangulating the hole directly. As can be seen in the figure, as for the first hole which occurs on the robe, (c) and (g) have recovered the curvature of the hole successfully. However, none of the algorithms could repair the disk topology of the pedestal, chiefly because they fail to learn the topology information of the input context. An island hole shown in Fig. 11 is filled, (c) removes the island mistakenly that the algorithm yields an imperfect surface patch to fill the surface holes.

**Complex holes with degenerate elements** Figure 13 provides a comparison of different approaches to fill the big hand model, in (e), the size of interpolating triangle faces varies from the original size of the hole neighbor surface. (d) and (f) manage to reconstruct the structure of quality of the transition between the input and completion parts, in (g), since the method relies on local morphological operations, large holes remain unfilled.

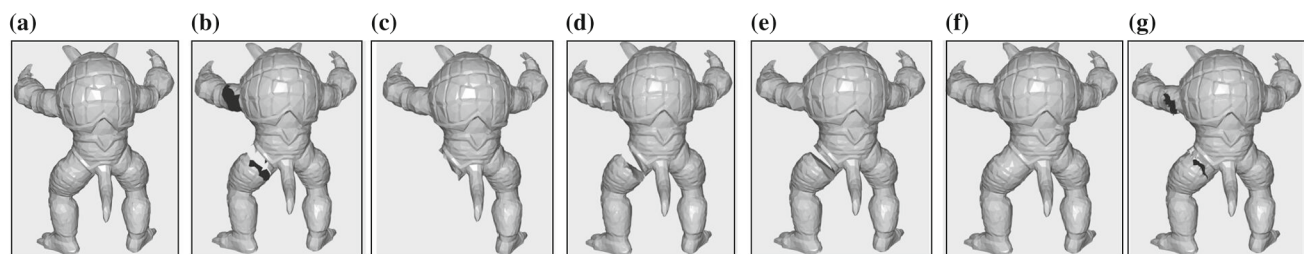
Volume-based methods of hole filling in point clouds generally interpolate new points in the irregular volume grid



**Fig. 9** The bunny model is same as Fig. 4, with a simple hole to be filled. **a** The original complete Bunny model. **b** The original Bunny model with a simple hole. **c** The hole filled with JuTao [21]. **d** The hole filled with the procedure of MeshFix [25]. **e** The hole filled with the procedure of Liepa [24]. **f** The hole filled with the procedure of Ramesh [23]. **g** The hole filled with the procedure of VolFill [20]



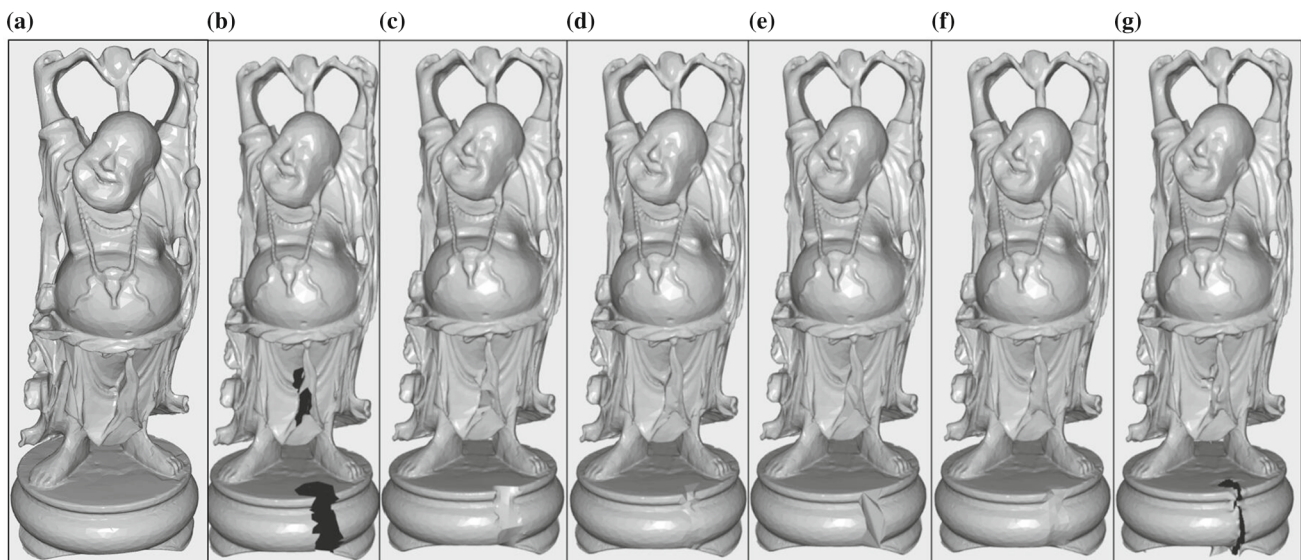
**Fig. 11** The Venus model **a** the original Venus model. **b** The original Venus model with holes. **c** The hole filled with JuTao [21]. **d** The hole filled with the procedure of MeshFix [25]. **e** The hole filled with the procedure of Liepa [24]. **f** The hole filled with the procedure of Ramesh [23]. **g** The hole filled with the procedure of VolFill [20]



