Triangulation of Implicitly Defined Mid-Surfaces

Jules Bloomenthal

Unchained Geometry, Seattle, WA 98112, USA
bloomenthal@unchainedgeometry.com

Abstract. A method to triangulate the medial axis [1] is modified. In the new method the coping surfaces of a thin-wall object are used for insidedness but ignored for footpoints, yielding the mid-surface, not the medial axis. Three distinct contour-followers are employed. After a brief, manual tagging of the coping surfaces, the method is automatic. It computes a triangulation to arbitrary precision and appears robust for complex, thin-wall objects.

Keywords: triangulation, mid-surface.

1 Introduction

The algorithm presented in [1] to compute the medial axis of a solid object may be modified to compute the mid-surface of a thin-wall object. In the original method, the medial axis is implicitly defined as the locus of centers of inscribed spheres that touch the generating object at two or more locations, or *footpoints*. The unit-length vector from an arbitrary 3D point to its footpoint is the point's *footpoint vector*.

The implementation defines a lattice of cells that collectively surrounds the object. For each cell edge, the footpoint vectors at the endpoints are compared: if they diverge significantly, the edge is presumed to transect the medial axis, as shown below. With binary sectioning, the surface is found to arbitrary precision.

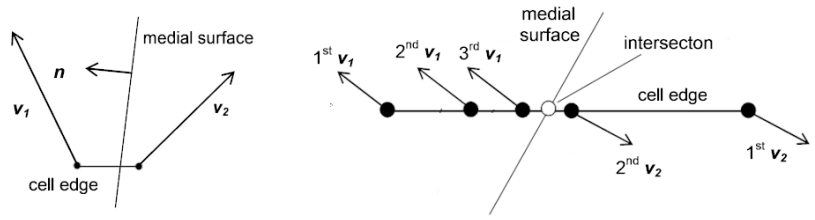

Fig. 1 Cell edge crossing the medial axis (left) and binary sectioning of edge (right)

J. Sarrate & M. Staten (eds.), *Proceedings of the 22nd International Meshing Roundtable*,
DOI: 10.1007/978-3-319-02335-9_15, © Springer International Publishing Switzerland 2013

The footpoint serves also to determine object insidedness: if the footpoint vector $\boldsymbol{v}$ of a given point and the (outward-facing) surface normal $\boldsymbol{n}$ at the footpoint are in the same hemisphere (that is, $\boldsymbol{n} \bullet \boldsymbol{v} > 0$), then the point is interior to the object. Employing this test, the medial axis is confined to the object interior.

For a thin-wall object, if the *coping surfaces* are utilized for insidedness but ignored for footpoints, then the algorithm yields the mid-surface, not the medial axis. A coping surface is the 'covering course of a wall' (from Merriam-Webster); in the figure below, right, the coping surfaces are the left and right edges of the cross-section.

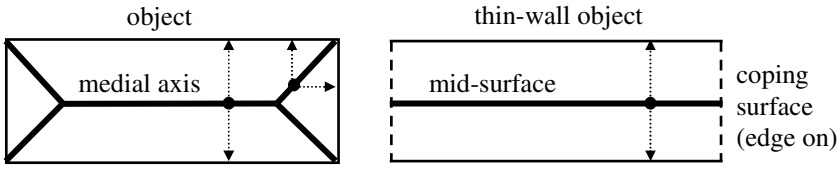

Fig. 2 Medial axis and mid-surface in cross-section; vectors are to non-coping footpoints

In the figure below, left, the coping surfaces are yellow, and the non-coping surfaces green. Below, right, a piece of the mid-surface is shown along with lines between mid-surface vertices and non-coping footpoints.

