COMPUTER-AIDED DESIGN & APPLICATIONS
2018, VOL. 15, NO. 6, 780–795
https://doi.org/10.1080/16864360.2018.1466807

A slice-based algorithm for automatic and feature-preserving hole-filling in a CAD mesh model

Nepal Adhikary and B. Gurumoorthy

Indian Institute of Science, India

ABSTRACT

This paper describes an automatic feature-preserving mesh hole filling algorithm to repair a mesh model, obtained by tessellating a 3D CAD model, with arbitrary holes or gaps. Missing triangles in a mesh model result in such holes or gaps. The proposed algorithm fills the hole while maintaining curvature continuity across the boundaries of the hole. In addition, the algorithm is able to identify missing features such as edges or vertices and reconstruct these. A composite surface patch bound by the boundary of the hole is first constructed and then tessellated to be in conformity with the original mesh to obtain the repaired mesh model. The input mesh remains unchanged in regions away from the holes so any features in the input mesh are preserved. The reconstruction also works in the presence of holes with islands. Algorithm can also fill holes in real world mesh model. Focus is for small holes that result in one missing corner per hole. Results from an implementation tested on various mesh models are presented.

KEYWORDS

CAD mesh models; slicing; hole-filling; reverse engineering; feature-preserving; corner recovery; surface fitting

1. Introduction

Polygon mesh has become a popular choice of representing 3D shapes for applications ranging from simulation and analysis, product design & manufacturing, bio-medical computing, gaming and animation, scientific computing, etc. As the triangle is the simplest representation of polygon and the facets of any polygon meshes can be triangulated, in the rest of the paper unless otherwise stated, it is assumed that the mesh model is made up of triangle facets.

Holes and ***gaps*** (figure 1a, 1b) are two common **artifacts** in a mesh model. Holes are missing triangles on the surface of a mesh model; whereas gaps are cracks [23] in between neighboring facets that should have been seamlessly connected. Mesh models are typically obtained from one of two broad methods – scanning a physical prototype of an object or by tessellating a digital model (usually constructed in a CAD system)[5]. In this paper, though the focus is on the latter, the proposed algorithm is able to fill holes also in scanned models. As pointed out in the recent survey of hole filling by Attene et al.[5] the process of tessellation of CAD models can result in mesh with problems. Tessellation algorithm usually creates separate meshes for each surface patch and combines them. During combining, algorithm usually fails to match-up all the triangles along the patch boundaries; thus producing gaps holes, overlaps and self-intersections [5,6] in the mesh model. Gaps in a mesh model may also occur because of translation of geometric definition from one CAD modeling system to another system having different point coincidence and angular deviation tolerances [1]. Holes may also be introduced in a mesh model during the suppression or removal of features in the part to enable or enhance the task of analysis or simulation [16]. A typical mesh model as generated from a CAD model, therefore, may have gaps and holes with or without islands. These gaps, holes and islands may also present in a mesh model obtained from scanning.

Holes and gaps are usually very small [31] compared to whole mesh model, but their sizes may vary depending on the acquisition procedure. Holes introduced during feature suppression are proportional to the size of the features that are suppressed or removed. It is clear therefore that there are no clear bounds on the size of a hole. When holes are large, it is better to reproduce some known pattern from within the patch or from shape repository to obtain more realistic results than filling those holes by smooth patches. This is because; filling large holes with smooth patches may lead to completely different model. Hence, hole-filling by surface patch must be restricted for smaller holes [5].

The mesh model may also have self-intersecting facets and other topological and geometric inconsistencies like incompatible normal of triangles [9] shown in figure 1c.

CONTACT Nepal Adhikary nepal_adhikary@yahoo.com; B. Gurumoorthy bgm@iisc.ac.in

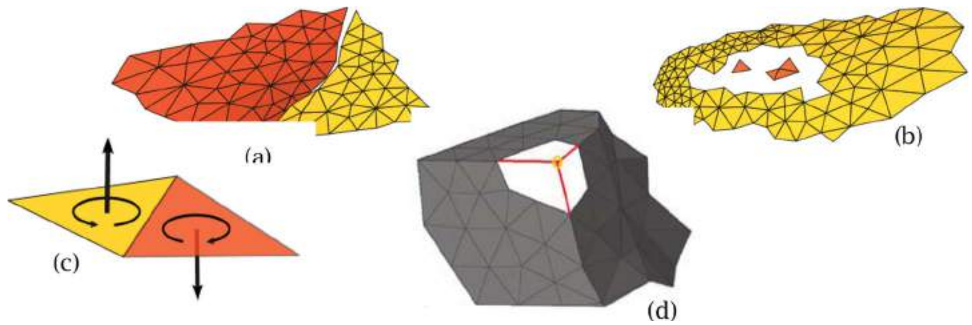

Figure 1. (a, b, c) Gap, hole with islands and inconsistent normal [5], (d) Missing red edges and yellow missing corner.

Hence, it is necessary to repair the acquired mesh model before using it in any application. Incompatible normals are fixed in a rather straightforward way using the Moebius rule [9]. A number of algorithms have been developed for resolving self-intersecting facets [4] and closing the gaps [6] in mesh model. Hole filling is also required in pre-processing mesh models for certain type of analysis such as acoustic calculations where a continuous radiating surface is required. This paper addresses the problem of repairing holes in a mesh model by filling them.

There are mainly two types of algorithm available to repair a hole. These are either volume based [22, 24] or surface based [25, 40] approach. Main drawback of volume based method is the loss of geometric details of the original model. This makes volume-based scheme unsuitable for CAD applications where fidelity with the original model is the basis of acceptance. Even most of the surface based approaches don't preserve the underlying geometry [25] of the mesh model. Holes become complex, if it contains many internal boundaries (islands, figure 1b). In the present context of mesh models obtained from CAD models, filling holes in the presence of islands also arises in the context of feature suppression. It is important to preserve the features in the mesh both in the vicinity of the hole and elsewhere. Presence of islands complicates hole repairing procedure and disqualifies many of the hole-filling methods proposed in the literature [5]. There are a few interactive algorithms available [27, 37] that repair holes with islands but none of them preserve input details (hole-boundary edges and island meshes) of holes and islands. So far there is no surface based automatic algorithm available that repairs holes in the presence of islands [5, 23] and preserves input geometric details.

An automatic surface-based procedure to fill holes in the mesh in presence of missing entities and islands that preserves the underlying geometry is described in this paper. Algorithm identifies the holes and islands and, extracts possible missing edges and missing corners (figure 1d) associated with each hole. Curvature continuous section curves are then obtained by slicing the triangles surrounding each hole. These section curves are then lofted to create surface patches used to fill the hole or part of it. The surface patches corresponding to all the portion of the hole are stitched and trimmed by the boundary edges of the hole and its islands. Finally, this trimmed patch is triangulated and the mesh obtained is merged with the main mesh model. This algorithm is illustrated in Figure 2.

In Figure 2a, a mesh model with a hole is shown. After segmentation of hole-boundary edges into two segments, a missing edge of this hole 'LM' (portion of AB) is regenerated. All these are shown in Figure -2b. Figure 2c to Figure 2f describe the procedure of generating curvature continuous section curves. These curves are used to generate lofted surface require to cover portion of a hole. A lofted surface is shown in Figure 2g. All these surfaces are stitched together to make a composite surface. This composite surface is then trimmed by hole-boundary edges and shown in Figure 2h. Tessellated and conformal representation of trimmed composite surface is shown in Figure 2i. Finally, conformal mesh is stitched with the input mesh along hole-boundary edges. This is shown in Figure 2j. The algorithm does not depend on the type of mesh model, whether it is coarse or dense mesh model. Further the algorithm does not impose any restriction on holes with islands; rather it utilizes the information from islands to better capture the underlying geometry of the mesh model. Rest of the paper is organized as follows: Section 2 contains literature survey, definition and terminology related to this paper is presented in Section 3. Section 4 describes the proposed algorithm. Results and discussion are presented in section 5. Finally, section 6 concludes the paper after identifying future work.