**Fig. 10** The original data of Armadillo with artificially introduced holes. **a** The original Armadillo model. **b** The original hole model made of 4908 vertices and 9695 triangles. **c** The hole filling result of MeshFix [25] lost a part of the disjoint component of the original input. Both **d** The hole filling result of JuTao [21] and **e** The hole filling result of [24]

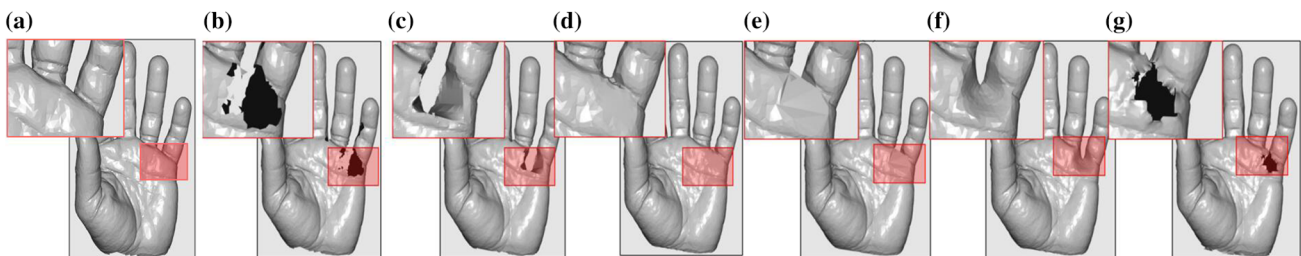
cannot fill the above complex hole exactly; the gap should be bridged which has done perfectly by **f** The hole filling result of Ramesh [23]. **g** The hole filling result of VolFill [20], with the large hole unhealed completely





**Fig. 12** The Buddha model **a** the original complete Buddha model. **b** The original Buddha model with holes in it. **c** The hole filled with JuTao [21]. **d** The hole filled with the procedure of MeshFix [25].

**e** The hole filled with the procedure of Liepa [24]. **f** The hole filled with the procedure of Ramesh [23]. **g** The hole filled with the procedure of VolFill [20]



**Fig. 13** The Hand model **a** the original complete Hand model. **b** The original Hand model with holes. **c** The hole filled with JuTao [21]. **d** The hole filled with the procedure of MeshFix [25]. **e** The hole filled

with the procedure of Liepa [24]. **f** The hole filled with the procedure of Ramesh [23]. **g** The hole filled with the procedure of VolFill [20]

to fill holes, while volume-based methods of hole filling in mesh sandwich the input mesh in the space and move vertices along the direction of healing the hole and generate new triangle faces to fill holes. In the filling process, building a local implicit surface representation based on radial basis functions (RBF) or moving least squares (MLS) to interpolate points or faces smoothly is a popular manner to handle the reconstruction of the hole surface. This kind of methods has the advantage that an analysis of topology in each volume grid facilitates the repair work of topology. Contrarily, due to the completion performed at a global scale, the main weakness of the volume-based methods is that they alter the original point density of the input or the original vertices positions of the mesh, so they cannot preserve the geometry information of the input model. Therefore, sharp features and fine details of the object are lost inevitably so that the methods are unsuitable for some application cases, for example the precise model to be repaired. However, surface-based methods of both hole filling methods in point clouds and in the mesh only work on some specified surface portions, with-

out alteration of the input. Hence, they fix geometric flaws effectively. Surface-based methods learn the neighbor surface information of the hole and utilize the information to generate parameters of the implicit function or the feature signs to help the recovery of the geometry. The drawback is that they ignore the topology information of the whole and local input, thus they fail to fill holes with correct topology.

