Ju T. Fixing geometric errors on polygonal models: A survey. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 24(1): 19–29 Jan. 2009

Fixing Geometric Errors on Polygonal Models: A Survey

Tao Ju (居　涛)

Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, Missouri, U.S.A.

E-mail: taoju@cs.wustl.edu

Received July 4, 2008; revised November 20, 2008.

Abstract Polygonal models are popular representations of 3D objects. The use of polygonal models in computational applications often requires a model to properly bound a 3D solid. That is, the polygonal model needs to be closed, manifold, and free of self-intersections. This paper surveys a sizeable literature for repairing models that do not satisfy this criteria, focusing on categorizing them by their methodology and capability. We hope to offer pointers to further readings for researchers and practitioners, and suggestions of promising directions for future research endeavors.

Keywords model repair, polygonal models, gaps, holes, intersections

1 Introduction

Polygonal representations of 3D objects, and in particular, triangular meshes, have become prevalent in numerous application domains. One of the key reasons of their popularity is that today's graphics hardware is highly specialized in displaying polygons, especially triangles, at interactive rates. Besides visualization, the flexibility and simplicity of polygons greatly facilitate designing, processing, transmitting, animating and interacting with 3D objects.

Polygonal models can be created in a number of ways. Some models are designed interactively using 3D modeling software. Some are reconstructed from raw data collected by imaging devices, such as scattered points (in 3D scanning) and grayscale volumes (in bio-medical imaging). In addition, other 3D representations, such as NURBS, subdivision surfaces, Constructive Solid Geometry, and implicit surfaces, often need to be converted into polygonal forms for visualization and computation.

To be useful in practice, a polygonal model needs to satisfy some *correctness* criteria demanded by the target application. While such criteria vary from one application to another, we differentiate between two criteria that are commonly required for performing computations on the model:

Geometric Correctness: The polygons should represent the exterior surface of a proper 3D solid. That is, the polygonal surface should be closed, manifold (i.e., having a disk-like surface neighborhood around each vertex) and free of self-intersections.

Topological Correctness: A polygonal model should have the same topology as the solid it represents. In particular, topological features, such as handles and connected components, should be preserved.

Geometric correctness is particularly important in engineering and manufacturing, where solid objects are needed for numerical computations, such as finite element analysis, and for actual production, such as rapid prototyping. The manifold requirement is also critical in geometry processing for computing differential quantities on surfaces, such as normals and curvatures. Topological correctness, on the other hand, further ensures that the polygonal model does not introduce extra complexity to the solid it represents, such as redundant handles and disconnected pieces, which would unnecessarily complicate geometry processing tasks such as simplification, parameterization, and segmentation. Note that the word "topology" here specifically refers to the topology of a 3D solid (e.g., handles and connected components), rather than how polygons are connected (e.g., being manifold).

Polygonal models created from various sources may not initially satisfy these correctness criteria. In this survey, we consider errors that violate the geometric correctness criterium, which we call *geometric errors*, and ways to fix them. In the rest of the paper, unless otherwise stated, we assume a polygonal model is made up of triangle faces.

Survey

This work is supported in part by NSF of USA under Grant Nos. CCF-0702662 and DBI-0743691.

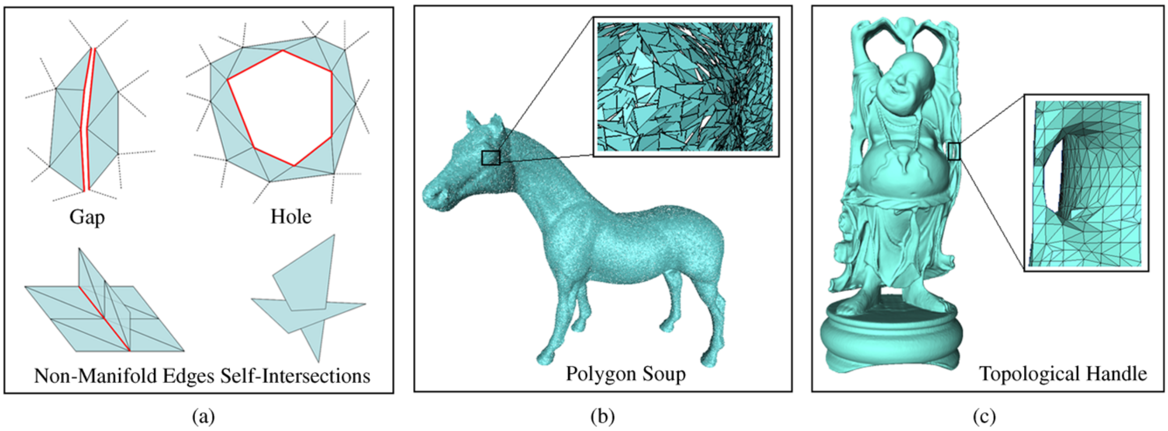

Fig.1. (a) Types of geometric errors. (b) A polygon soup that exhibits a mixture of geometric errors. (c) A topological handle is not considered as a geometric error.

1.1 Geometric Errors

One or more of the following errors may appear on a polygonal model that is not geometrically correct, which are illustrated in Fig.1(a).

• *Gaps* are cracks between neighboring polygons that should have been seamlessly connected. Gaps often appear on the junction between multiple polygonal patches that are created from different sources (e.g., multiple NURBS patches or 3D scans) or at different resolutions (as in level-of-detail modeling).

• *Holes* are missing geometry on the polygonal surface. They are usually artifacts resulted from surface reconstruction from insufficiently sampled data, such as incomplete 3D scans.

• *Non-Manifold Elements* include edges that are not contained in exactly two polygons and vertices that are not contained in a disk-like neighborhood. Non-manifold edges, in particular, often occur where there are redundant membranes interior to the model.

• *Self-Intersections* between polygons can occur as the result of a range of surface reconstruction, conversion, or editing procedures.

The above classification is not mutually exclusive. For example, the boundary of a gap or hole (highlighted in Fig.1(a)) typically consists of non-manifold edges contained in only one polygon.

It is quite common for a model to exhibit a mixture of these errors. This case is amplified in a type of polygon models known as "polygon soups" (or "bag of polygons"), which are collections of polygons without any connectivity information between them and possibly with numerous self-intersections (Fig.1(b)).

Again, the above geometric errors should be distinguished from topological ones. For example, the small ring-shaped handle in Fig.1(c) is a topological artifact introduced by surface reconstruction from scattered points. The model itself is still geometrically correct, as the surface is closed and manifold. Note that some researchers refer to topological handles as "3D holes"[1].