2. Literature review

Many hole-filling methods have been proposed in literature. Perez *et al.* provide a very detailed review of

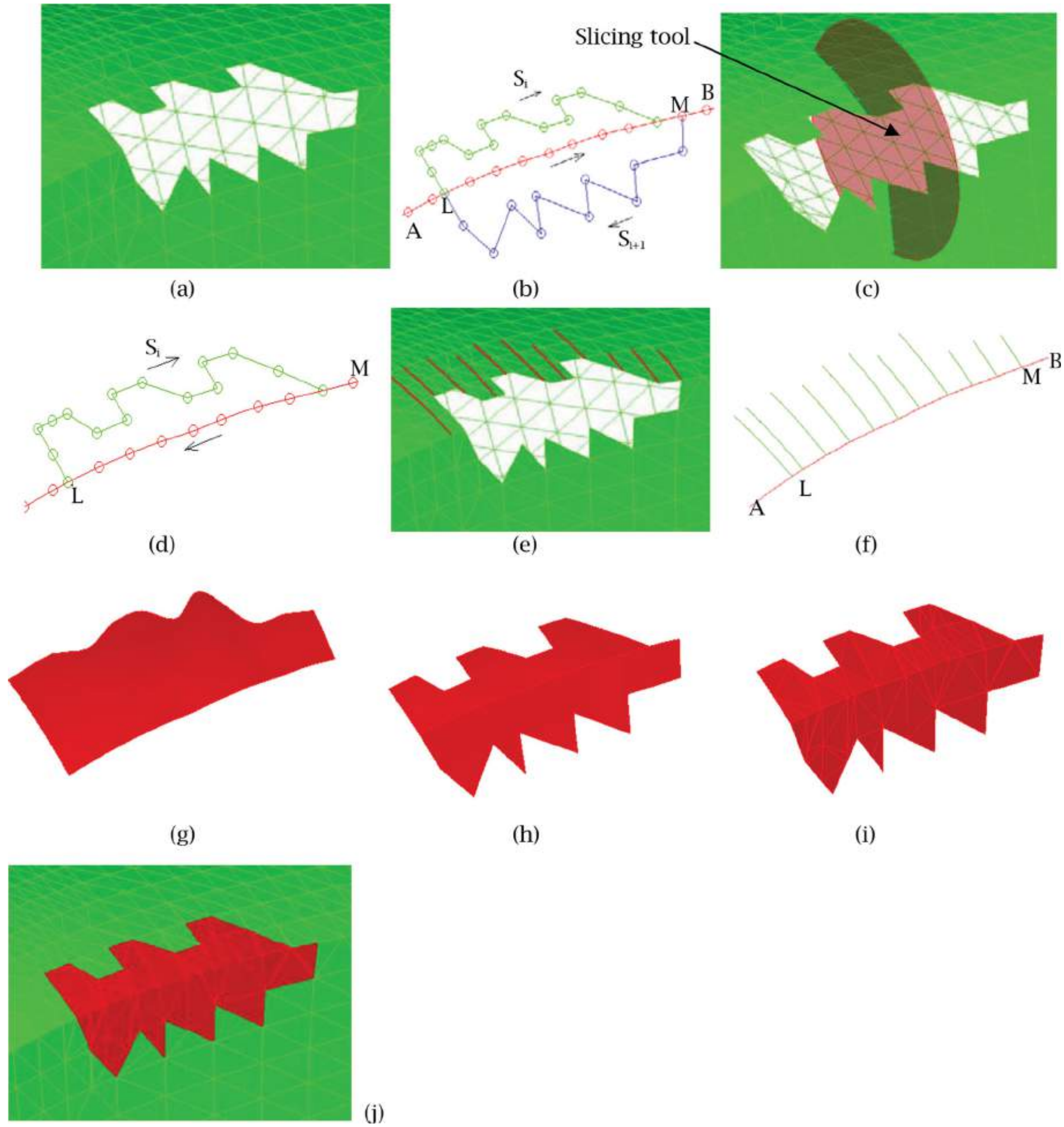

Figure 2. Illustration of algorithm, (a) input mesh with hole, (b) hole-boundary segments Si, Si + 1 and regenerated missing edge LM, (c) input mesh and slicing tool, (d) a portion of the hole to be sliced, (e) input mesh and slices, (f) curvature continuous section curves in green generated by using points from slices and discrete curvature continuous points determined from the end point of each slice, (g) the lofted surface generated by interpolating section curves, (h) composite surface after trimming by boundary edges of holes and islands followed by stitching, (i) tessellated and conforming representation of composite surface, (j) input mesh model in green and output mesh patch filling the hole in red.

hole-filling techniques [31]. While they have classified the methods used in the literature into five categories, in this paper the methods in the literature are broadly classified as either volumetric or surface-oriented approach [9, 23]. The difference is because this paper is focused only on filling holes in mesh models.

2.1. Volumetric approach to hole filling

In volume based algorithms, input mesh models are converted into an intermediate representation by discrete volumes called voxels and then different algorithms are used to fill the holes. After filling those holes, output surface mesh models are extracted from the discrete volumetric representation [8, 12, 24, 29]. Volume based scheme is robust. It can resolve various types of geometric errors including self-intersections. Major disadvantage of volumetric scheme is the loss of geometric details of the input mesh model. This is due to the reconstruction of surface mesh from intermediate representation resulting in loss of sharp edges and corners on input model.

2.2. Surface-oriented approach to hole filling

In this scheme, holes are filled directly by adding triangles based on the existing triangles around the hole. The main advantage of surface based algorithms is that it modifies a mesh model locally around a hole. Gregory and Zhou [17] proposed an algorithm for filling polygonal hole surrounded by bi-cubic surface patches for smooth model. A basic hole-filling algorithm by joining matching vertices on adjacent boundaries and optimal triangulation for unmatched parts was proposed by Barequet and Sharir [6]. Algorithm does not preserve underlying feature geometry. Jun [25] proposed a hole-filling method by subdividing a complex hole (without island) into multiple simple planar holes and filling each by adding triangles. This algorithm too does not preserve underlying feature geometry and requires smoothing to improve the quality of output. Liepa [28] extended the approach of Barequet and Sharir [6] by using initial triangulation as an approximation followed by refinement and fairing. Algorithm does not consider the cases of missing corners and missing edges. Zhao et al. [40] constructed hole-filling patches by an advancing front mesh technique (AFM), derived approximate normal for the boundary region of input mesh, and finally improved the quality of meshes by solving Poisson's equation for the normals. Kumar et al. [26] proposed NURBS based technique to create intuitive geometry to fill simple holes. Algorithm is not suitable for a hole that causes a corner or mesh-boundary edges to vanish. Error of approximation will be very high for those cases. Wang and Oliveira [38] proposed an algorithm by using the vicinity of hole to interpolate the missing portion of the surface using moving least squares (MLS) technique. The proposed algorithm works well if the areas inside and around the holes are locally planar. Filling holes based on grey prediction technique for scanned dental models is proposed by Wang and Hung [39]. Results largely depend on the prediction-controlled variables such as normal vectors. They concluded by mentioning the need of curvature as prediction variable for further improving the hole-filling quality. Pernot et al. [32] addressed a hole-filling method by minimizing the curvature variation between the surrounding and inserted meshes for simple holes (without island). Algorithm depends on clean hole-boundaries. Arbitrary or zigzag boundaries need cleaning and smoothing before starting their algorithm. Algorithm gives good results only when the gap to be filled is shaped like a disk and performs poorly for holes have missing features (missing edges or missing corners).

An iterative context-based hole-filling approach is proposed by Sharf et al. [34]. To determine best matching patches, algorithm copies and adapts part of the surface that respects the geometric characteristics around the hole, and then, fits those patches by aligning them with the surrounding surface. Harary et al. [18] extended the concept of context-based surface completion by introducing coherence objective while filling a hole. Coherence objective is achieved by minimizing both global and local errors for the hole to be filled. Algorithm utilizes patch similarity metric, which takes into account both local and global shape attributes. A mesh completion algorithm is proposed by Bendels et al. [7] based on local neighborhood information of the holes and therefore making the filling insensitive with respect to similarity transformations. Harary et al. [19] proposed an interactive surface completion technique. In their approach, sharp features are identified based on user assisted feature points, two of them are at the start and other two are at the end of the feature curve. These sharp features, sub-divide the hole into multiple sub-holes to be filled. Authors did not consider the case of missing corner feature.

Li. et al. [27] proposed an interactive hole filling method. Algorithm subdivides hole with island into sub-holes with no island by using features curves. These feature curves are identified (generated) manually by selecting points around the hole as well as from the islands. Holes are filled with the help of smooth patches created for sub-holes by interpolating points around the influencing area of the hole. Quality of output depends on the coarseness of mesh model. Algorithm cannot handle holes that cause a corner in the model to vanish. It does not preserve original boundary edges of the hole or island in the final output mesh model. Also it does not preserve original triangles of the islands. Chonga et al. [11] proposed an algorithm where islands are co-related with holes interactively and hole with islands is converted to a hole with no islands by joining boundary of islands with the hole-boundary along the shortest path. Algorithm then fills these simple holes by adding triangles. Wang et al. [37] proposed another algorithm where islands are identified and co-related with hole interactively for guaranteed reconstruction of hole-filling patch. In short, there is no known surface based algorithm available that can repair a complex hole (hole with islands) automatically. As volume based scheme are robust and can resolve various errors, issues related to islands gets resolved automatically in volumetric schemes [20, 21] during reconstruction of surface mesh from intermediate representation.

Poisson based reconstruction tool is proposed to fill holes by Centin et al. [10]. Algorithm does not preserve input boundaries, geometric details both for hole and islands. The Poisson reconstruction technique assumes that the object to be repaired is bound by a smooth closed surface. This assumption does not hold for most engineering objects, the main focus of the paper. Fortes et al. [15] proposed an algorithm to fill hole by reconstructing a smooth Powell–Sabin spline in and outside the domain of hole with a prescribed or approximated volume condition inside this hole. How this procedure will work for trimmed patches (non-four sided) is not explained. Curve shortening algorithm to fill hole with an island is proposed by Altantsetseg [3]. They need to modify all input boundaries (and successive internal points added) to generate smooth curves. Finally, they fill the hole by Delaunay triangulation, which does not maintain conformity with input meshes. Curve shortening algorithm restricts the number of islands to one per hole. It is a major limitation of the proposed algorithm. Also, it is not clear if their algorithm can handle features removed by the presence of a hole (such as missing corners).

