

Fixing Geometric Errors on Polygonal Models: A Survey

Tao Ju¹

¹Computer Science and Engineering, Washington University in St. Louis, St. Louis, Missouri, USA

E-mail: taoju@cs.wustl.edu

Received MONTH DATE, YEAR.

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 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 geometrically correct polygonal model should additionally 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.

* This work is supported by NSF under Grant CCF-0702662.

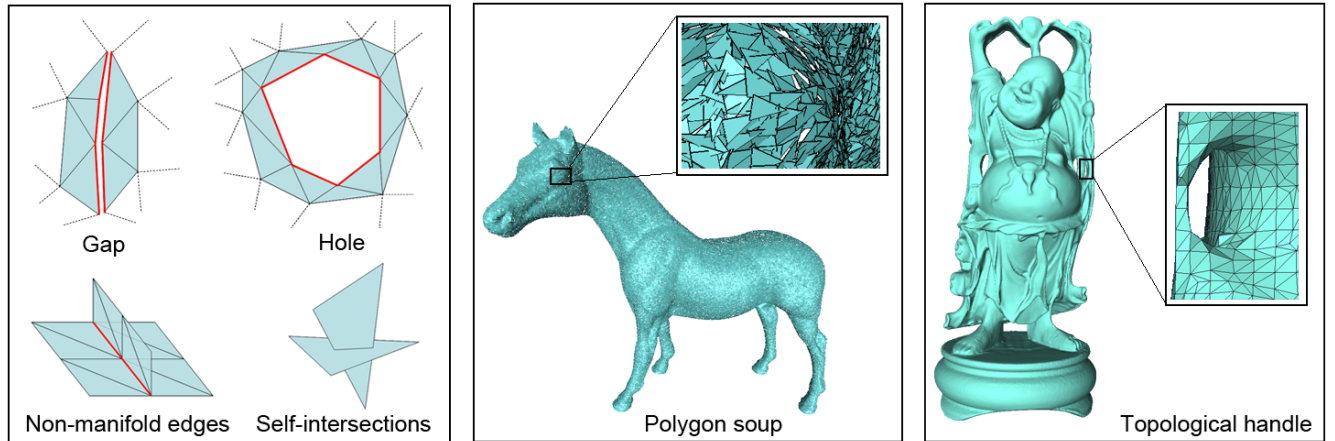


Figure 1: Left: types of geometric errors. Middle: a polygon soup that exhibits a mixture of geometric errors. Right: a topological handle is not considered as a geometric error.

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 Figure 1 left.

- **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 Figure 1 left) 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 (Figure 1 middle).

Again, the above geometric errors should be distinguished from *topological* ones. For example, the small ring-shaped handle in Figure 1 right 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 de-

signed to fix.

The readers are also referred to the surveys by Veleba and Felkel [84], which reviews a smaller selection of methods but gives a more detailed classification of geometric errors, and by Breckon and Fisher [23], 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 scanners. Many methods exist for generating geometrically correct surfaces directly from point data (see the survey by Mencl and Muller *et. al.* [55]). 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 fill satisfactorily.

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 [20, 53]. 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 [63, 66].

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 right set of boundaries [50]. Also, methods that construct smooth hole-filling patches often need users to identify the surface region whose appearance is to be matched by the hole-filling geometry [58, 75].

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 [70, 78, 11, 21, 63]. The method of Borodin *et. al.* [21], for example, formulates vertex merging as contraction of edges spanning across the gap, which is performed iteratively using a error-controlled priority queue as in progressive mesh simplification [37, 32].

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.* [9], this problem is also NP-hard. Heuristic searches have been developed that incrementally triangulate the loop guided by minimal areas [12], minimal distances [53], or angle measures [20, 86, 83]. In particular, the minimal-area heuristic of Barequet and Sharir [12] has been adopted and improved by other researchers [51, 7] 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 [71].