1.2 Overview

This paper surveys methods for correcting geometric errors on polygonal models. There is a vast literature on model repair, and this short survey does not intend either to cover all existing approaches or to discuss individual methods in depth. Instead, it categorizes a considerable number of representative methods by their similarity in methodology or common geometric errors that they address, offering pointers and directions for further reading. In addition, we suggest a number of promising directions for future research endeavors.

In this paper, various model repair methods are broadly classified into two genres. The first genre identifies and fixes errors directly on the polygons (hence called *mesh-based*), while the second genre indirectly repairs the model using an intermediate volumetric grid (hence called *volume-based*). Within each genre, the methods are organized by the specific types of geometric errors they are designed to fix.

The readers are also referred to the surveys by Veleba and Felkel[2], which reviews a smaller selection of methods but gives a more detailed classification of geometric errors, and by Breckon and Fisher[3], which reviews methods involving human perception for fixing a specific type of geometric errors (i.e., holes).

A closely related topic is surface reconstruction from raw data, and particularly, from scattered points obtained by 3D scanning. Many methods exist for generating geometrically correct surfaces directly from point data (see the survey by Mencl and Muller[4]). Although it is possible to repair a polygonal model by first converting it to points and then applying these methods, we are more interested here in techniques that utilize the polygonal geometry and connectivity.

2 Mesh-Based Approaches

Mesh-based approaches explicitly search for errors on the polygonal surface by examining the locations of vertices and how they are connected by polygons. Once found, the errors are fixed by performing *surgeries* on the input surface, such as adding or removing vertices, modifying their locations, and modifying the polygon connectivity.

2.1 Gaps and Holes

The majority of mesh-based methods has focused on filling the gaps and holes on a polygonal model, which are the most visible artifacts and are undesirable in most applications. Gaps and holes share similar boundary structures, which can be detected using similar techniques, yet differ in shape and size, which demands different methods to achieve a satisfactory fill.

2.1.1 Detecting Gaps and Holes

A simple and commonly used method for locating the boundary of gaps and holes is to trace edges contained in only one polygon[5,6]. Note that almost all hole-filling algorithms require that the traced boundary edges form closed boundary loops, each loop enclosing a hole. However, this requirement may not be met on noisy models, in which case manual or automated cleaning is necessary before proceeding with hole filling[7,8].

User-interactions have also been used for locating holes, with the goal of improving the reliability or quality of hole-filling. For example, when a complex hole is bounded by multiple boundaries (e.g., an outer boundary with several interior islands), the user can help to identify the desired set of boundaries[9]. Also, methods that construct smooth hole-filling patches often need users to identify the surface region around or near the hole whose appearance is to be matched by the hole-filling geometry[10,11].

2.1.2 Filling Gaps

Gaps are typically narrow bands of spaces between neighboring polygons, and can be filled by either stitching the two sides of a gap with a triangle strip or merging the corresponding vertices. Barequet and Sharir[12] showed that finding the triangle strip that minimizes a global function (such as total edge lengths) is an NP-hard problem, and used a heuristic search instead for an approximate solution. Similar distance measures have been used to guide the merging of vertices on the two sides of the gap[8,13–16]. The method of Borodin *et al.*[16], for example, formulates vertex merging as contraction of edges spanning across the gap, which is performed iteratively using an error-controlled priority queue as in progressive mesh simplification[17,18].

2.1.3 Filling Holes

Unlike gaps, holes have larger areas, and usually cannot be filled with a satisfactory look using a simple triangle strip or by merging boundary vertices. Once a closed boundary loop is identified, the bounded hole is typically filled by a 3D triangulation of the loop. Early approaches focus on finding a manifold and intersection-free triangulation that spans the loop. As shown by Barequet *et al.*[19], this problem is also NP-hard. Heuristic searches have been developed that incrementally triangulate the loop guided by minimal areas[12], minimal distances[6], or angle measures[5,20,21]. In particular, the minimal-area heuristic of Barequet and Sharir[12] has been adopted and improved by other researchers[22,23] for handling crenellation triangles and skinny triangles. In the special case where the hole boundary is almost co-planar, the hole triangulation can be reduced to a 2D problem by projecting the loop onto a best fitting plane[24].

More recent hole-filling methods focus on the appearance of the fill — naive triangulation of the hole may not appear smooth with its immediate surroundings. To achieve smoothness across the hole boundary, progress has been made in two main directions. In the first direction, an initial triangulation is constructed (e.g., using the approaches mentioned above) and then improved in a postprocessing step using various geometry fairing techniques (an example is shown in Fig.2(a)). For G^1 (or higher) continuity at the hole boundary, researchers have explored fairing techniques that minimize Laplacian variation[22], normal variation[25], curvature variation[7], thin-plate energy[26], and Willmore energy[27]. Mesh refinement, such as the edge-swapping and triangle-splitting algorithm of Pfeifle and Seidel[28], is also necessary if the hole triangulation has a different density from its surrounding. A similar approach was adopted by Levy[9], who triangulated the hole in a 2D parametric domain but faired the triangulation in 3D using a minimal energy surface functional. In the second direction, the smooth hole-filling patch is obtained by adapting scattered data fitting techniques, such as

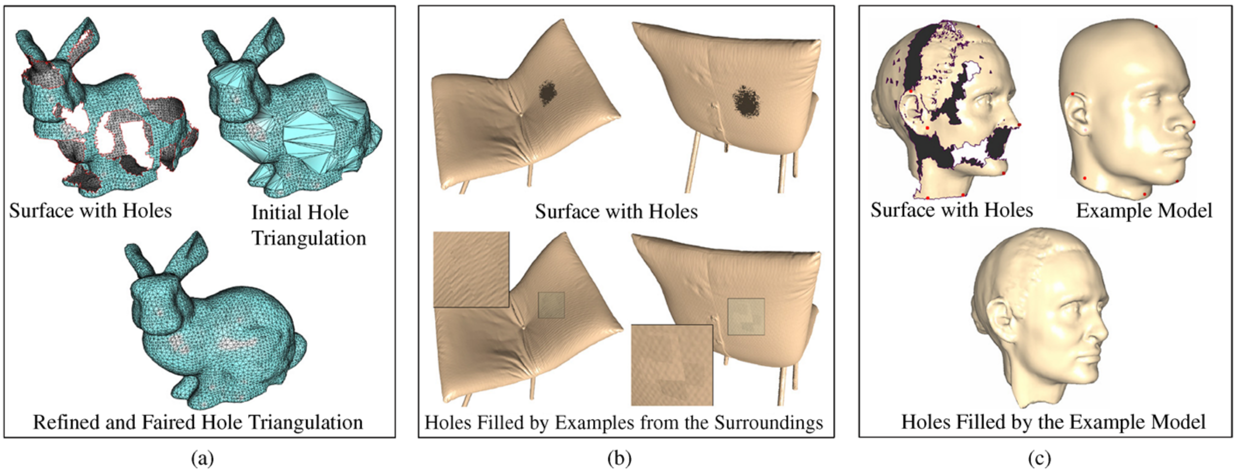

Fig.2. (a) Fill holes by triangulating followed by refinement and fairing[22]. (b) Recover missing geometry using examples on the same model[11]. (c) Recover missing geometry using another example model[35].