An automatic hole-filling algorithm for holes, without islands, was presented by Wang et al. [36] for CAD models. Main contribution of the paper is to regenerate missing feature curves and missing corner associated with a hole ***without island*** in CAD mesh model. Main limitation of their algorithm is its robustness for regenerating missing curves and corners. While regenerating missing curves, algorithm uses ordered double normal, Gaussian mapping, convex/concave analysis, energy minimization and number of heuristics for matching feature vertices. Matching approach is suitable for regular and simple features. Heuristics based on double normal and ordering of vertices based on orientation is not robust for topologically complex holes in high curvature region. Algorithm generates interpolating points for missing feature curve but the order of continuity maintained is not mentioned. The algorithm detects missing corners based on energy minimization and regenerates them by using missing curves. Their algorithm will be able to identify corner correctly only when the missing corner is surrounded by planar surfaces, since interpolated missing curves naturally meet at a corner in this case. Another limitation of the algorithm is that if a corner is surrounded by high curvature hole-boundary, then the missing corner cannot be identified.

Available commercial software, such as MeshMixer and GeoMagic, provide interactive solution to repair holes in a mesh model. User has to specify the type of geometry, flat or curved, to fill holes. Triangles around the hole-boundary get modified. They do not preserve the input mesh. Also, it is not clear if these products can address islands in the hole as they do not make any claims about this.

To overcome these limitations, an automatic feature preserving slice based hole-filling algorithm has been proposed in this paper which can regenerate all missing curves and missing corner associated with a hole by using geometric and topological information while filling that hole even in presence of islands.

3. Definition and terminology

The following terminologies are used in this paper:

A cross section generated by slicing the triangles of a mesh model with a plane is called slice. ***Slices*** obtained by slicing a portion of mesh model of Figure 3a, using multiple offsets of the slicing plane (shown in red in Figure 3a), are shown in Figure 3b. In Figure 3b, each slice shown consists of one or two sets of connected edges. In this paper, slices of the form shown in Figure 3b will be used by the hole-filling algorithm.

A ***region*** [2] is a part of mesh where dihedral angle along their common boundaries is less than MIN_THRESHOLD or more than MAX_THRESHOLD [2]. ***Region boundaries*** are the connected set of edges of a region where dihedral angle is greater than MIN_THRESHOLD or less than MAX_THRESHOLD. The thresholds are chosen based on typical tessellation error in obtaining the mesh model [2]. A region corresponds to a face in B-rep model. Region and region boundaries are shown in Figure 5 and Figure 6. A curve is said to be ***discrete curvature continuous*** (Figure 4) at a point, if the forward and backward difference of tangents at that point on the discrete curve are same. Forward discrete curvature at a point Pi on the discrete curve is defined by [9, 13]:

$$K_{if} = (T_{i+1} - T_i)/(D_{i+1,i} - D_{i,i-1}) \tag{1}$$

where T_i and T_{i+1} are the tangents at points P_i and P_{i+1} respectively, and $D_{i+1,i} = \|P_{i+1} - P_i\|$

Tangent at point P_i is given by

$$T_i = (P_i - P_{i-1})/|P_i - P_{i-1}| \tag{2}$$

Similarly backward discrete curvature at P_i is defined by

$$K_{ib} = (T_i - T_{i-1})/(D_{i,i-1} - D_{i-1,i-2}) \tag{3}$$

A hole may have one or many boundaries. Each boundary is a connected set of edges in a region or across

Table 1. Classification of hole-filling algorithms.

Reference	Classification	Algorithm	Limitations
Bischoff et al.[8]	Volumetric scheme	Voxel computation	Loss of geometric details
Ju [24]	Volumetric scheme	Voxel computation	Feature details not preserved
Nooruddin and Turk [29]	Volumetric scheme	Voxel based approach	Loss of geometric details
Gregory and Zhou [17]	Surface based	Spline based algorithm	Hole must be surrounded by rectangular smooth patches.
Barequet and Sharir [6]	Surface based	Filling by joining matching vertices of a gap	Underlying geometry not preserved, cannot detect gaps for all configurations.
Jun [25]	Surface based	Filling by adding extra triangles followed by smoothing	Does not preserve underlying feature geometry, islands not handled.
Liepa [28]	Surface based	Approximation followed by refinement and fairing	Missing corner and missing edges not considered
Zhao et al. [40]	Surface based	Advancing front mesh technique (AFM) followed by smoothing	Missing features as well as islands not considered
Kumar et al. [26]	Surface based	NURBS based technique	Not suitable for complex hole with missing features
Wang and Oliveira [38]	Surface based	Interpolation using moving least squares (MLS) technique	Works well for locally planar hole
Wang and Hung [39]	Surface based	Grey prediction technique	Result depends on types of prediction control variable
Pernot et al. [32]	Surface based	Curvature optimization	Depends on clean hole-boundaries, performs poorly for holes have missing features, does not consider islands
Sharf et al. [34].	Surface based	Mesh completion by point-sampled surfaces	Needs complete remeshing, not directly applicable to mesh repairing
Harary et al. [18]	Surface based	Mesh completion by point sample patches	Implied remeshing, does not preserve original features
Bendels et al. [7]	Surface based	Mesh completion by point sampled surfaces	Not suitable for mesh repairing problem
Harary et al. [19]	Surface based	Interactive Mesh completion	Interactive method, missing corner not considered.
Li. et al. [27]	Surface based	Interactive spline based approach	Interactive procedure, does not consider missing corners, depends on coarseness of mesh
Chonga et al. [11]	Surface based	Interactive hole filling	Does not consider missing features
Wang et al. [37]	Surface based	Interactive hole filling	Does not consider missing features
Centin et al. [10]	Surface based	Poisson based tool	Hole should bound by smooth surface. Algorithm does not preserve geometric details.
Fortes et al. [15]	Surface based	Powell-Sabin	Hole should be inside a four sided patch.
Altantsetseg et al. [3]	Surface based	Curve shortening method	Does not preserve geometric details. It cannot handle missing corner.
Wang et al. [36]	Surface based	Advancing front method (AFM)	Planar hole, robustness issue.

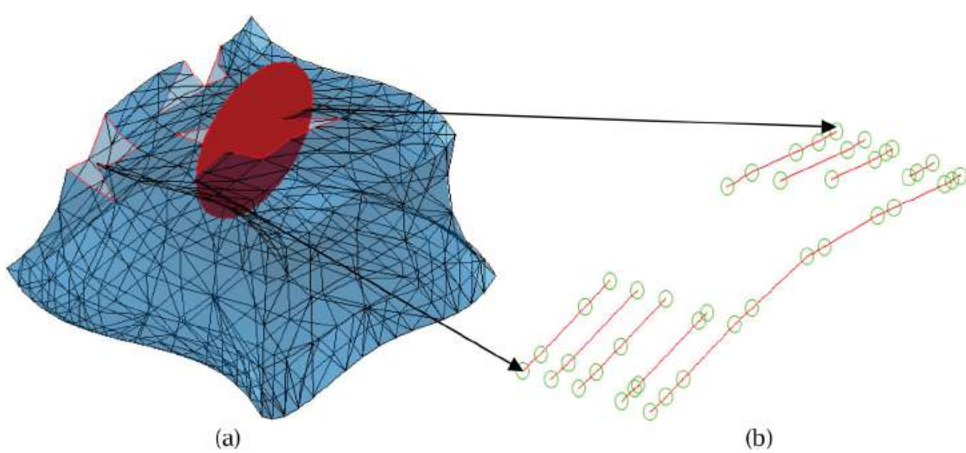

Figure 3. (a) Mesh model with holes and slicing plane in red, (b) slices – enlarged view.

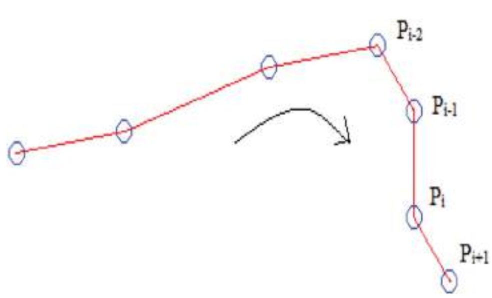

Figure 4. Discrete curve.

regions such that each edge of that boundary belongs to only one triangle [25, 32]. The outermost boundary is called as **hole-boundary,** while all other boundaries inside it are known as **island-boundaries**. If a hole lies inside a region, it is classified as **simple-hole** or ***single-segment*** hole. In this case, boundary-edges of that hole constitute single-segment. If a hole is spread across regions, boundary-edges will be segmented into *multiple-segments* and hole will be classified as **multi-segment hole**. Both single-segment and multi-segment holes are

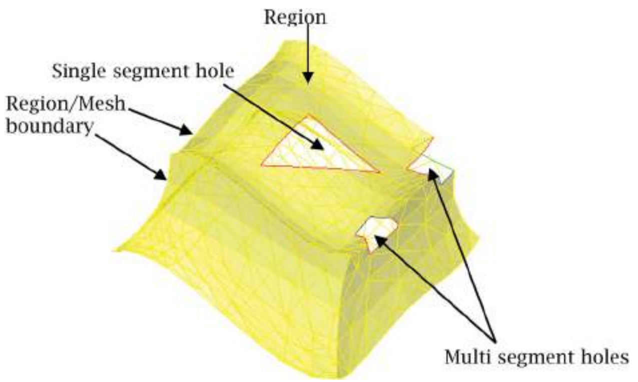

Figure 5. Single and multiple segment holes.