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

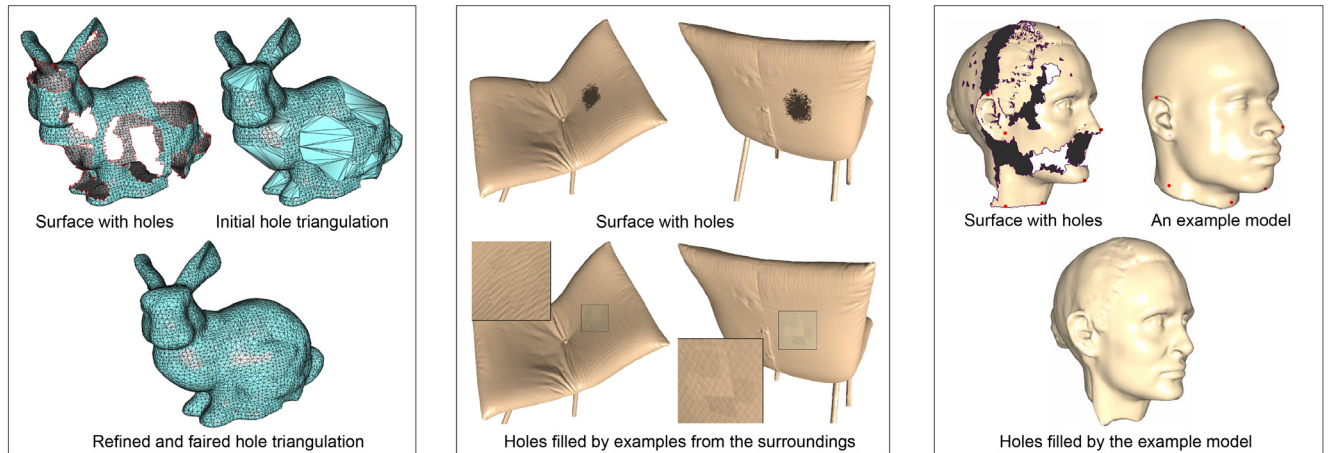


Figure 2: Left: fill holes by triangulating followed by refinement and fairing [51]. Middle: recover missing geometry using examples on the same model [75]. Right: recover missing geometry using another example model [48].

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 post-processing step using various geometry fairing techniques (an example is shown in Figure 2 left). For G^1 (or higher) continuity at the hole boundary, researchers have explored fairing techniques that minimize Laplacian variation [51], normal variation [93], curvature variation [66], thin-plate energy [8], and Willmore energy [28]. Mesh refinement, such as the edge-swapping and triangle-splitting algorithm of Pfeifle and Seidel [67], is also necessary if the hole triangulation has a different density than its surrounding. A similar approach was adopted by Levy [50], 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 Moving Least Squares [87, 80], Radial Basis Functions [22, 27], and NURBS fitting [49], to fit the geometry surrounding the hole. A similar approach was taken by Jia and Tang [39], 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 Figure 2 middle), 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 [75, 14, 24, 62, 89] or in the 2D parameterized domain [58]. A result using the method of Sharf *et al.* [75] is shown in Figure 2 middle. Additional information, such as a photo of the missing geometry [90], or the presence of a global symmetry [64], 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 [18, 45, 2, 17, 5], utilizing a library of existing face or body models. Similar techniques were developed for repairing tooth scans [3, 17] and CAD models [91]. The method of Kraevoy and Sheffer [48] further allows the input model and the example to exhibit rather different shapes, and utilizes cross-parameterization and blending to infer the missing geometry (Figure 2 right). While these methods all require the knowledge of the kind of examples to be used, the method of Pauly *et al.* [65] auto-

matically 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 [10, 35], others eliminate the membranes by representing them as double-sided walls and removing overlapping polygons [13, 19].

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, particularly 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. In particular, 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.* [69]. As an example, the *zippering* method of Turk and Levoy [81] 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, determines the 3D solid in the first place (in a non-polygonal representation), and then extracts 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 Figure 3 left.

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 [52] (and later improvements [59, 56, 82]) 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 [33] and Dual Contouring [42], were 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 [6, 92, 74] and free of self-intersections [43]. Extensions have also been proposed to other grid types, such as a kd-tree [34].