Moving Least Squares[29,30], Radial Basis Functions[31,32], and NURBS fitting[33], to fit the geometry surrounding the hole. A similar approach was taken by Jia and Tang[34], who completed the missing geometry by a tensor voting process guided by the normal directions of nearby polygons.

If a hole covers an area where there is interesting geometric details (for example, the texture on the couch in Fig.2(b)), a hole-filling patch that recovers those features would be more desirable than an otherwise smooth-looking one. Methods that are capable of recovering lost geometric features typically rely on the presence of these features in some existing examples, and hence are called *example-based* methods. These methods have been flourishing in recent literature, and can be roughly classified by whether the examples come from the input model itself or from some existing models. In the first class, the methods are inspired by the success of example-based techniques in the 2D image domain, such as texture synthesis and image completion, and involve patching the hole by finding and extending similar geometry in nearby regions on the input model, either in the 3D domain[11,36−39] or in the 2D parameterized domain[10]. A result using the method of Sharf *et al.*[11] is shown in Fig.2(b). Additional information, such as a photo of the missing geometry[40], or the presence of a global symmetry[41], can also help to infer the missing geometry.

On the other hand, if the example geometry is not available on the input model itself, the second class of example-based methods resort to shape matching and deformation to recover the missing geometry from existing models. A number of these methods have been developed in the context of face or character modeling from incomplete scans[42−46], utilizing a library of existing face or body models. Similar techniques were developed for repairing tooth scans[45,47] and CAD models[48]. The method of Kraevoy and Sheffer[35] further allows the input model and the example to exhibit rather different shapes, and utilizes cross-parameterization and blending to infer the missing geometry (Fig.2(c)). While these methods all require the examples and the model to be repaired to belong to a same class of objects (e.g., human heads), the method of Pauly *et al.*[49] automatically searches for examples in a model library using partial shape matching and blends deformed example parts from multiple models together. Note that this method requires point data to be present in the hole region for matching and deformation.

2.2 Non-Manifold Edges

Besides edges forming the boundary of a gap or hole, the input model may contain non-manifold polygonal edges shared by more than two polygons. Non-manifold connectivity often represents membranes interior to the model or redundant pieces of geometry. While some works separate these models into manifold patches possibly with boundaries[50,51], others eliminate the membranes by representing them as double-sided walls and removing overlapping polygons[52,53].

2.3 Geometric Intersections

Arguably, the most challenging type of errors for mesh-based methods is geometric intersections. Not only it is difficult to reliably detect intersections given the limited numerical precision of computers, these errors can also be generated as the result of other mesh-repair operations, such as hole-filling. The problem becomes more tractable if the locations of the intersections are known, for example, in the region where two intersection-free patches meet. A number of algorithms have been developed for resolving intersections when merging overlapping patches reconstructed from partial scans of a 3D object, and we refer readers to an excellent review in the paper by Rocchini *et al.*[54] As an example, the *zippering* method of Turk and Levoy[55] involves removing overlapped triangles between two patches, clipping one patch against another and removing the small triangles introduced during clipping.

3 Volume-Based Approaches

The mesh-based methods discussed above employ different techniques to identify and fix each type of geometric error, with the goal that the repaired polygonal surface forms the boundary of some 3D solid. Volume-based methods, on the other hand, determine the 3D solid in the first place (in a non-polygonal representation), and then extract the polygonal boundary of the solid as the repaired surface. Specifically, the input polygonal model is first converted into an intermediate volumetric grid, where each grid point is associated with a positive or negative sign indicating whether it is inside or outside the model. Next, a polygonal surface is reconstructed that separates the grid points of different signs. This process is illustrated in 2D in Fig.3(a).

The main benefit of going through an intermediate grid is that methods already exist for reconstructing a geometrically correct surface from a signed grid, a procedure known as *contouring* or *iso-surfacing* and studied extensively in the visualization community. Most notably, the Marching Cubes algorithm[56] (and later improvements[57−59]) extracts a closed, manifold and intersection-free triangulated surface from any signed uniform cubic grid by placing vertices on the grid edges and triangulating interior to each cubic cells. To efficiently process large models at fine resolutions, a class of *dual* methods, such as SurfaceNet[60] and Dual Contouring[61], were also proposed to contour a grid with adaptive resolution. These methods place vertices interior to the grid cells and are able to extract a crack-free surface on any octree grid. Recent improvements to these methods further ensure that the extracted surface is manifold[62−64] and free of self-intersections[65]. Extensions have also been proposed to other grid types, such as a kd-tree[66].

Assuming that a geometrically correct surface can be created from any signed grid, the focus of volume-based methods is on determining whether a grid point lies inside or outside the input model. Note that if the input model is free of geometric errors, the polygonal surface partitions the space into well-defined inside and outside volumes, from which the signs at grid points can be obtained (see a survey by Jones *et al.*[67] for sign generation methods on geometrically correct models). However, if a model contains gaps, holes, or self-intersections, defining its inside and outside is a non-trivial task. In the following we organize volume-based repair methods by the type of erroneous models they can accommodate, and the methodology they employ to determine the signs at the grid points.