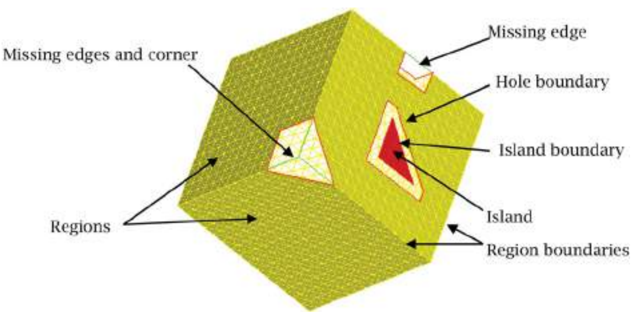

Figure 6. Missing features in a hole and island inside a hole.

illustrated in Figure 5. Triangles surrounded by ***island-boundary*** constitute a region called *island*. Island inside a hole is shown in Figure 6. If hole-boundary edges are spread across regions, the hole will have missing features like missing edges or missing corner point. Missing edges and corner points are referred to as missing features in a hole. Examples of holes with missing features are shown in Figure 5 and Figure 6.

3.1. Data structures used in the algorithm

Following are the major data structures used in the algorithm.

Hole data structure contains the following information – hole type, e.g. single-segment or multi-segment holes, list of boundary edges, the min-max box defined by the vertices on the boundary of the hole (hole-enclosing-box), list of islands contained in the hole and the list of segments forming the boundary of the hole.

Hole {
- hole-type { single-segment, multi-segment-ordinary, multi-segment-corner};
- List of boundary edges;
- hole-enclosing-box;
- List of islands;
- List of segments of this hole;

};

Segment data structure contains the following information: list of vertices from start of the segment to its end, start vertex of the segment, discontinuous region boundary edge of the mesh model attached to start vertex, end vertex of the segment and discontinuous region boundary edge of the mesh model attached to end vertex. For a single-segment hole, start and end vertices will be same as there will not be any mesh boundary edge interacting with hole-segment.

Segments {
- List of vertices;
- Start vertex of this segment;
- Edge on the boundary of a region incident on the start vertex;
- Missing curve at start;
- End vertex of this segment;
- Edge on the boundary of a region incident on the end vertex;
- Missing curve at end;

};

4. Proposed solution

This section describes the procedure of automatic detection and filling of holes in a mesh model. Following are the main steps in the hole filling procedure:

I. Identification of holes and islands and cluster islands of a hole
II. Segmentation of hole-boundary
III. Reconstruction of missing features
IV. Generation of composite surface
V. Trimming of composite surface
VI. Faceting and sewing of composite surface

Above steps are explained in detail in the following subsections.

4.1. Identification of holes and islands and cluster islands of a hole

Holes and islands (figure 2a, figure 5 and figure 6) in a mesh model are identified by first identifying their bounding contours and then classifying each contour as hole or island. Here, contour is defined as the connected set of ordered edges that are incident on one triangle. After identifying and then classifying all the above contours as holes and islands, algorithm associates a set of islands that belongs to a particular hole (outer most hole). Following is the algorithm to classify all contours as holes and islands and then cluster as set of islands of a hole.

For I = 1 to number of contours identified above **{**

Call *PlaneOfMaximumSpan* and then *ReferenceEdge* for the Ith contour
(Reference edge is the edge closest to being orthogonal to normal of the plane of maximal span)
CALL *DevelopContourOnPlaneOfMaximumSpan* for the Ith contour and set it as **D-Contour**
CALL *DevelopATriangleOnPlaneOfMaximumSpan* for a triangle incident on above reference edge.
(Algorithms to develop a contour and a triangle are described below)
SET the normal of the plane of maximum span is same as the orientation of the developed triangle (based on the ordering of its vertices)
CALL *ClassifyCentroid* of the developed triangle with respect to **D-contour {**

If *centroid* is INSIDE the **D-contour**, original contour is an ***island*** and the corresponding D-contour is classified as developed island or **D-island**;
Else original contour is classified as a ***hole*** and **D-contour is** classified as developed hole or **D-hole**

}

}

For all D-holes**{**

For all D-islands**{**

If D-island is on same plane as D-hole and D-island is INSIDE D-hole,

Attach the corresponding contour as island belonging to hole

Else if D-island is NOT OUTSIDE visual hull of D-hole **{**

If dist (D-island, D-hole) is less than HOLE_SIZE

Attach the corresponding contour as island belonging to hole

}

}

}

Here visual hull is the projection of D-island on D-hole. ACIS [1] utilities are used to determine the projection of planar D-islands on to a planar D-hole.

4.1.1. Development of a contour

In this paper contours of holes and islands are developed by projecting each successive ordered edge in the contour onto its plane of maximum span and then pushing/pulling those projected edges onto a reference circle defined for each contour. The reference circle for each contour is obtained as follows. Center of the circle is the projection of the centroid of the contour on its plane of maximum span. Radius of the circle is determined such that the perimeter of the circle is the same as the perimeter of the contour. The edges of the developed contour are adjusted in such a way that each projected edge becomes a chord of the reference circle. Algorithm maintains the length of each chord to be the same as the length of the corresponding original edge in the contour. After the development, each contour becomes an approximate polygon of its reference circle. These approximate polygons are termed as D-contours and are free of self-intersection errors that could be present in the projections.

4.1.2. Development of a triangle

To develop the triangle with one edge incident on the reference edge, algorithm uses the corresponding edge (developed-edge) in the D-contour and places the third vertex of the triangle on either left side or right side of D-edge based on the orientation of original triangle. While placing the third vertex, algorithm also maintains the length of other two edges of the developed triangle same as the length of the edges in the original triangle in mesh.

The above check is also used to associate any holes that are inside islands inside a hole. These internal holes however have no further role in the hole-filling algorithm except in the final step of trimming the reconstructed surface corresponding to the holes. The hole-filling algorithm works with holes that are contained in any island within a hole. The contour of a hole need not be planar or even nearly planar to be able to be projected on a plane. Proposed algorithm is capable of developing any hole on a curved/doubly curved surface onto a plane.

Influencing area [27] of a hole is defined by the triangles around that hole. The triangles of the influencing area of a hole are determined by increasing the size of the bounding box of the hole along the three dimensions by 10% of the length of its largest diagonal. Triangles contained in this enlarged box form the influencing area and are used to guide the filling of that hole.

4.2. Segmentation of hole-boundary

This section describes the procedure to identify the portions of hole-boundary that belong to single region. Boundary of a hole is subdivided into multiple segments

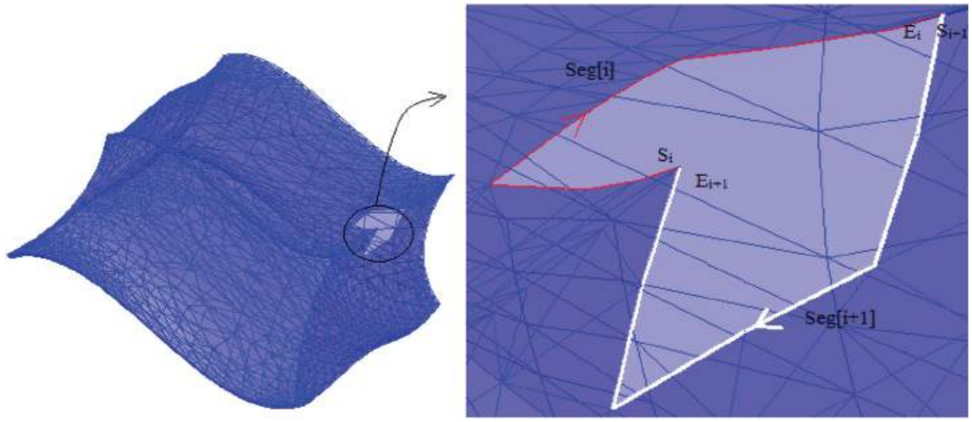

Figure 7. Segmentation of boundaries of a hole.

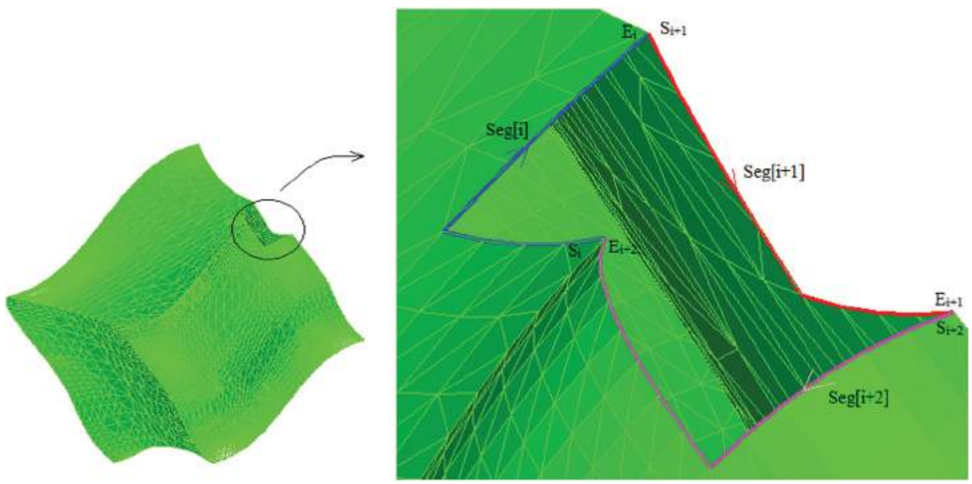

Figure 8. Segmentation of boundaries of corner hole.

such that each segment lies in one region. For each vertex on the boundary of the hole, the dihedral angle for edges that are incident on that vertex and not on the boundary of the hole is first determined. The boundary of the hole is then segmented at vertices where an edge has a dihedral angle between *MIN_THRESHOLD* and *MAX_THRESHOLD* [2]. Two segments of a hole-boundary are shown in the Figure 2b. A multi-segment hole is shown in Figure 7, where one segment is shown in red and the other in white. Figure 8 shows another multi-segment corner hole that has been segmented into three parts.