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.* [40] 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 of errors. In this scenario, the geometry still partitions the entire space into disjoint re-

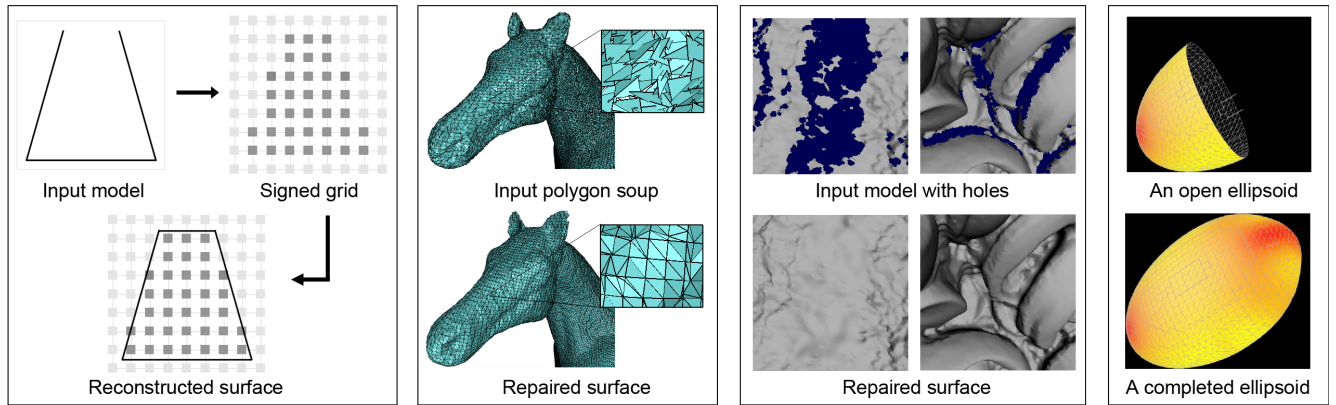


Figure 3: Left: The flow diagram of volume-based repair. Middle-left: repair polygon soups using an orientation-independent method [41]. Middle-right: fill holes by propagating signed distances using volumetric diffusion [30]. Right: fill holes (an incomplete ellipsoid) by fitting local quadrics to the signed distances [54].

gions. A number of methods therefore proceed by labelling these regions as either inside or outside via a connected-component search. The method of Oomes *et. al.* [61] 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.* [4] 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 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, ones that do not rely on the orientation of the polygons and ones that do. The former is more suitable for repairing polygon soups (see Figure 3 middle-left), while the latter tends to produce better-looking hole-filling geometry when the polygons are consistently orientated (see Figure 3 middle-right and right).

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) so that an inside and outside partitioning is well-defined on the rest of the grid.

On the global side, the method of Nooruddin and Turk [60] determines the sign at a grid point by ray stabbing and voting. The parity of the number of intersections between a ray and the model contributes one vote to the inside or outside classification of a 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.* [79], who also 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 [38], 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 [41] and Bischoff *et. al.* [16], 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 mor-

phological 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 in 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 [29, 31].

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 [29], consensus voting [73], and Bayesian classification [31]. The recent work of Sagawa *et al.* [72] 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 [60, 79] that only consider the number of ray-model intersections.

Among propagation-based methods, Davis *et al.* [30] introduced a physically-based propagation algorithm that simulates heat diffusion, which yields hole-filling geometry that naturally extends the nearby geometry (see an example in Figure 3 middle-right). The diffusion method was further extended in [36] to recover sharp features in the hole area and formulated in [85] 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 [54] (Figure 3 right) or Moving Least Square polynomials [77], 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 [26] and Poisson reconstruction [46], 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 most successful in repairing models where a small portion of the surface is contaminated with errors (such as in CAD models) or in filling holes with 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 [52] 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 [47] and Dual Contouring [42], have been proposed and used in recent repair methods [41, 16], they still cannot exactly reproduce all geometric features, and additionally, the tessellations on the original model cannot be recovered. Al-

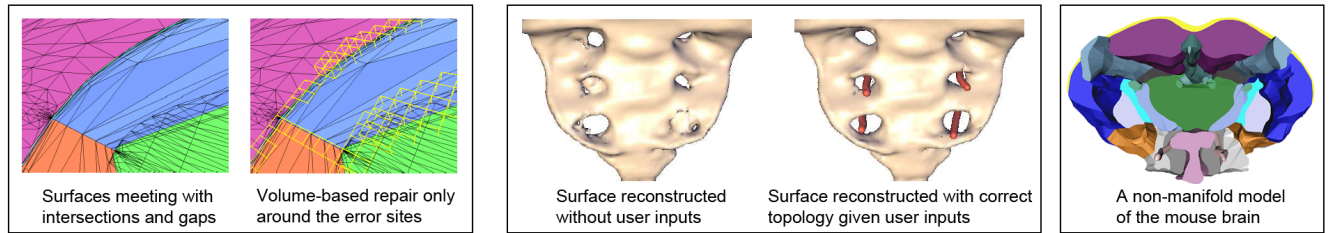


Figure 4: Left: repair geometric errors using a partial volume grid around regions with errors [15]. Middle-right: reconstruct surface from scattered points with correct topology by incorporating user inputs (red bars) [76]. Right: a non-manifold polygonal model of the mouse brain partitioned into various anatomical regions (represented by colors) by internal membranes.