3.1 Models Without Gaps or Holes

We first consider polygonal models that do not contain gaps or holes, but possibly exhibiting other types

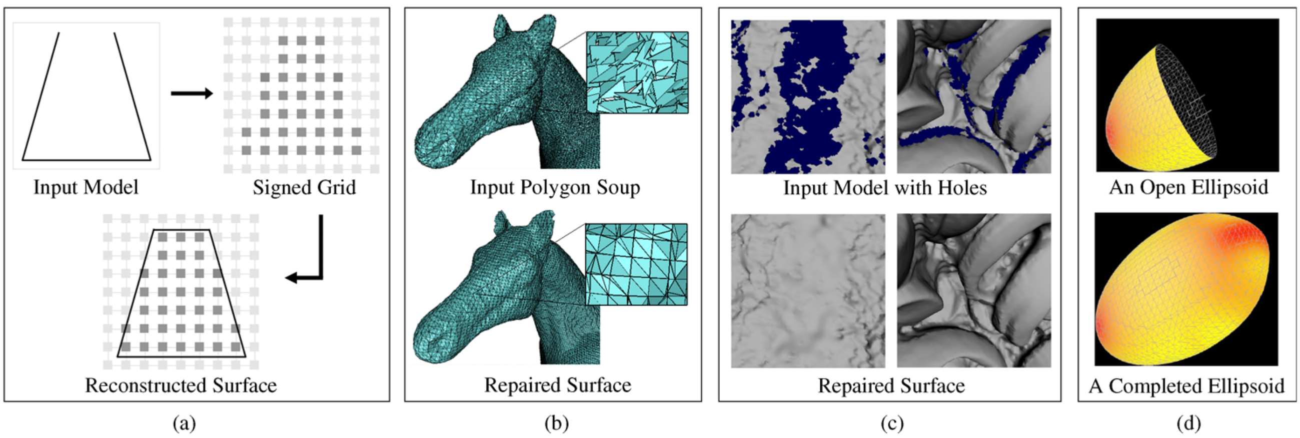

Fig.3. (a) Flow diagram of volume-based repair. (b) Repair polygon soups using an orientation-independent method[68]. (c) Fill holes by propagating signed distances using volumetric diffusion[69]. (d) Fill holes (an incomplete ellipsoid) by fitting local quadrics to the signed distances[70].

of errors. In this scenario, the geometry still partitions the entire space into disjoint regions. A number of methods therefore proceed by labeling these regions as either inside or outside via a connected-component search. The method of Oomes *et al.*[71] first discretizes the polygons onto a uniform grid as a set of object grid points and then identifies the inside region bounded by these points by flooding outward from a seed point interior to the model. Similarly, the method of Andujar *et al.*[72] discretizes the polygons onto an adaptive octree grid and then identifies the outside region by flooding inward from an outside bounding box. Note that outward (or inward) flooding is limited to identifying a single, connected inside (or outside) region, and thus cannot detect multiple object components (or cavities inside the object).

3.2 Models with Gaps and Holes

Open models with gaps and holes are more challenging for sign generation. In particular, the above-mentioned flooding approach would fail to separate the inside of the model from the outside. Methods that handle open models generally fall into two categories, the ones that do not rely on the orientation of the polygons and the ones that do. The former is more suitable for repairing polygon soups (see Fig.3(b)), while the later tends to produce better-looking hole-filling geometry when the polygons are consistently orientated (see Figs.3(c) and 3(d)).

3.2.1 Orientation-Independent Methods

These methods rely solely on the location and connectivity of the vertices in the input model. The methods can be broadly divided into global ones, which determine the sign at each grid point by tests involving the whole model or optimization on the whole grid, and local ones, which modify a small region on the grid (typically where the gaps and holes are) while ensuring that an inside and outside partitioning is well-defined on the rest of the grid.

In a global approach, Nooruddin and Turk[73] determined the sign at a grid point by ray stabbing and voting. The parity of the number of intersections between a ray shot from a grid point and the model contributes one vote to the inside or outside classification of that grid point, and multiple rays are shot from each grid point to make the final decision. One of the drawbacks of this method is that ray stabbing is a global operation, and a local surface error may cause a distant part of the space to be incorrectly signed. The limitation is partially overcome by Spillmann *et al.*[74], who considered the distances to the surface so that grid points further away from the surface are less likely to exhibit variation in signs. The minimal variation of sign change is formalized recently in a graph setting by Hornung and Kobbelt[75], who casted the inside and outside separation task as a min-cut problem in a volume graph where edges across the model surface are weighted less.

More localized schemes were adopted in the recent work of Ju[68] and Bischoff *et al.*[76], which explicitly identify and patch hole boundaries on the grid. Ju traced and patched hole boundaries on a dual surface of the primal grid, which yields an inside/outside partitioning on the primal grid. Bischoff *et al.*, on the other hand, performed morphological erosion and dilation from hole boundary cells to generate a plausible partitioning. Due to the localized operations, both methods can operate on space-efficient octree grids that are only refined along the model surfaces, and hence are capable of processing models at very high grid resolutions.

3.2.2 Orientation-Dependent Methods

If the input polygons are consistently oriented, their orientations provide extra hints that can guide the algorithms to determine signs more reliably or generate smoother-looking hole-filling geometry. Methods in this class make use of the signed distances from a grid point to the oriented polygons. While some methods use these signed distances to improve the inside/outside classification of each grid point, others propagate the signed distance field from more reliable regions (e.g., where the input geometry is correct) to less reliable regions (e.g., gaps and holes) to achieve smooth geometry completion. Note that the signed distances can also be generated from data-dependent information, such as line-of-sight directions of the scanner[77,78].

In a classification-based approach, the sign at each grid point is determined from signed distances to multiple polygons or in multiple scan directions by interpolation[77], consensus voting[79], and Bayesian classification[78]. The recent work of Sagawa *et al.*[80] further improves the classification at each grid point by sign flipping to minimize the area of the extracted iso-surface. Due to the use of signed distances, these approaches achieve arguably more robust results than orientation-independent ray-stabbing methods[73,74] that only consider the number of ray-model intersections.

Among propagation-based methods, Davis *et al.*[69] introduced a physically-based propagation algorithm that simulates heat diffusion, which yields hole-filling geometry that blends naturally with the nearby surface shape (see an example in Fig.3(c)). The diffusion method was further extended in [81] to recover sharp features in the hole area and formulated in [82] as a variational problem, similar to 2D image inpainting but now on the 3D signed distance volume, that minimizes curvature-based functionals on the iso-surface. In a different approach, the propagation can be accomplished by fitting smooth functions, such as local quadrics[70] (Fig.3(d)) or Moving Least Square polynomials[83] to the signed distance field near the hole. Along the same direction, a number of other fitting or interpolation techniques, such as Radial Basis Functions[84] and Poisson reconstruction[85], can also be considered (although have not yet appeared in literature).

4 Comparison and Discussion

In this section, we summarize and compare the two genres of model repair methods, and discuss a number of potential research directions.