4.3. Reconstruction of missing features

If a hole has boundary segments across different regions, it indicates the presence of missing features. This section explains the procedure to reconstruct missing boundaries.

Missing edges are identified first followed by the construction of missing corners if any. The reconstruction starts at one end vertex of a hole segment identified in the previous section. Points on the missing boundary are traced from this point. The first such point is determined so as to maintain curvature continuity with the mesh boundary (edge) responsible for segmenting the hole boundary at this point. Subsequent points on the missing boundary are determined so as to maintain continuity of curvature with the previous point determined. Curvature continuity is maintained by equating the expressions for backward and forward discrete curvature at the current point (refer Section **Terms and Definitions**). Tracing terminates when a new point determined lies outside the bounding box of the hole. This procedure is repeated for all the segments of the hole-boundary.

The points generated on the missing boundary are then interpolated to obtain a curve. This procedure is repeated for all segments identified. When the number of missing edges identified in this step is more than one, presence of a corner point is indicated. The missing corner point should be the point of intersection of the missing edges. However as the curves are not exact, extracted

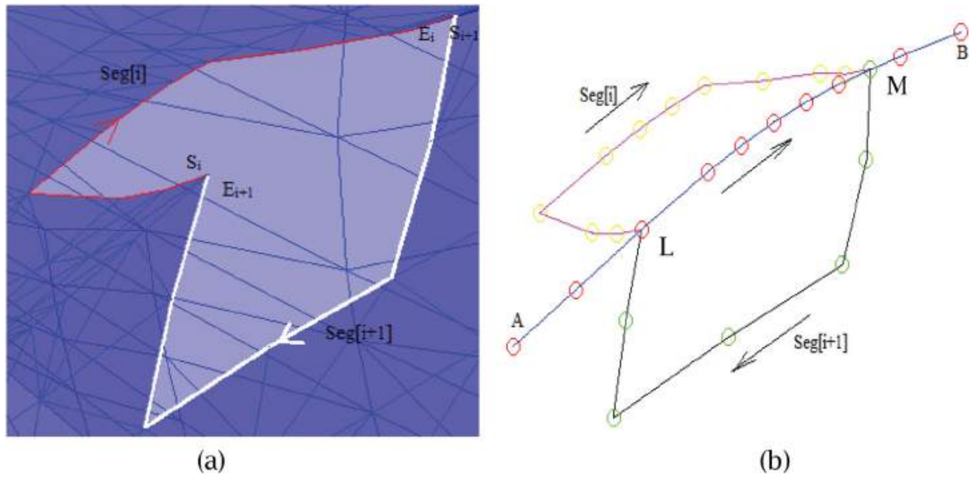

Figure 9. (a) Multi-segment hole, (b) segments and missing discrete curve LM.

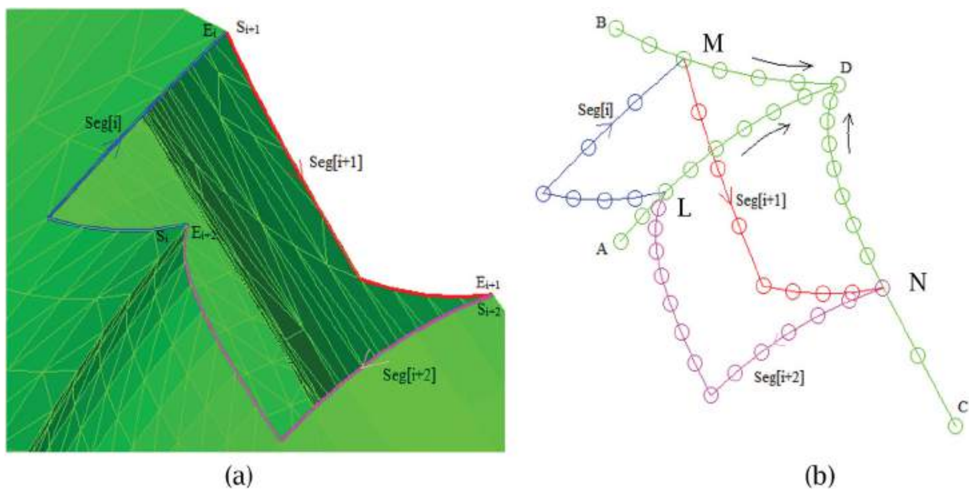

Figure 10. (a) Multi-segment corner hole, (b) segments, missing discrete curves - LD, MD, ND and missing corner D.

missing curves come near to a corner, but do not meet exactly at a point. The corner point is determined by finding the closest point on each extracted missing edge to the other extracted edges and taking the average of these points. Each missing edge is then modified to have this point as the other end point by interpolating old points and new corner point. Figure 9 and Figure 10 show cases where the hole boundary consists of many segments and the hole contains portions of a boundary edge and a corner point. After this step, all missing boundaries as well as corner points for holes that belong to different regions are obtained. In Figure 2b, LM represents the regenerated missing edge of a hole.

4.4. Generation of composite surface

At the end of the previous step, boundary of each portion of the hole is available. In this step each portion is processed to obtain a smooth surface patch representing that portion. As shown in Figure 11a, each portion of the hole is represented by the bounding edges (including the missing ones determined in the previous step). First, section curves that define the surface corresponding to this portion are determined. The section curves are obtained in a manner similar to that used to determine the missing boundaries. Slice points are generated by slicing the influencing triangles around the portion of hole under consideration and the triangles of the islands associated with the hole. Figure 11b shows the plane used to slice the influencing region and Figure 12a shows the slice points obtained. Section curves are traced from each slice point on the boundary curve so that each new point maintains curvature continuity with the previous point. The first point traced ensures continuity of curvature with the mesh in its vicinity (see Figure 12b). As in the case of determining missing boundary, curvature continuity is maintained by equating the curvature at the previous point, obtained by backward and forward difference.

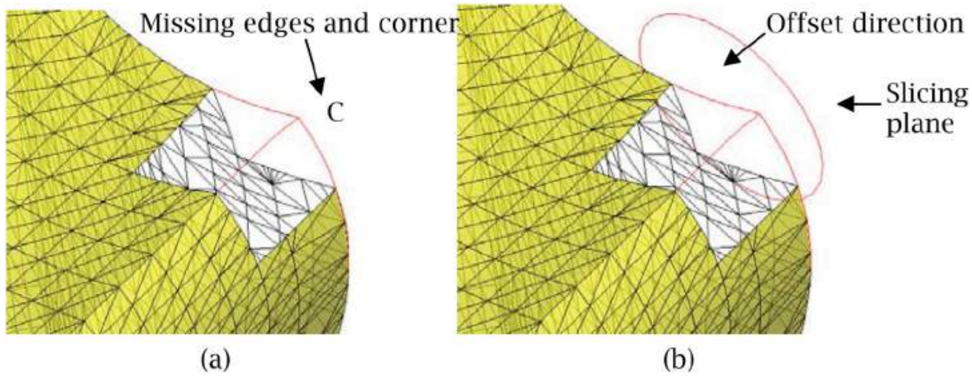

Figure 11. (a) Corner hole, missing edges in red and missing corner C, (b) corner hole, missing features and slicing plane in red.

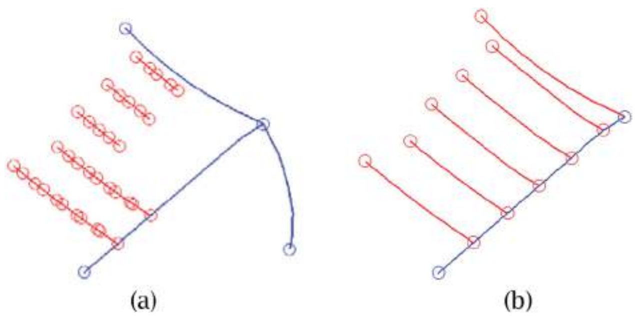

Figure 12. (a) Red partial-slices and blue missing edges, (b) red smooth section curves and blue guide curve.