though other geometry-preserving grid types have been proposed in the literature, such as kd-trees [34] and Extended Octrees [25], 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.* [15] and Podolak and Rusinkiewicz [68] 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 (Figure 4 left). However, these methods typically assume specific types of input models, such as a non-intersecting orientable surface possibly containing holes [68] or a number of manifold patches possibly meeting with intersections and cracks [15]. In the second direction, the method of [57] 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 or noisy 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 [34] or Extended Octrees [25]) 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 ar-

reas 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 [41, 15]) 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 [75, 65])?

While we focus on repairing geometric errors in this survey, the repaired models may exhibit topological artifacts such as redundant handles (as in Figure 1 right) 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 [88, 44]), 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.* [76] makes an laudable first-step in this direction by incorporating user-inputs in generating a topologically satisfactory surface directly from scattered points (Figure 4 middle). 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 un-orientable 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 Figure 4 right). 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] Zouina Aktouf, Gilles Bertrand, and Laurent Perroton. A three-dimensional holes closing algorithm. *Pattern Recogn. Lett.*, 23(5):523–531, 2002.
- [2] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 587–594, New York, NY, USA, 2003. ACM.
- [3] V. Savchenko and N. Kojekine. An approach to blend surfaces. In *CGI'02*, pages 139–150, 2002.
- [4] Carlos Andujar, Pere Brunet, and Dolors Ayala. Topology-reducing surface simplification using a discrete solid representation. *ACM Trans. Graph.*, 21(2):88–105, 2002.
- [5] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, 2005.
- [6] Koji Ashida and Norman I. Badler. Feature preserving manifold mesh from an octree. In *Symposium on Solid Modeling and Applications*, pages 292–297, 2003.
- [7] Marco Attene and Bianca Falcidieno. Remesh: An interactive environment to edit and repair triangle meshes. In *SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*, page 41, Washington, DC, USA, 2006. IEEE Computer Society.
- [8] Alexandra Bac, Nam-Van Tran, and Marc Daniel. A multistep approach to restoration of locally undersampled meshes. In *GMP*, pages 272–289, 2008.
- [9] Gill Barequet, Matthew Dickerson, and David Eppstein. On triangulating three-dimensional polygons. In *SCG '96: Proceedings of the twelfth annual symposium on Computational geometry*, pages 38–47, New York, NY, USA, 1996. ACM.
- [10] Gill Barequet, Christian A. Duncan, and Subodh Kumar. Rsvp: A geometric toolkit for controlled repair of solid models. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):162–177, 1998.
- [11] Gill Barequet and Subodh Kumar. Repairing cad models. In *VIS '97: Proceedings of the 8th conference on Visualization '97*, pages 363–ff., Los Alamitos, CA, USA, 1997. IEEE Computer Society Press.
- [12] Gill Barequet and Micha Sharir. Filling gaps in the boundary of a polyhedron. *Comput. Aided Geom. Des.*, 12(2):207–229, 1995.
- [13] Daniel R. Baum, Stephen Mann, Kevin P. Smith, and James M. Winget. Making radiosity usable: automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. *SIGGRAPH Comput. Graph.*, 25(4):51–60, 1991.
- [14] G.H. Bendels, R. Schnabel, and R. Klein. Detail-preserving surface inpainting. In *Proceedings of the 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 41–48, 2005.
- [15] Stephan Bischoff and Leif Kobbelt. Structure preserving cad model repair. *Comput. Graph. Forum*, 24(3):527–536, 2005.