## 7 Conclusions

This paper has analyzed the hole filling techniques in surface reconstruction quantitatively and qualitatively and provided some comparisons of these approaches. To summarize, each of the various methods aims to solve special applications such as the recovery of the high curvature, the geometry information, the density consistence of the input points or mesh, the preservation of the original topology and the texture of the input models. For most of the methods, assume that the missing surface is generally smooth, scholars have been devoting

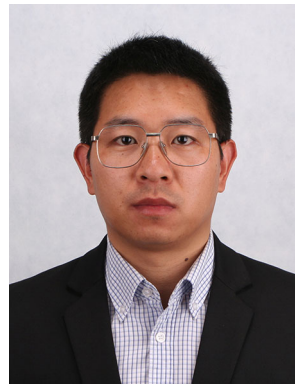
to find out a simple and effective solution to fill holes whether in point clouds or in the mesh. In point clouds, hole filling methods can reconstruct the loss surface smoothly and uniformly; for the hole with high curvature variation, interpolating points directly in the point set is a convenient way to fill holes; a smooth procedure to adjust the position of points using an implicit function is employed afterwards, promising a better visual effect. Unfortunately, these methods are sensitive to the density of the point set. In the mesh, it is crucial to triangulate the hole region by means of volume-based means or surface-based means. Most of the operations in the filling process are the treatment of triangle faces in the mesh, a suitable triangle is added in the mesh data by judging its size, position, orientation. Using the normal triangle face, it is easy to detect sharp features of the model and topology features. Furthermore, the knowledge of Computer Graphics benefits hole repair works in mesh, that is why hole filling in the mesh develops rapidly in recent years. Most of these methods work well for simple holes in nearly flat surface, but not for complicated geometric holes. Thus, it will be valuable to make an analysis of the input model in topology and geometry way before filling holes directly in the input, extracting primitives of the local and global contexts as a guide to recover the topology of the loss region. Combined with the probability of the extracted features, the technique is more likely to make a correct decision on how to repair the hole area, reducing the occurrence of discarding disconnected components and unbridged gaps which should be closed. A step of learning the characteristics of the given surface by calculating the orientation will be conducive to repairing the hole in a feature-sensitive manner. Also there are a few limitations in our work. Though many model data have been tested to evaluate the mainstream hole filling methods, there are probably typical circumstances which are ignored unconsciously. Because of the limitation of time and energy, we have not done complete research test of the hole filling techniques in point clouds deeply. In Sect. 4, we make an improvement of MeshFix method and further improve the algorithm to obtain more excellent performance.

**Acknowledgements** This work is supported by the National Natural Science Foundation of China (No. 61471338), President Fund of UCAS, Youth Innovation Promotion Association CAS (2015361).

## References

- Kirkvik, A.S.: Completing a model based on laser scan generated point cloud data. Institutt for Datateknikk Og Informasjonsvitenskap (2010)
- Rudolf, F., Weinbub, J., Rupp, K., Resutik, P., Morhammer, A., Selberherr, S.: Free open source mesh healing for TCAD device simulations. In: Lirkov, I., Margenov, S. D., Waśniewski, J. (eds.) Large-Scale Scientific Computing, vol.9374, pp. 293–300 Springer, Switzerland (2015). doi:10.1007/978-3-319-26520-9\_32
- Bendels, Gerhard H., Schnabel, R., Klein, R.: Detecting holes in point set surfaces. *J. Wscg* **14** (2006)
- Xie, H., McDonnell, T., Qin, H.: Surface reconstruction of noisy and defective data sets. In *Visualization*, IEEE, pp. 259–266 (2004)
- Weber, C., Hahmann, S., Hagen, H.: Sharp feature detection in point clouds. In: *Proceedings of the 2010 Shape Modeling International Conference*, pp. 175–186. IEEE Computer Society, Washington, DC (2010)
- Tran, T.T., Cao, V.T., Laurendeau, D.: Extraction of reliable primitives from unorganized point clouds. *3D Res.* **6**(4), 1–12 (2015)
- Franchini, E., Morigi, S., Sgallari, F.: Implicit shape reconstruction of unorganized points using PDE-based deformable 3D manifolds. *Numer. Math. Theory Methods Appl.* **3**(4), 405–430 (2010)
- Casciola, G., Lazzaro, D., Montefusco, L.B., Morigi, S.: Fast surface reconstruction and hole filling using positive definite radial basis functions. *Numer. Algorithms* **39**(1–3), 289–305 (2005)
- Park, S., Guo, X., Shin, H., Qin, H.: Surface completion for shape and appearance. *Vis. Comput.* **22**(3), 168–180 (2006)
- Sharf, A., Alexa, M., Cohen-Or, D.: Context-based surface completion. *ACM Trans. Graph. (TOG)* **23**(3), 878–887 (2004)
- Wang, J., Oliveira, M.: A hole-filling strategy for reconstruction of smooth surfaces in range images. In: *Computer Graphics and Image Processing. SIBGRAPI 2003. XVI Brazilian Symposium*, pp. 11–18. IEEE (2003)
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points, vol. 26, No. 2, pp. 71–78. ACM (1992)
- Chalmoviansky, P., Jttler, B.: Filling holes in point clouds. In: *Mathematics of Surfaces*, pp. 196–212. Springer, Berlin, Heidelberg (2003)
- Wu, X., Chen, W.: A scattered point set hole-filling method based on boundary extension and convergence. In: *Intelligent Control and Automation (WCICA), 2014 11th World Congress on* (pp. 5329–5334). IEEE.(2014)
- Doria, D., Radke, R.J.: Filling large holes in lidar data by inpainting depth gradients. In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 65–72. IEEE (2012)
- Yang, N.E., Kim, Y.G., Park, R.H.: Depth hole filling using the depth distribution of neighboring regions of depth holes in the Kinect sensor. In: *Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE International Conference*, pp. 658–661. IEEE (2012)
- Jun, Y.: A piecewise hole filling algorithm in reverse engineering. *Comput. Aided Des.* **37**(2), 263–270 (2005)
- Qiang, H., Shusheng, Z., Xiaoliang, B., et al.: Hole filling based on local surface approximation. In: *Computer Application and System Modeling (ICCSM), International Conference*, vol. 3, pp. V3-242–V3-245. IEEE (2010)
- Attene, M., Campen, M., Kobbelt, L.: Polygon mesh repairing: an application perspective. *ACM Comput. Surv. (CSUR)* **45**(2), 15 (2013)
- Davis, J., Marschner, S.R., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: *3D Data Processing Visualization and Transmission. Proceedings. IEEE. First International Symposium*, pp. 428–441 (2002)
- Ju, T.: Robust repair of polygonal models. In: *ACM Transactions on Graphics (TOG)*, vol. 23, No. 3, pp. 888–895 (2004)
- Guo, T.Q., Li, J.J., Weng, J.G., Zhuang, Y.T.: Filling holes in complex surfaces using oriented voxel diffusion. In: *Machine Learning and Cybernetics, 2006 International Conference*, pp. 4370–4375. IEEE (2006)