4.1 Comparison

Mesh-based methods have been mostly successful in repairing models where a small portion of the surface is contaminated with errors (such as in CAD models) or in filling holes bounded by relatively simple, identifiable boundaries. The appearance of the hole-filling geometry can be quite appealing thanks to geometric fairing and example-based techniques. In addition, the repair does not affect regions on the model away from the error sites. The main drawback of mesh-based methods, however, is their lack of robustness. For example, simple edge-tracing in noisy data may not yield complete boundary loops required by most hole-filling methods. More importantly, geometric intersections (especially those caused by the repair operations, such as hole-filling) are difficult to prevent, detect, or repair.

In comparison, volume-based methods excel in their robustness in resolving various types of geometric errors (including self-intersections). The use of adaptive grids, such as octrees, further allows these methods to process models at high resolutions with small memory footprint and fast spatial queries, which are needed for repairing large models reconstructed from 3D scans. The major drawback of volume-based methods is the loss of geometric details of the original model (not only at the error sites) resulted from reconstructing the entire surface from an intermediate volume. Most volume-based methods use the Marching Cubes algorithm[56] for reconstruction, which generates a "blobby" surface that loses the sharp corners and edges on the input model. While feature-preserving contouring algorithms, such as Extended Marching Cubes[86] and Dual Contouring[61], have been adopted in recent repair methods[68,76], they still cannot exactly reproduce all geometric features, and additionally, the tessellations on the original model cannot be recovered. Although other geometry-preserving grid types have been proposed in the literature, such as kd-trees[66] and Extended Octrees[87], so far they have not been used for model repair.

4.2 Discussion

There have been a number of attempts in combining the robustness of volume-based approaches with geometry-preservation of mesh-based approaches. These attempts follow two general directions with potentials for future research. In the first direction, the methods of Bischoff *et al.*[88] and Podolak and Rusinkiewicz[89] convert only portions of the model containing geometric errors to an intermediate volume, and connect the reconstructed surfaces with the rest of the original model (Fig.4(a)). However, these methods typically assume specific types of input models, such as a non-intersecting orientable surface possibly containing holes[89] or a number of manifold patches possibly meeting with intersections and cracks[88]. In the second

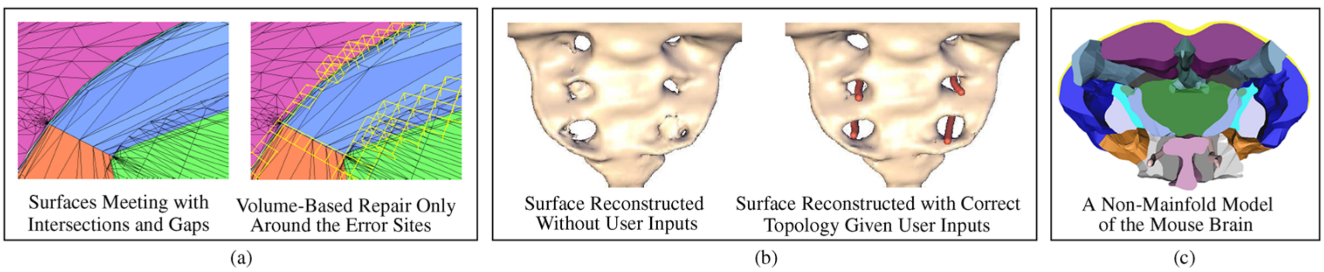

Fig.4. (a) Repair geometric errors using a partial volume grid around regions with errors[88]. (b) Reconstruct surface from scattered points with correct topology by incorporating user inputs (red bars)[91]. (c) Non-manifold polygonal model of the mouse brain partitioned into various anatomical regions (represented by colors) by internal membranes.

direction, the method of Murali and Funkhouser[90] utilizes a BSP-tree as the spatial grid, which has the advantage over rectilinear grids of being aligned with the polygonal geometry. However, the generation of BSP-trees for large polygonal inputs is time-consuming and prone to numerical errors. It would be interesting to investigate the use of other grid structures (such as kd-trees[66] and Extended Octrees[87]) for volume-based detail-preserving model repair.

A difficult problem for both mesh-based and volume-based approaches is how to fill complex holes on noisy data by plausible surfaces. As 3D scans become increasingly popular, completing missing geometry from the scans also becomes an important task. Holes in such scans are often characterized by complex boundaries and large areas of missing geometry. Filling these holes raises a number of challenges that have not been sufficiently addressed in the literature. For example, how to automatically identify the boundary of a hole if it consists of multiple boundary loops? How to determine the missing geometry if no examples are available, either on the model or in an existing library? How to combine the robustness of orientation-independent volume-based methods (such as [68,88]) with the smoothness of the surfaces offered by function fitting (such as Moving Least Squares and Radial Basis Functions) and example-based techniques (such as [11, 49])?

While we focus on repairing geometric errors in this survey, the repaired models may exhibit topological artifacts such as redundant handles (as in Fig.1(c)) and disconnected pieces. Note that these artifacts may either have come from the input model, or have been introduced during the repair process. While methods exist for fixing such errors in a separate post-processing step after geometric repair (see, for example, literature review in [92, 93]), the repaired model may deviate further from the original input. This motivates the need to develop algorithms that tackle both geometric and topological errors in a single pass to minimize information loss. Taking one step further, it would be most interesting to see algorithms that go straight from raw data, such as scattered points, to a geometrically and topologically correct model, eliminating the intermediate repair steps. The recent approach by Sharf *et al.*[91] makes an laudable first-step in this direction by incorporating user-inputs in generating a topologically satisfactory surface directly from scattered points (Fig.4(b)). Still, there is much room in future work along this direction for more reliable geometry reconstruction in the presence of large area of missing data, more robust control of surface topology, and more convenient user inputs.

Last but not least, the geometric criterium stated in Section 1 is biased towards the need to model 3D solids. In many applications, however, the polygonal models do not necessarily need to bound solids. Some models represent thin structures (e.g., a metal plate) or unorientable surfaces (e.g., the Mobius strip). Others may represent complex structures with internal partitions (e.g., a brain with internal anatomical divisions such as cortex and cerebellum, as in Fig.4(c)). These models contain non-manifold features, either at the boundaries or at internal membranes. How to fix other geometric errors (such as gaps, holes, intersections, and incorrect non-manifold elements) on such models while preserving the desirable set of non-manifold features? Unlike manifold models, non-manifold models do not partition the space into inside and outside, creating new challenges for volume-based repair.