- [16] Stephan Bischoff, Darko Pavic, and Leif Kobbelt. Automatic restoration of polygon models. *ACM Trans. Graph.*, 24(4):1332–1352, 2005.
- [17] Volker Blanz, Albert Mehl, Thomas Vetter, and Hans-Peter Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 293–300, Washington, DC, USA, 2004. IEEE Computer Society.
- [18] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [19] Jan Helge Bohn. Removing zero-volume parts from cad models for layered manufacturing. *IEEE Comput. Graph. Appl.*, 15(6):27–34, 1995.
- [20] Jan Helge Bohn and Michael J. Wozny. 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*, pages 297–319, Amsterdam, The Netherlands, The Netherlands, 1992. North-Holland Publishing Co.
- [21] P. Borodin, M. Novotni, and R. Klein. Progressive gap closing for mesh repairing. In *Advances in Modelling, Animation and Rendering*, pages 201–213, 2002.
- [22] John Branch, Flavio Prieto, and Pierre Boulanger. Automatic hole-filling of triangular meshes using local radial basis function. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 727–734, Washington, DC, USA, 2006. IEEE Computer Society.
- [23] Toby P. Breckon and Robert B. Fisher. Amodal volume completion: 3d visual completion. *Comput. Vis. Image Underst.*, 99(3):499–526, 2005.
- [24] Toby P. Breckon and Robert B. Fisher. Non-parametric 3d surface completion. In *3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, pages 573–580, Washington, DC, USA, 2005. IEEE Computer Society.
- [25] Pere Brunet and Isabel Navazo. Solid representation and operation using extended octrees. *ACM Trans. Graph.*, 9(2):170–197, 1990.
- [26] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76, New York, NY, USA, 2001. ACM.
- [27] Chun-Yen Chen and Kuo-Young Cheng. A sharpness-dependent filter for recovering sharp features in repaired 3d mesh models. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):200–212, 2008.
- [28] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. A finite element method for surface restoration with smooth boundary conditions. *Comput. Aided Geom. Des.*, 21(5):427–445, 2004.
- [29] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, New York, NY, USA, 1996. ACM.
- [30] James Davis, Stephen R. Marschner, Matt Garr, and Marc Levoy. Filling holes in complex surfaces using volumetric diffusion. In *3DPVT*, pages 428–438, 2002.
- [31] Ryo Furukawa, Tomoya Itano, Akihiko Morisaka, and Hiroshi Kawasaki. Improved space carving method for merging and interpolating multiple range images using information of light sources of active stereo. In *ACCV (2)*, pages 206–216, 2007.

- [32] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [33] Sarah F. F. Gibson. Using distance maps for accurate surface representation in sampled volumes. In *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization*, pages 23–30, New York, NY, USA, 1998. ACM.
- [34] Alexander Gress and Reinhard Klein. Efficient representation and extraction of 2-manifold isosurfaces using kd-trees. *Graph. Models*, 66(6):370–397, 2004.
- [35] Andre Gueziec, Gabriel Taubin, Francis Lazarus, and Bill Horn. Cutting and stitching: Converting sets of polygons to manifold surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):136–151, 2001.
- [36] Tong-Qiang Guo, Ji-Jun Li, Jian-Guang Weng, and Yue-Ting Zhuang. Filling holes in complex surfaces using oriented voxel diffusion. In *2006 International Conference on Machine Learning and Cybernetics*, pages 4370–4375, 2006.
- [37] Hugues Hoppe. Progressive meshes. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 99–108, New York, NY, USA, 1996. ACM.
- [38] Alexander Hornung and Leif Kobbelt. Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 41–50, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [39] Jiaya Jia and Chi-Keung Tang. Inference of segmented color and texture description by tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):771–786, 2004.
- [40] Mark W. Jones, J. Andreas Baerentzen, and Milos Sramek. 3d distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):581–599, 2006.
- [41] Tao Ju. Robust repair of polygonal models. *ACM Trans. Graph.*, 23(3):888–895, 2004.
- [42] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 339–346, New York, NY, USA, 2002. ACM.
- [43] Tao Ju and Tushar Udeshi. Intersection-free contouring on an octree grid. In *Pacific Graphics '06*, 2006.
- [44] Tao Ju, Qian-Yi Zhou, and Shi-Min Hu. Editing the topology of 3d models by sketching. *ACM Trans. Graph.*, 26(3):42, 2007.
- [45] Kolja Kahler, Jorg Haber, Hitoshi Yamauchi, and Hans-Peter Seidel. Head shop: generating animated head models with anatomical structure. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 55–63, New York, NY, USA, 2002. ACM.
- [46] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [47] Leif P. Kobbelt, Mario Botsch, Ulrich Schweincke, and Hans-Peter Seidel. Feature sensitive surface extraction from volume data. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 57–66, New York, NY, USA, 2001. ACM.
- [48] Vladislav Kraevoy and Alla Sheffer. Template-based mesh completion. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 13, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