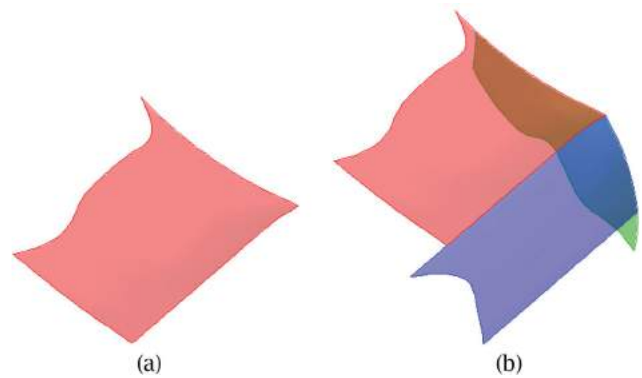

Figure 13. (a) Lofted surface (b) composite surface.

Tracing of new points starts from a slice point on the hole-boundary and moves to the other end of the slice on the boundary (either original or regenerated boundary). If slice contains points from islands, algorithm starts from a slice point on the hole-boundary and moves towards the start slice point on island boundary and then traces again from end slice point on island boundary and moves towards the end point on hole-boundary. For multiple islands, tracing follows the pattern: hole to island, island-to-island and island-to-hole boundary. This pairing is possible as all the boundary edges of holes and islands are identified.

The section curves obtained are lofted to obtain a surface patch corresponding to the portion of the hole as shown in Figure 13a. Once the surface patch is obtained for all the portions of the hole, a composite surface is obtained by stitching all the surface patches generated from each portion of the hole. This composite surface will cover the entire hole including islands associated with this hole (see Figure 13b). Figure 2c, 2d, 2e, 2f and 2g also describe this step.

The direction of advancement of the slicing plane is set to be along the longer edge of the bounding box, of the portion of the hole that lies in single region. However, it is believed that this choice can be arbitrary along either direction. The slicing tool is actually a plane and therefore only the normal and a point on the normal is enough to define the tool. The point is obtained by offsetting the tool along the normal by a specified distance relative to the length of that longer edge of the box.

4.5. *Trimming of composite surface*

The composite surface patch obtained in the previous step is shown in Figure 13b. In this step the composite surface is trimmed by the boundary edges of the holes and islands. This is accomplished by using API of ACIS [1]. At the end of this step a *trimmed-composite surface patch* that fills the entire hole is obtained. Steps of trimming are shown in Figure 14 as well as in Figure 2h.

4.6. *Faceting and sewing*

Figure 15 shows the trimmed composite patch, obtained after the previous step. Next step is to tessellate the *trimmed-patch* and merge the resulting *mesh with* the main mesh model along the original boundaries of the holes and islands. The main challenge here is to ensure conformity of the mesh corresponding to the trimmed

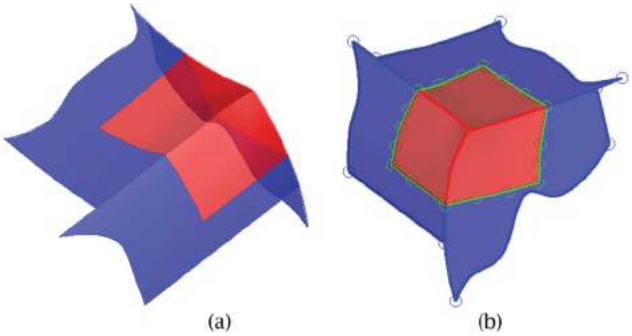

Figure 14. (a) Split composite surface by hole-boundaries (b) split faces and projected edges in green (another view of split faces in 'a').

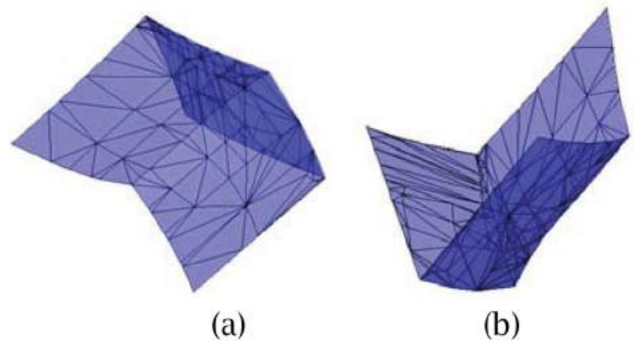

Figure 16. (a) Mesh patch from trimmed composite surface, (b) another view of mesh patch in (a).

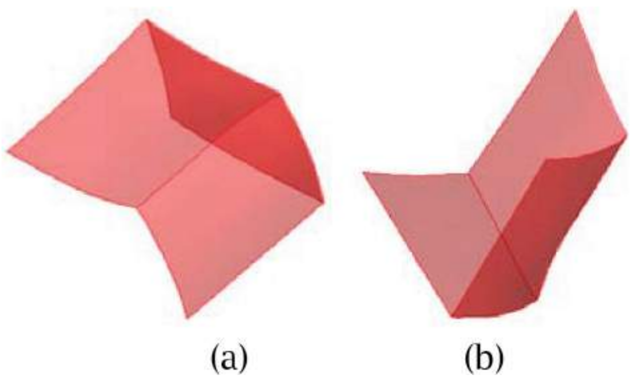

Figure 15. (a) Composite surface after trimming, (b) another view of the trimmed surface in (a).

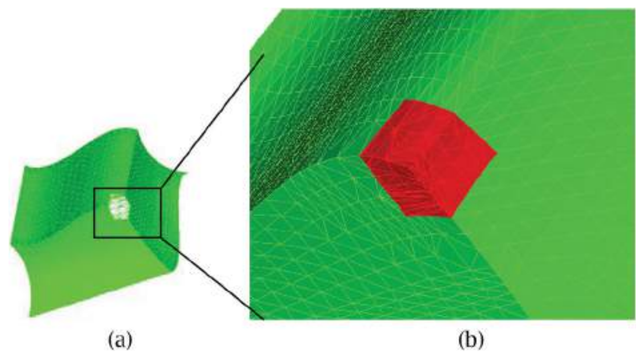

Figure 17. (a) Mesh with multi-segment corner hole, (b) corner hole-filled by red mesh patch.

composite surface with the original boundary edges of the holes and islands in the input mesh model.

As vertices on the boundary are known, any constrained triangulation scheme [35] can ensure conformity of the tessellation of the trimmed composite surface with the original mesh model. In this work, the tessellation function of ACIS [1] is used to both mesh the composite surface patch and merge this mesh with the input mesh model along the conformal edges. Figure 16 shows mesh corresponding to the trimmed-composite surface shown in Figure 15. Final output is shown in Figure 17b. Figure 2i and 2j also represents this step.

5. Results and discussion

The proposed algorithms have been implemented using VC++ in Visual Studio 2008 environment in a laptop with following configuration: Intel core i3 processor, 3GB RAM 64 bit Windows 7 operating system.

APIs of 3D ACIS Kernel modeler [1] have been used for some of the geometric computations. Schema DL and GL viewer of ACIS have been used for visualization. The effectiveness and efficiency of the algorithm have been verified by testing various mesh models available as .stl files obtained from CAD models. Results of testing on different mesh models are presented in this section. Figure 17a shows a corner hole shared by three regions. A mesh-patch in red, shown in Figure 17b, is created to fill the hole.

Figure 18a shows a hole in a helical surface with a square cross section. The hole is spread over the two regions. Mesh-patch filling the hole is shown in red in Figure 18b. Figure 19a shows a hole that contains an island shown in blue. Mesh-patch that fills the hole is shown in red in Figure 19b. A hole in a gear tooth mesh model is shown in Figure 20a. Red mesh-patch that fills this is shown in Figure 20b.

In order to verify the benefit of first constructing a smooth surface capturing the underlying surface in the region of the hole, experiments were done with synthetic data. For a few parts, holes were created in the mesh by removing some triangles from the original mesh model. Mesh-patch to repair the hole so created is obtained by two methods - the approach proposed in this paper and direct tessellation of the discrete points generated from slice sections. The distance of the vertices in the

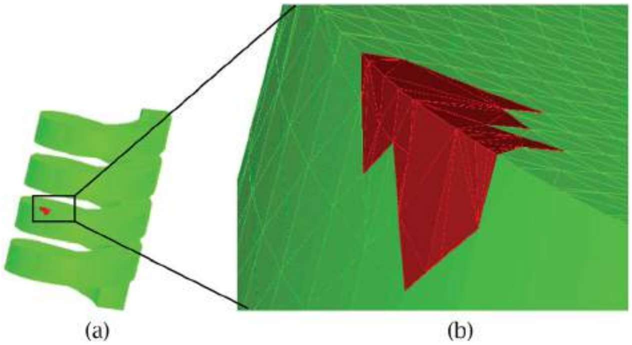

Figure 18. (a) Input mesh with hole, (b) hole-filled by red mesh patch.

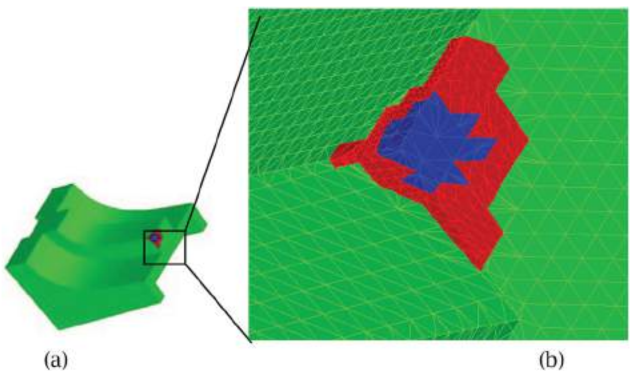

Figure 19. (a) Input mesh with hole and island in blue, (b) hole-filled by red mesh-patch.