References

[1] Aktouf Z, Bertrand G, Perroton L. A three-dimensional holes closing algorithm. *Pattern Recogn. Lett.*, 2002, 23(5): 523–531.

[2] Veleba D, Felkel P. Detection and correction of errors in surface representation. In *Proc. the 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Plzen-Bory, Czech Republic, 2007, pp.65–72.

[3] Breckon T P, Fisher R B. Amodal volume completion: 3D visual completion. *Comput. Vis. Image Underst.*, 2005, 99(3): 499–526.

[4] Mencl R, Muller H. Interpolation and approximation of surfaces from three-dimensional scattered data points. In *Proc. Scientific Visualization* (*Dagstuhl'97*), IEEE Computer Society, Washington DC, USA, 1999, pp.223–232.

[5] Bohn J H, Wozny M J. A topology-based approach for shell-closure. In *Selected and Expanded Papers from the IFIP TC5/WG5.2 Working Conference on Geometric Modeling for Product Realization*, Amsterdam, The Netherlands, North-Holland Publishing Co., 1992, pp.297–319.

[6] Makela I, Dolenc A. Some efficient procedures for correcting triangulated models. In *Proc. Solid Free Form Fabrication Symposium*, Austin, Texas, USA, August 9–11, 1993, pp.126–134.

[7] Pernot J P, Moraru G, Veron P. Filling holes in meshes using a mechanical model to simulate the curvature variation minimization. *Comput. Graph.*, 2006, 30(6): 892–902.

[8] Patel P S, Marcum D L, Remotigue M G. Stitching and filling: Creating conformal faceted geometry. In *Proc. the 14th International Meshing Roundtable*, San Diego, USA, Sept. 2005, pp.239–256.

[9] Levy B. Dual domain extrapolation. *ACM Trans. Graph.*, 2003, 22(3): 364–369.

[10] Nguyen M X, Yuan X, Chen B. Geometry completion and detail generation by texture synthesis. *The Visual Computer*, 2005, 21(8-10): 669–678.

[11] Sharf A, Alexa M, Cohen-Or D. Context-based surface completion. In *ACM SIGGRAPH 2004 Papers* (*SIGGRAPH'04*), Los Angeles, California, USA, August 8–12, 2004, pp.878–887.

[12] Barequet G, Sharir M. Filling gaps in the boundary of a polyhedron. *Comput. Aided Geom. Des.*, 1995, 12(2): 207–229.

[13] Rock S J, Wozny M J. Generating topological information from a "bucket of facets". In *Proc. Solid Freeform Fabrication Symposium*, Austin, USA, August 3–5, 1992, pp.251–259.

[14] Sheng X, Meier I R. Generating topological structures for surface models. *IEEE Comput. Graph. Appl.*, 1995, 15(6): 35–41.

[15] Barequet G, Kumar S. Repairing CAD models. In *Proc. the 8th Conference on Visualization* (*VIS'97*), Los Alamitos: IEEE Computer Society Press, CA, USA, 1997, pp.363–ff.

[16] Borodin P, Novotni M, Klein R. Progressive gap closing for mesh repairing. In *Proc. CGI'02*, Bradford, UK, 2002, pp.201–213.

[17] Hoppe H. Progressive meshes. In *Proc. the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH'96*), New Orleans, USA, August 1996, pp.99–108.

[18] Garland M, Heckbert P S. Surface simplification using quadric error metrics. In *Proc. the 24th Annual Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH'97*), Los Angeles, USA, August 3–8, 1997, pp.209–216.

[19] Barequet G, Dickerson M, Eppstein D. On triangulating three-dimensional polygons. In *Proc. the Twelfth Annual Symposium on Computational Geometry* (*SCG'96*), Philadelphia, USA, August 1996, pp.38–47.

[20] Wagner M, Labsik U, Greiner G. Repairing non-manifold triangle meshes using simulated annealing. *International Journal of Shape Modeling*, 2003, 9(2): 137–153.

[21] Varnuska M, Parus J, Kolingerova I. Simple holes triangulation in surface reconstruction. In *Proc. Algoritmy*, Vysoke Tatry, Podbanske, March 2005, pp.280–289.

[22] Liepa P. Filling holes in meshes. In *Proc. the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP'03*), Aachen, Germany, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2003, pp.200–205.

[23] Attene M, Falcidieno B. ReMESH: An interactive environment to edit and repair triangle meshes. In *Proc. the IEEE International Conference on Shape Modeling and Applications 2006* (*SMI'06*), Matsushima, Japan, March 2006, p.41.

[24] Roth G, Wibowoo E. An efficient volumetric method for building closed triangular meshes from 3-D image and point data. In *Proc. the Conference on Graphics Interface'97*, Kelowna, Canada, June 23–27, 1997, pp.173–180.

[25] Zhao W, Gao S, Lin H. A robust hole-filling algorithm for triangular mesh. *Vis. Comput.*, 2007, 23(12): pp.987–997.

[26] Bac A, Tran N V, Daniel M. A multistep approach to restoration of locally undersampled meshes. In *Proc. GMP*, Hangzhou, China, April 23–25, 2008, pp.272–289.

[27] Clarenz U, Diewald U, Dziuk G, Rumpf M, Rusu R. A finite element method for surface restoration with smooth boundary conditions. *Comput. Aided Geom. Des.*, 2004, 21(5): 427–445.

[28] Pfeifle R, Seidel H P. Triangular B-splines for blending and filling of polygonal holes. In *Proc. the Conference on Graphics Interface* (*GI'96*), Toronto, Ont., Canada, Canadian Information Processing Society, 1996, pp.186–193.

[29] Wang J, Oliveira M M. Filling holes on locally smooth surfaces reconstructed from point clouds. *Image Vision Comput.*, 2007, 25(1): 103–113.

[30] Tekumalla L S, Cohen E. A hole-filling algorithm for triangular meshes. Technical Report UUCS-04-019, University of Utah, 2004.

[31] Branch J, Prieto F, Boulanger P. Automatic hole-filling of triangular meshes using local radial basis function. In *Proc. the Third International Symposium on 3D Data Processing, Visualization, and Transmission* (*3DPVT'06*), Chapel Hill, USA, June 14–16, 2006, pp.727–734.

[32] Chen C Y, Cheng K Y. A sharpness-dependent filter for recovering sharp features in repaired 3D mesh models. *IEEE Transactions on Visualization and Computer Graphics*, 2008, 14(1): 200–212.