- [49] Amitesh Kumar, Alan M. Shih, Yasushi Ito, Douglas H. Ross, and Bharat K. Soni. A hole-filling algorithm using non-uniform rational b-splines. In *Proceedings of 16th International Meshing Roundtable*, pages 169–182, 2007.
- [50] Bruno Levy. Dual domain extrapolation. *ACM Trans. Graph.*, 22(3):364–369, 2003.
- [51] Peter Liepa. Filling holes in meshes. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 200–205, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [52] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.
- [53] I. Makela and A. Dolenc. Some efficient procedures for correcting triangulated models. In *Proc. Solid Free form Fabrication Symposium*, pages 126–134, Austin, Texas, USA, 1993.
- [54] Takeshi Masuda. Filling the signed distance field by fitting local quadrics. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 1003–1010, Washington, DC, USA, 2004. IEEE Computer Society.
- [55] Robert Mencl and Heinrich Muller. Interpolation and approximation of surfaces from three-dimensional scattered data points. In *Dagstuhl '97, Scientific Visualization*, pages 223–232, Washington, DC, USA, 1999. IEEE Computer Society.
- [56] Claudio Montani, Riccardo Scateni, and Roberto Scopigno. A modified look-up table for implicit disambiguation of marching cubes. *The Visual Computer*, 10(6):353–355, 1994.
- [57] T. M. Murali and Thomas A. Funkhouser. Consistent solid and boundary representations from arbitrary polygonal data. In *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 155–ff., New York, NY, USA, 1997. ACM.
- [58] Minh X. Nguyen, Xiaoru Yuan, and Baoquan Chen. Geometry completion and detail generation by texture synthesis. *The Visual Computer*, 21(8-10):669–678, 2005.
- [59] Gregory M. Nielson and Bernd Hamann. The asymptotic decider: resolving the ambiguity in marching cubes. In *VIS '91: Proceedings of the 2nd conference on Visualization '91*, pages 83–91, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.
- [60] Fakir S. Nooruddin and Greg Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph.*, 9(2):191–205, 2003.
- [61] Stijn Oomes, Peter Snoeren, and Tjeerd Dijkstra. 3d shape representation: Transforming polygons into voxels. In *SCALE-SPACE '97: Proceedings of the First International Conference on Scale-Space Theory in Computer Vision*, pages 349–352, London, UK, 1997. Springer-Verlag.
- [62] Seyoun Park, Xiaohu Guo, Hayong Shin, and Hong Qin. Surface completion for shape and appearance. *Vis. Comput.*, 22(3):168–180, 2006.
- [63] Paresh S. Patel, David L. Marcum, and Michael G. Remotigue. Stitching and filling: Creating conformal faceted geometry. In *Proc. 14th International Meshing Roundtable*, pages 239–256, 2005.
- [64] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics*, 27(3), 2008. to appear.
- [65] Mark Pauly, Niloy J. Mitra, Joachim Giesen, Markus Gross, and Leonidas J. Guibas. Example-based 3d scan completion. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 23, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.
- [66] Jean-Philippe Pernot, George Moraru, and Philippe Veron. Filling holes in meshes using a mechanical model to simulate the curvature variation minimization. *Comput. Graph.*, 30(6):892–902, 2006.