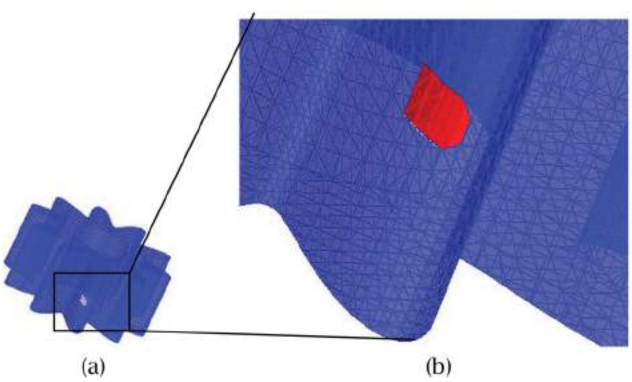

Figure 20. (a) Input mesh with hole, (b) hole-filled by red mesh-patch.

two mesh-patches respectively from the original mesh (a measure of the error in repair) are then compared.

Results, as shown in Table 2, demonstrate that the quality of filling a hole by a curvature continuous smooth surface patch followed by tessellation is superior over direct tessellation of discrete points generated from the slice sections. Average error for smooth surface approach is found to be less than that for direct tessellation in all the cases. The results also show that the difference in error between the two approaches are much larger when the holes are in regions of high curvature (for the meshes in Figures 17 and 20). It is seen that the error for the proposed approach is also high in an absolute sense in these two cases. It is believed that generating more section curves in such cases (missing corners, high curvature regions) by using a smaller offset during slicing would reduce the error and improve the quality of the repaired mesh.

This automatic slice based geometric hole-filling approach will work for all kinds of holes including islands in a hole. Algorithm does not depend on density of triangles in the input mesh model, though some of the existing algorithms [27, 40] depend on the coarseness of the mesh model. The algorithm identifies and generates missing features such as missing curves and missing corners associated with any hole. The quality of repair is good as the algorithm maintains curvature continuity of the new patch with the rest of the mesh. Another advantage of the proposed approach is that the vertices and edges in the input mesh are preserved.

The proposed hole-filling algorithm is also tested for a single segment hole in an organic model (mesh obtained

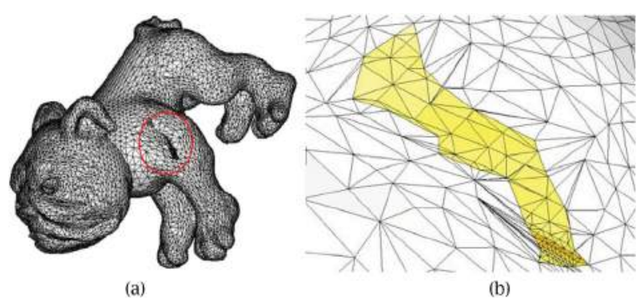

Figure 21. (a) Input mesh with hole, (b) enlarged view of encircled hole after hole-filling.

Table 2. Experimental results.

Examples	Number of input Triangles	Number of new Triangles	Error*E-03 : Average distance of new vertices of mesh-patch from original mesh model without hole	Error*E-03 : Average distance of vertices of direct tessellated mesh from original mesh model without hole
Figure 17	4372	220	44.017	49.675
Figure 18	47090	52	3.017	5.213
Figure 19	25667	87	0.063	0.094
Figure 20	16180	48	30.407	82.636

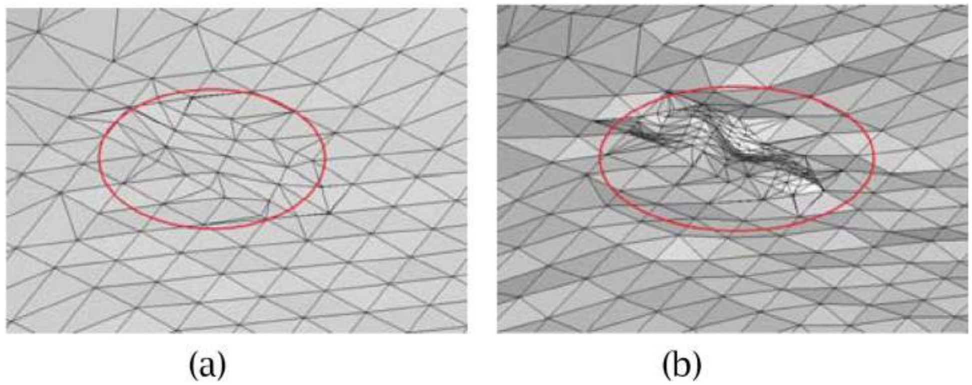

Figure 22. Non-smooth mesh-patch generated to fill the holes are encircled, for noise (a) 5% and (b) 25% respectively.

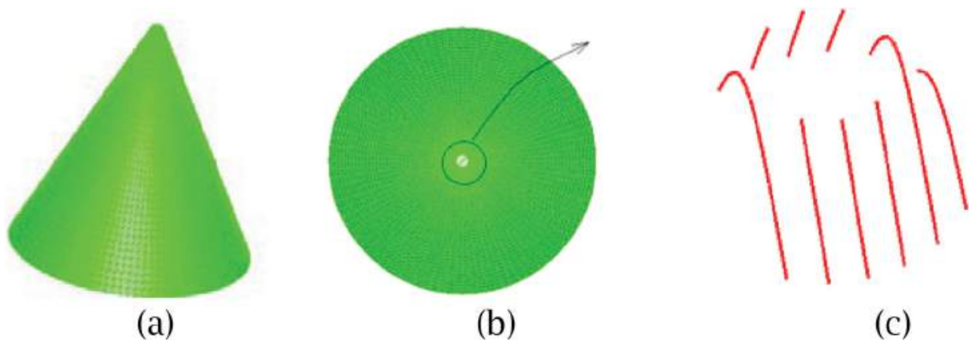

Figure 23. (a) Cone with missing apex, (b) top view of (a), (c) slices around apex.

from scanned data) shown in the figure 21a. This hole does not have any missing features. Enlarged view of the hole (encircled in figure 21 (a)), after hole-filling is shown in Figure-21b. This result demonstrates that the proposed algorithm is able to fill holes even in scanned models.

Since mesh models are imported data from various sources, noise and geometric errors may be present [l]. The algorithm has been tested with noisy input data (obtained by randomly altering the vertices in a mesh obtained from a CAD model). ***Noise*** is defined as percentage deviation of each mesh vertex with respect to a reference dimension. For this experiment, reference dimension is set to 1% of the length of the diagonal of the bounding box of the mesh. It was found through experiments on a sample part, with a diagonal of the bounding box of the mesh of length 77 units, that the procedure has no difficulty in repairing the holes correctly for noise even beyond 25% of the reference dimension, but the smoothness of the hole-filled mesh-patch starts deteriorating after 5% of the noise. Figure 22 shows the generated mesh-patch (encircled portion) for noises 5% and 25% respectively for the sample model.

There could be some degenerate cases where a hole may include the singularity of the underlying surfaces. One such case is shown in Figure 23, where the hole in the mesh is in the apex region of a conical surface. The slices

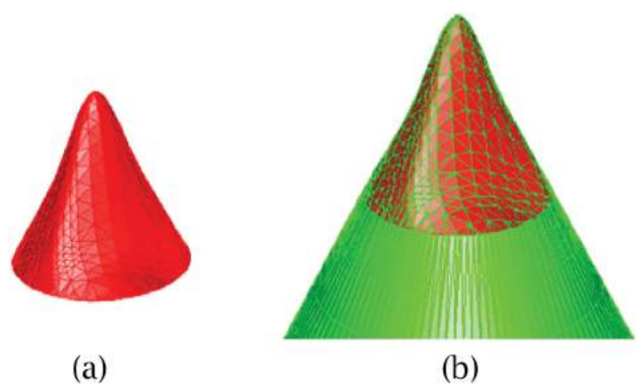

Figure 24. (a) Recovered apex by this algorithm, (b) final output.

generated by the proposed approach to trace the section curves are shown in figure 23c. It can be seen that the tracing of new section curves from both ends of a slice will yield a better result as shown in Figure 24 against the process of tracing only from one end.

Algorithm is developed by assuming that there may not be more than one missing corner per hole. As we should restrict hole-filling by surface patches only for smaller holes [5], one missing corner per hole is a reasonable assumption. Lofting curves are generated by maintaining curvature continuity. Curvature continuity along cross direction is maintained automatically while generating the lofted surface by using lofted curves. While merging input mesh along the conformal boundaries of the input mesh, ACIS [1] merging function is used for this constrained merging operation. Constrained merging (merging between known edges) is much more robust compared to general merging. Since the hole is fit with a smooth surface first that is then tessellated to obtain the mesh, there is no possibility of self-intersecting facets in the resulting mesh.