[33] Kumar A, Shih A M, Ito Y, Ross D H, Soni B K. A hole-filling algorithm using non-uniform rational B-splines. In *Proc. the 16th International Meshing Roundtable*, Seattle, USA, Oct. 14–17, 2007, pp.169–182.

[34] Jia J, Tang C K. Inference of segmented color and texture description by tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2004, 26(6): 771–786.

[35] Kraevoy V, Sheffer A. Template-based mesh completion. In *Proc. the Third Eurographics Symposium on Geometry Processing* (*SGP'05*), Vienna, Austria, Eurographics Association, Aire-la-Ville, Switzerland, 2005, p.13.

[36] Bendels G H, Schnabel R, Klein R. Detail-preserving surface inpainting. In *Proc. the 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, Pisa, Italy, June 2005, pp.41–48.

[37] Breckon T P, Fisher R B. Non-Parametric 3D surface completion. In *Proc. the Fifth International Conference on 3-D Digital Imaging and Modeling* (*3DIM '05*), Ottawa, Canada, June 13–16, 2005, pp.573–580.

[38] Park S, Guo X, Shin H, Qin H. Surface completion for shape and appearance. *Vis. Comput.*, 2006, 22(3): 168–180.

[39] Xiao C, Zheng W, Miao Y, Zhao Y, Peng Q. A unified method for appearance and geometry completion of point set surfaces. *Vis. Comput.*, 2007, 23(6): 433–443.

[40] Xu S, Georghiades A, Rushmeier H, Dorsey J, McMillan L. Image guided geometry inference. In *Proc. the Third International Symposium on 3D Data Processing, Visualization, and Transmission* (*3DPVT'06*), Chapel Hill, USA, June 14–16, 2006, pp.310–317.

[41] Pauly M, Mitra N J, Wallner J, Pottmann H, Guibas L. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics*, 2008, 27(3): 1–11.

[42] Blanz V, Vetter T. A morphable model for the synthesis of 3D faces. In *Proc. the 26th Annual Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH'99*), Los Angeles, August 8–13, 1999, pp.187–194.

[43] Kahler K, Haber J, Yamauchi H, Seidel H P. Head shop: Generating animated head models with anatomical structure. In *Proc. the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (*SCA'02*), San Antonio, Texas, USA, July 21–26, 2002, pp.55–63.

[44] Allen B, Curless B, Popović Z. The space of human body shapes: Reconstruction and parameterization from range scans. In *ACM SIGGRAPH 2003 Papers* (*SIGGRAPH'03*), San Diego, California, USA, July 27–31, 2003, pp.587–594.

[45] Blanz V, Mehl A, Vetter T, Seidel H P. A statistical method for robust 3D surface reconstruction from sparse data. In *Proc. 2nd International Symp. 3D Data Processing, Visualization, and Transmission,* (*3DPVT'04*), Thessaloniki, Greece, Sept. 6–9, 2004, pp.293–300.

[46] Anguelov D, Srinivasan P, Koller D, Thrun S, Rodgers J, Davis J. SCAPE: Shape completion and animation of people. *ACM Trans. Graph.*, 2005, 24(3): pp.408–416.

[47] Savchenko V, Kojekine N. An approach to blend surfaces. In *Proc. CGI'02*, Bradford, UK, July 1–5, 2002, pp.139–150.

[48] Huang Y, Wang Q, Huang Z, Wu J. Repairing incomplete measured data with a deformable model under constraints of feature shapes. *The International Journal of Advanced Manufacturing*, 2006, 28(1/2): 92–100.

[49] Pauly M, Mitra N J, Giesen J, Gross M, Guibas L J. Example-based 3D scan completion. In *Proc. the Third Eurographics Symposium on Geometry Processing* (*SGP'05*), Vienna, Austria, July 4–6, 2005, p.23.

[50] Barequet G, Duncan C A, Kumar S. RSVP: A geometric toolkit for controlled repair of solid models. *IEEE Transactions on Visualization and Computer Graphics*, 1998, 4(2): 162–177.

[51] Gueziec A, Taubin G, Lazarus F, Horn B. Cutting and stitching: Converting sets of polygons to manifold surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 2001, 7(2): 136–151.

[52] Baum D R, Mann S, Smith K P, Winget J M. Making radiosity usable: Automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. *SIGGRAPH Comput. Graph.*, 1991, 25(4): 51–60.

[53] Jan Helge Bohn. Removing zero-volume parts from CAD models for layered manufacturing. *IEEE Comput. Graph. Appl.*, 1995, 15(6): 27–34.

[54] Rocchini C, Cignoni P, Ganovelli F, Montani C, Pingi P, Scopigno R. The marching intersections algorithm for merging range images. *Vis. Comput.*, 2004, 20(2): 149–164.

[55] Turk G, Levoy M. Zippered polygon meshes from range images. In *Proc. the 21st Annual Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH'94*), New York, NY, USA, ACM, 1994, pp.311–318.

[56] Lorensen W E, Cline H E. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph.*, 1987, 21(4): 163–169.

[57] Nielson G M, Hamann B. The asymptotic decider: Resolving the ambiguity in marching cubes. In *Proc. the 2nd Conference on Visualization* (*VIS'91*), Los Alamitos, CA, USA, IEEE Computer Society Press, 1991, pp.83–91.

[58] Montani C, Scateni R, Scopigno R. A modified look-up table for implicit disambiguation of marching

[59] Allen van Gelder, Jane Wilhelms. Topological considerations in isosurface generation. *ACM Trans. Graph.*, 1994, 13(4): 337–375. cubes. *The Visual Computer*, 1994, 10(6): 353–355.

[60] Gibson S F F. Using distance maps for accurate surface representation in sampled volumes. In *Proc. the 1998 IEEE Symposium on Volume Visualization* (*VVS'98*), New York, USA, ACM, 1998, pp.23–30.

[61] Ju T, Losasso F, Schaefer S, Warren J. Dual contouring of hermite data. In *Proc. the 29th Annual Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH'02*), San Antonio, Texas, USA, July 21–26, 2002, pp.339–346.

[62] Ashida K, Badler N I. Feature preserving manifold mesh from an octree. In *Proc. Symposium on Solid Modeling and Applications*, Seattle, USA, June 16–20, 2003, pp.292–297.

[63] Zhang N, Hong W, Kaufman A E. Dual contouring with topology-preserving simplification using enhanced cell representation. In *Proc. IEEE Visualization*, Austin, USA, Oct. 10–15, 2004, pp.505–512.