- [67] Ron Pfeifle and Hans-Peter Seidel. Triangular b-splines for blending and filling of polygonal holes. In *GI '96: Proceedings of the conference on Graphics interface '96*, pages 186–193, Toronto, Ont., Canada, Canada, 1996. Canadian Information Processing Society.
- [68] Joshua Podolak and Szymon Rusinkiewicz. Atomic volumes for mesh completion. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 33, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.
- [69] Claudio Rocchini, Paolo Cignoni, Fabio Ganovelli, Claudio Montani, Paolo Pingi, and Roberto Scopigno. The marching intersections algorithm for merging range images. *Vis. Comput.*, 20(2):149–164, 2004.
- [70] S.J. Rock and M.J. Wozny. Generating topological information from a "bucket of facets". In *Solid freeform fabrication symposium proceedings*, pages 251–259, 1992.
- [71] Gerhard Roth and Eko Wibowoo. An efficient volumetric method for building closed triangular meshes from 3-d image and point data. In *Proceedings of the conference on Graphics interface '97*, pages 173–180, Toronto, Ont., Canada, Canada, 1997. Canadian Information Processing Society.
- [72] Member-Ryusuke Sagawa and Fellow-Katsushi Ikeuchi. Hole filling of a 3d model by flipping signs of a signed distance field in adaptive resolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(4):686–699, 2008.
- [73] Ryusuke Sagawa and Katsushi Ikeuchi. Taking consensus of signed distance field for complementing unobservable surface. In *3DIM*, pages 410–417, 2003.
- [74] Scott Schaefer, Tao Ju, and Joe Warren. Manifold dual contouring. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):610–619, 2007.
- [75] Andrei Sharf, Marc Alexa, and Daniel Cohen-Or. Context-based surface completion. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 878–887, New York, NY, USA, 2004. ACM.
- [76] Andrei Sharf, Thomas Lewiner, Gil Shklarski, Sivan Toledo, and Daniel Cohen-Or. Interactive topology-aware surface reconstruction. *ACM Trans. Graph.*, 26(3):43, 2007.
- [77] Chen Shen, James F. O'Brien, and Jonathan R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 896–904, New York, NY, USA, 2004. ACM.
- [78] Xuejun Sheng and Ingo R. Meier. Generating topological structures for surface models. *IEEE Comput. Graph. Appl.*, 15(6):35–41, 1995.
- [79] J. Spillmann, M. Wagner, and M. Teschner. Robust tetrahedral meshing of triangle soups. In *VMV'06: Proceedings of Vision, Modeling, Visualization*, pages 9–15, 2006.
- [80] L.S. Tekumalla and E. Cohen. A hole-filling algorithm for triangular meshes. Technical Report UUCS-04-019, University of Utah, 2004.
- [81] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318, New York, NY, USA, 1994. ACM.
- [82] Allen van Gelder and Jane Wilhelms. Topological considerations in isosurface generation. *ACM Trans. Graph.*, 13(4):337–375, 1994.
- [83] M. Varnuska, J. Parus, and I. Kolingerova. Simple holes triangulation in surface reconstruction. In *Proceedings of Algoritmy*, pages 280–289, 2005.
- [84] D. Veleba and P. Felkel. Detection and correction of errors in surface representation. In *The 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 65–72, 2007.
- [85] Joan Verdera, Vicent Caselles, Marcelo Bertalmio, and Guillermo Sapiro. Inpainting surface holes. In *Int. Conference on Image Processing*, pages 903–906, 2003.
- [86] Marc Wagner, Ulf Labsik, and Gunther Greiner. Repairing non-manifold triangle

- meshes using simulated annealing. *International Journal of Shape Modeling*, 9(2):137–153, 2003.
- [87] Jianning Wang and Manuel M. Oliveira. Filling holes on locally smooth surfaces reconstructed from point clouds. *Image Vision Comput.*, 25(1):103–113, 2007.
- [88] Zoe J. Wood, Hugues Hoppe, Mathieu Desbrun, and Peter Schroder. Removing excess topology from isosurfaces. *ACM Trans. Graph.*, 23(2):190–208, 2004.
- [89] Chunxia Xiao, Wenting Zheng, Yongwei Miao, Yong Zhao, and Qunsheng Peng. A unified method for appearance and geometry completion of point set surfaces. *Vis. Comput.*, 23(6):433–443, 2007.
- [90] Songhua Xu, Athinodoros Georghiadis, Holly Rushmeier, Julie Dorsey, and Leonard McMillan. Image guided geometry inference. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 310–317, Washington, DC, USA, 2006. IEEE Computer Society.
- [91] H. Yunbao, W. Qifu, H. Zhengdong, and W. Jianjie. Repairing incomplete measured data with a deformable model under constraints of feature shapes. *The International Journal of Advanced Manufacturing*, 28(1-2):92–100, 2006.
- [92] Nan Zhang, Wei Hong, and Arie E. Kaufman. Dual contouring with topology-preserving simplification using enhanced cell representation. In *IEEE Visualization*, pages 505–512, 2004.
- [93] Wei Zhao, Shuming Gao, and Hongwei Lin. A robust hole-filling algorithm for triangular mesh. *Vis. Comput.*, 23(12):987–997, 2007.