6. Conclusions

An automatic and feature preserving hole-filling algo rithm has been presented in this paper. This algorithm is automatic and is able to fill the hole even when the hole results in the removal of edges and corners in the object. Holes that run across multiple regions (faces) in the object can therefore be handled. It has been shown that using a smooth surface that maintains C^2 continuity with the mesh at the hole boundary results in a more accurate mesh model. Algorithm is developed by assuming that there may not be more than one missing corner per hole. As we should restrict hole-filling by surface patches only for smaller holes [5], one missing corner per hole is a reasonable assumption. The immediate future work is to improve the overall quality of the lofted surface patch so that the filling error can be brought down further.

ORCID

Nepal Adhikary http://orcid.org/0000-0002-7927-2927
B. Gurumoorthy http://orcid.org/0000-0001-9857-9011

References

[1] ACIS 3D Kernel, http://www.spatial.com, 3D Kernel modeler.
[2] Adhikary, N.; Gurumoorthy, B.: A sliced based approach to recognize and extract free-form volumetric features in a CAD mesh model, Computer-Aided Design and Applications, 13(5), 2016,587-599
[3] Altantsetseg, E.; Khorloo, O.; Matsuyama, K.; and Konno, K.: Complex hole-filling algorithm for 3D models, In Proceedings of CGI '17, Yokohama, Japan, June 27-30, 2017, 6pages. https://doi.org/1145/3095140.3095150
[4] Attene, M.: Direct repair of self-intersecting meshes, Graphical Models, 76(6), 2014, 658–668. https://doi.org/10.1016/j.gmod.2014.09.002
[5] Attene, M.; Campen, M.; Kobbelt, L.: Polygon mesh repairing: An application perspective, ACM Comput. Surv., 45(15), 2013.
[6] Barequet, G.; Sharir, M.: Filling gaps in the boundary of a polyhedron, Computer Aided Geometric Design, 12(2), 1995, 207–229. https://doi.org/10.1016/0167-8396(94)00011-G
[7] Bendels, G. H.; Schnabel, R.; Klein, R.: Detail-preserving surface in painting, The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, 2005, 41-49.
[8] Bischoff, S.; Pavic, D.; Kobbelt, L.: Automatic restoration of polygon models, ACM Trans on Graphics, 24(4), 2005, 1332–1352. https://doi.org/10.1145/1095878.1095883
[9] Botsch, M.; Kobbelt, L.; Pauly, M.; Alliez, P.; Levy, B.: Polygon Mesh Processing, Massachusetts: A K Peters; 2010.
[10] Centin, M.; Pezzotti, N.; Signoroni, A.: Poisson-driven seamless completion of triangular meshes, Computer Aided Geometric Design. Volumes 35–36, May 2015, Pages 42-55.
[11] Chonga, C. S.; Kumar, A. S.; Lee, H. P.: Automatic mesh-healing technique for model repair and finite element model generation, Finite Elements in Analysis and Design, 43(15), 2007, 1109–1119. https://doi.org/10.1016/j.finel.2007.06.009
[12] Curless, B.; Levoy, M.: A volumetric method for building complex models from range images, In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques, 1996, 303-312. https://doi.org/10.1145/237170.237269
[13] Dahlquist, G.; Bjorck, A.: Numerical Methods. Englewood Cliffs, NJ, Prentice-Hall, 1974.
[14] Davis, J.; Marschner, S. R.; Garr, M.; Levoy, M.: Filling holes in complex surfaces using volumetric diffusion, In: Processing of 3D Data Processing Visualization and Transmission, Italy, June 2002, IEEE Computer Society Press, New York, 428-38. https://doi.org/10.1109/tdpvt.2002.1024098.
[15] Fortes, M.A.; Gonzalez, P.; Palomares, A.; Pasadas, M.: Filling holes with geometric and volumetric constraints, Computers & Mathematics with Applications, 74(4), 15 August 2017, Pages 671-683.
[16] Gao, S.; Zhao, W.; Lin, H.; Yang, F.; Chen, X.: Feature suppression based CAD mesh model simplification, Computer-Aided Design, 42(12), 2010, 1178-1188. https://doi.org/10.1016/j.cad.2010.05.010
[17] Gregory, J. A.; Zhou, J.: Filling polygonal holes with bicubic patches, Computer Aided Geometric Design, 11(4), 1994, 391–410. https://doi.org/10.1016/0167-8396(94)90205-4
[18] Harary, G.; Tal, A.: Context-based coherent surface completion, ACM Transactions on Graphics, 33(1), 2014. https://doi.org/10.1145/2532548
[19] Harary, G.; Tal, A.; Grinspun, E.: Feature-preserving surface completion using four points, Eurographics Symposium on Geometry Processing, 33(5), 2014, 45-54. https://doi.org/10.1111/cgf.12430
[20] Hétroy, F.; Rey, S.; Andújar, C.; Brunet, P.; Vinacua, A.: Mesh repair with user-friendly topology control, Computer-Aided Design, 43(1), 2011, 101–113. https://doi.org/10.1016/j.cad.2010.09.012
[21] Jacobson, A.; Kavan, L.; Sorkine-Hornung, O.: Robust inside-outside segmentation using generalized winding numbers, ACM Trans. Graph. 32(4), 2013, https://doi.org/10.1145/2461912.2461916
[22] Joshua, P.; Szymon, R.: Atomic volumes for mesh completion, In: Proceedings Eurographics Symposium on Geometry Processing, Dublin, Ireland, 2005, 33-41, Blackwell Press.

[23] Ju, T.: Fixing geometric errors on polygonal models: a survey, J Computer Sci Technology, 24(1), 2009,19-29, https://doi.org/10.1007/s11390-009-9206-7
[24] Ju, T.: Robust repair of polygonal models, ACM Transactions on Graphics, 23(3), 2004, 888–895, https://doi.org/10.1145/1015706.1015815
[25] Jun, Y.: A piecewise hole filling algorithm in reverse engineering, Computer-Aided Design 37(2), 2005, 263–270. https://doi.org/10.1016/j.cad.2004.06.012
[26] KUMAR, A.; SHIH, A. M.; ITO, Y.; ROSS, D. H.; SONI, B. K.: A hole-filling algorithm using non-uniform rational b-splines, In Proceedings of the 16th International Meshing Roundtable 2007, 169–182.
[27] Li, Z.; Meek, D. S.; Walton, D. J.: Polynomial blending in a mesh hole-filling application, Computer-Aided Design, 42(4), 2010, 340–349, https://doi.org/10.1016/j.cad.2009.12.006
[28] Liepa, P.: Filling Holes in Meshes, In: SGP '03: proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on geometry processing; 2003, 200-205.
[29] Nooruddin, F.S.; Turk, G.: Simplification and repair of polygonal models using volumetric techniques, IEEE Trans Vis Computer Graph 9(2), 2003, 191-205, https://doi.org/10.1109/TVCG.2003.1196006
[30] Pauly, M.; Mitra, N.; Giesen, J.; Gross, M.; Guibas, L.J.: Example-Based 3D scan completion. In: Proceedings of Eurographics symposium on geometry processing, 2005, 23-32.
[31] Perez, E.; Salamanca, S.; Merchan, P.; Adan, A.: A comparison of Hole-filling methods in 3D, Int. J. Appl. Math. Comput. Sci., Vol. 26, No. 4, 2016, pages885–903 (DOI: doi:10.1515/amcs-2016-0063)
[32] Pernot, J-P.; Moraru, G.; Veron, P.: Filling holes in meshes using a mechanical model to simulate thecurvature variation minimization, Computers & Graphics, 30(6), 2006, 892–902. https://doi.org/10.1016/j.cag.2006.08.020
[33] Podolak J.; Rusinkiewicz, S.: Atomic volumes for mesh completion. In Eurographics Symposium on Geometry Processing 2005, 33–42.
[34] Sharf, A.; Alexa, M.; Cohen-Or D.: Context-based surface completion, ACM Transactions on Graphics – Proceedings of ACM SIGGRAPH, 23(3), 2004, 878-887, https://doi.org/10.1145/1186562.1015814
[35] Shewchuk, J. R.: General-Dimensional Constrained Delaunay and Constrained Regular Triangulations, I: Combinatorial Properties, Discrete & Computational Geometry, 39(1), 2008, 580–637, https://doi.org/10.1007/s00454-008-9060-3
[36] Wang, X.; Liu, X.; Lu, L.; Li, B.; Cao, J.; Yin, B.; Shi, X.: Automatic hole-filling of CAD models with feature-preserving. Computers & Graphics 36(2), 2012, 101-110, https://doi.org/10.1016/j.cag.2011.12.007
[37] Wang J.; OLIVEIR, M. M.: A Hole-Filling Strategy for Reconstruction of Smooth Surfaces in Range Images, In: XVI Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI, 2003, https://doi.org/10.1109/SIBGRA.2003.1240986
[38] Wang, J.; Oliveira M. M.: Filling holes on locally smooth surfaces reconstructed from point clouds. Image and Vision Computing, 25(1), 2007, 103–13. https://doi.org/10.1016/j.imavis.2005.12.006
[39] Wang, L-C.; Hung, Y-C.: Hole filling of triangular mesh segments using systematic grey prediction, Computer-Aided Design, 44(12), 2012, 1182–1189, https://doi.org/10.1016/j.cad.2012.07.007
[40] Zhao, W.; Gao, S.; Lin, H.: A robust hole-filling algorithm for triangular mesh, The Visual Computer, 23(12), 2007, 987–997, https://doi.org/10.1007/s00371-007-0167-y