[64] Schaefer S, Ju T, Warren J. Manifold dual contouring. *IEEE Transactions on Visualization and Computer Graphics*, 2007, 13(3): 610–619.

[65] Ju T, Udeshi T. Intersection-free contouring on an octree grid. In *Proc. Pacific Graphics'06*, Oct. 11–13, 2006, Taipei, Taiwan, China.

[66] Gress A, Klein R. Efficient representation and extraction of 2-manifold isosurfaces using kd-trees. *Graph. Models*, 2004, 66(6): 370–397.

[67] Jones M W, Baerentzen J A, Sramek M. 3D distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics*, 2006, 12(4): 581–599.

[68] Ju T. Robust repair of polygonal models. *ACM Trans. Graph.*, 2004, 23(3): 888–895.

[69] Davis J, Marschner S R, Garr M, Levoy M. Filling holes in complex surfaces using volumetric diffusion. In *Proc. 3DPVT*, Padova, Italy, June 19–21, 2002, pp.428–438.

[70] Masuda T. Filling the signed distance field by fitting local quadrics. In *Proc. 2nd International Symp. the 3D Data Processing, Visualization, and Transmission* (*3DPVT'04*), Thessaloniki, Greece, June 19–21, 2004, pp.1003–1010.

[71] Oomes S, Snoeren P, Dijkstra T. 3D shape representation: Transforming polygons into voxels. In *Proc. the First International Conference on Scale-Space Theory in Computer Vision* (*SCALE-SPACE'97*), London, UK, Springer-Verlag, July 1997, pp.349–352.

[72] Andujar C, Brunet P, Ayala D. Topology-reducing surface simplification using a discrete solid representation. *ACM Trans. Graph.*, 2002, 21(2): 88–105.

[73] Fakir S Nooruddin, Greg Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph.*, 2003, 9(2): 191–205.

[74] Spillmann J, Wagner M, Teschner M. Robust tetrahedral meshing of triangle soups. In *Proc. Vision, Modeling, Visualization* (*VMV'06*), Aachen, Germany, Nov. 22–24, 2006, pp.9–15.

[75] Hornung A, Kobbelt L. Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In *Proc. the Fourth Eurographics Symposium on Geometry Processing* (*SGP'06*), Sardinia, Italy, June 2006, pp.41–50.

[76] Bischoff S, Pavic D, Kobbelt L. Automatic restoration of polygon models. *ACM Trans. Graph.*, 2005, 24(4): 1332–1352.

[77] Curless B, Levoy M. A volumetric method for building complex models from range images. In *Proc. the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH'96*), New Orleans, USA, August 4–9, 1996, pp.303–312.

[78] Furukawa R, Itano T, Morisaka A, Kawasaki H. Improved space carving method for merging and interpolating multiple range images using information of light sources of active stereo. In *Proc. ACCV (2)*, Tokyo, Japan, Nov. 2007, pp.206–216.

[79] Sagawa R, Ikeuchi K. Taking consensus of signed distance field for complementing unobservable surface. In *Proc. 3DIM*, Banff, Canada, Oct. 2003, pp.410–417.

[80] Sagawa M R, Ikeuchi F K. Hole filling of a 3D model by flipping signs of a signed distance field in adaptive resolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008, 30(4): 686–699.

[81] Guo T Q, Li J J, Weng J G, Zhuang Y T. Filling holes in complex surfaces using oriented voxel diffusion. In *Proc. 2006 International Conference on Machine Learning and Cybernetics*, Dalian, China, August 2006, pp.4370–4375.

[82] Verdera J, Caselles V, Bertalmio M, Sapiro G. Inpainting surface holes. In *Proc. Int. Conference on Image Processing*, Barcelona, Spain, Sept. 2003, pp.903–906.

[83] Shen C, O'Brien J F, Shewchuk J R. Interpolating and approximating implicit surfaces from polygon soup. In *ACM SIGGRAPH 2004 Papers* (*SIGGRAPH'04*), Los Angeles, California, USA, August 8–12, 2004, pp.896–904.

[84] Carr J C, Beatson R K, Cherrie J B, Mitchell T J, Fright W R, McCallum B C, Evans T R. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. the 28th Annual Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH'01*), Los Angeles, USA, August 12–17, 2001, pp.67–76.

[85] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction. In *Proc. the Fourth Eurographics Symposium on Geometry Processing* (*SGP'06*), Sardinia, Italy, June 26–28, 2006, pp.61–70.

[86] Kobbelt L P, Botsch M, Schwanecke U, Seidel H P. Feature sensitive surface extraction from volume data. In *Proc. the 28th Annual Conference on Computer Graphics and Interactive Techniques* (*SIGGRAPH'01*), Los Angeles, USA, August 12–17, 2001, pp.57–66.

[87] Brunet P, Navazo I. Solid representation and operation using extended octrees. *ACM Trans. Graph.*, 1990, 9(2): 170–197.

[88] Bischoff S, Kobbelt L. Structure preserving CAD model repair. *Comput. Graph. Forum*, 2005, 24(3): 527–536.

[89] Podolak J, Rusinkiewicz S. Atomic volumes for mesh completion. In *Proc. the third Eurographics Symposium on Geometry Processing* (*SGP'05*), Vienna, Austria, July 4–6, 2005, p.33.

[90] Murali T M, Funkhouser T A. Consistent solid and boundary representations from arbitrary polygonal data. In *Proc. the 1997 Symposium on Interactive 3D Graphics* (*SI3D'97*), New York, NY, USA, April 27–30, 1997, pp.155–ff.

[91] Sharf A, Lewiner T, Shklarski G, Toledo S, Cohen-Or D. Interactive topology-aware surface reconstruction. *ACM Trans. Graph.*, 2007, 26(3): 43.

[92] Wood Z J, Hoppe H, Desbrun M, Schroder P. Removing excess topology from isosurfaces. *ACM Trans. Graph.*, 2004, 23(2): 190–208.

[93] Ju T, Zhou Q Y, Hu S M. Editing the topology of 3D models by sketching. *ACM Trans. Graph.*, 2007, 26(3): 42.

Tao Ju is an assistant professor in the Department of Computer Science and Engineering at the Washington University in St. Louis, USA. He obtained his M.S. and Ph.D. degrees in computer science from Rice University in 2003 and 2005, and received post-doctoral training at Baylor College of Medicine in the summer of 2005. He conducts research in computer graphics and bio-medical applications, and is particularly interested in geometric modeling and processing. His work is supported by NSF and NIH.