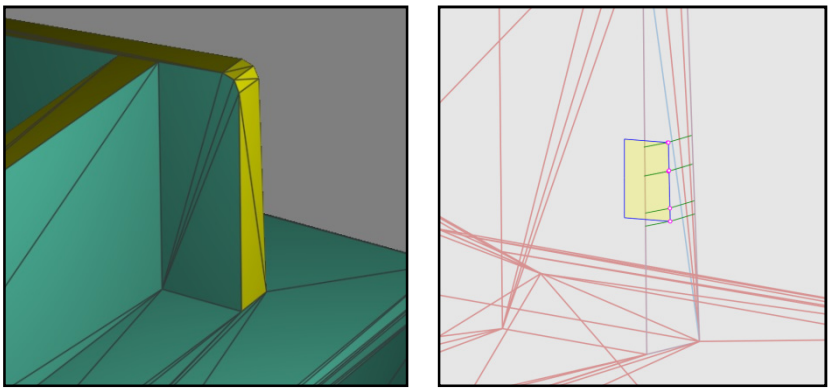

Fig. 3 Coping surfaces (left) and footpoints on object (right)

This use of coping surfaces is the principal novelty of this paper. The paper also introduces, in section 4, an improved method to connect mid-surface vertices.

2 Previous Work

The use of the mid-surface for design and analysis is described in [2] and [3]; methods for its computation are described in [2] and [5-8].

In [2] offsets to a defining 'outer' surface are used to compute an 'inner' and 'mid' surface. In [5] the mid-surface is computed based on a modified Delaunay triangulation, given a known pairing of surface elements. In [6] the mid-curve of a coping surface is used to determine loops in adjacent surfaces so that surface pairs that sandwich pieces of the mid-surface can be identified. In [7] and [8] surface pairs are identified after complicating features of the object are removed.

These methods all depend to some extent on a high-level description of the object to guide the selection of surface pairs. For non-trivial objects, some manual pairing may be required. Several commercial software packages offer mid-surface generation if surfaces are presented pair-wise by manual input or *a priori* specification.

Test objects demonstrating these methods are relatively simple and not all object topologies are accommodated. The mid-surface results of individual pairings must be stitched together, which can be problematic.

In contrast, the method presented in this paper is automatic except for a brief, manual tagging of the coping surfaces of the object.[1] The method only requires the ability to compute the nearest point on a thin-wall object; no feature description or high level abstraction is considered.

3 Overview

The input to the method is the boundary representation of a 3D, thin-wall object. The present implementation accepts a triangulated mesh, but any surface representation suffices provided it supports an operator to compute the footpoint of an arbitrary 3D point.[2]

The method in [1] generates the medial axis piecewise within semi-adjacent tetrahedra. Any set of tetrahedra that collectively enclose the object can be used (this precludes a Delaunay triangulation, which consists of points on but not outside the object).

The present implementation uses tetrahedra whose vertices lie on a cubic lattice; the lattice points are readily identified by an integer triplet and the relationship between tetrahedra (*i.e.*, the shared edges and faces) is fixed.

Within each transected tetrahedron, a piecewise solution for the mid-surface is generated.

---

[1] The test object (fig. 13) contains several hundred coping triangles; these were manually tagged in 15 minutes.

[2] For a triangulated mesh, the footpoint is the nearest of the proximate points, one for each triangle. The number of triangles to be tested can be substantially reduced with an octree. The test object (fig. 13) is a mesh of 2600 triangles; with an octree depth of 5, the number of triangles per octree node ranges from 3 to nearly 200; most tests involve nodes with fewer than 35 triangles.

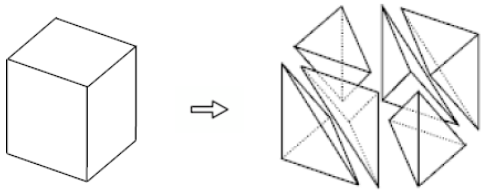

Fig. 4 Six tetrahedra within a lattice cell

For manifold sections, the intersections of tetrahedral edges with the mid-surface are computed and connected to form a single triangle or quadrilateral per transected tetrahedron, as shown below.