23. Centin, Marco, Pezzotti, Nicola, Signoroni, Alberto: Poisson-driven seamless completion of triangular meshes. *Comput. Aided Geom. Des.* **35**, 42–55 (2015)
24. Liepa, P.: Filling holes in meshes. In: *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 200–205. Eurographics Association (2003)
25. Attene, M.: A lightweight approach to repairing digitized polygon meshes. *Vis. Comput.* **26**(11), 1393–1406 (2010)
26. Wei, M., Wu, J., Pang, M.: An integrated approach to filling holes in meshes. In: *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference*, vol. 3, pp. 306–310. IEEE (2010)
27. Harary, G., Ayellet T., Grinspun, E.: Context-based coherent surface completion. *ACM Trans. Graph. (TOG)* **33**(1), 5 (2014)
28. Branch, J., Prieto, F., Boulanger, P.: Automatic hole-filling of triangular meshes using local radial basis function. In: *3D Data Processing, Visualization, and Transmission, Third International Symposium*, pp. 727–734. IEEE (2006)
29. Tekumalla, L.S., Elaine, C.: Reverse engineering point clouds to fit tensor product B-spline surfaces by blending local fits. [arXiv:1411.5993](https://arxiv.org/abs/1411.5993) (2014)
30. Dellepiane, M., Venturi, A., Scopigno, R.: Image guided reconstruction of un-sampled data: a coherent filling for uncomplete Cultural Heritage models. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference*, pp. 939–946. IEEE (2009)
31. Quinsat, Y.: Filling holes in digitized point cloud using a morphing-based approach to preserve volume characteristics[J]. *Int. J. Adv. Manuf. Technol.* **81**(1–4), 411–421 (2015)



**Jun Xiao** is now an associate professor in University of the Chinese Academy of Sciences. He obtained his Ph.D. in the Graduate University of the Chinese Academy of Sciences in 2008. His research interests include computer graphics, computer vision, image processing and 3D reconstruction.



**Ying Wang** is now a professor in University of the Chinese Academy of Sciences. She received her Ph.D. in Beijing Institute of Technology in 1996. Her research interests include computer graphics, visualization and computer vision.



**Xiaoyuan Guo** received her B.Sc. in Information Security from the School of Computer and Engineering, Tianjin University of Technology in 2014. She is currently a master student at the School of Engineering Science, University of the Chinese Academy and Sciences, Beijing, China. Her research interests include hole filling of point clouds, 3D reconstruction and meshing and remeshing.