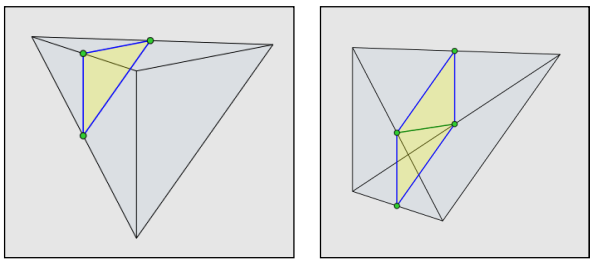

Fig. 5 Surfaces intersecting tetrahedron as triangle (left) and quadrilateral (right)

The medial axis and the mid-surface are, however, non-manifold; they contain orientable component surfaces attached to other components along a non-manifold seam. These components are internally defined by manifold edges and are circumscribed by non-manifold and boundary edges, as shown below.

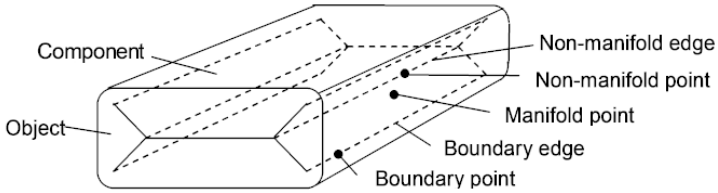

Fig. 6 Non-manifold seams and boundary edges

Therefore, in addition to manifold vertices located on tetrahedral edges, non-manifold and boundary vertices may be located on tetrahedral faces. And, because a non-manifold seam may be terminated arbitrarily by a boundary edge, non-manifold vertices also occur internal to a tetrahedron.

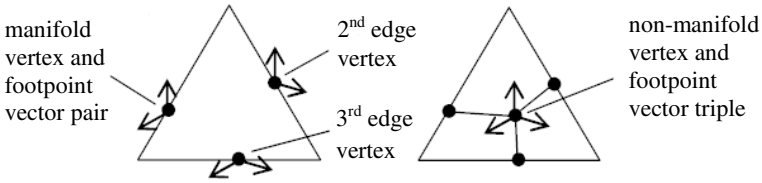

Fig. 7 Manifold edge vertices, left, and non-manifold face vertex, right

In the following figure, the white point is internal to the tetrahedron and is at the top of the non-manifold seam, which is shown in red. The bottom of the seam rests on the lower face of the tetrahedron. The upper boundary edges, in blue, are the border of the three component surfaces (shown in yellow tint, radiating about the red non-manifold seam), trimmed by a coping surface.

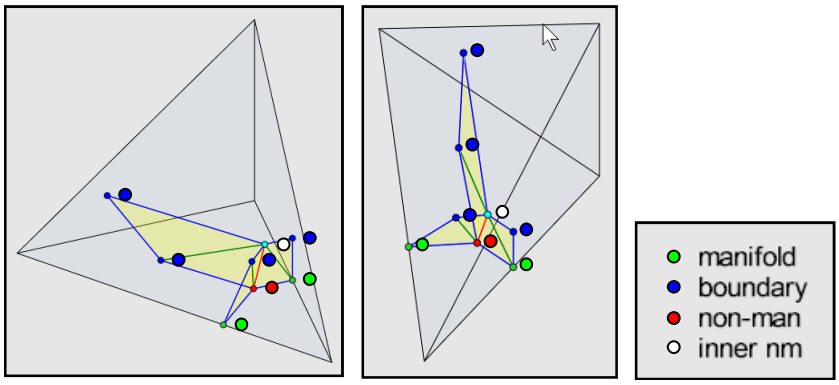

Fig. 8 Non-manifold seam and boundary edges within a tetrahedron (two views)

This example suggests an algorithm that consists of the following steps:

- determination of ‘edge vertices’ by binary-subdivision along transected tetrahedral edges,
- determination of ‘face vertices’ by 2D contour-following within a tetrahedral face beginning at an edge vertex, and
- determination of ‘inner vertices’ by 3D contour-following along a non-manifold seam or boundary edge within the tetrahedron.

The resulting triangulation produces manifold edges shared by two triangles, non-manifold edges shared by three or more triangles, and boundary edges shared by a single triangle.

4 Connectivity

After all vertex locations are computed for a tetrahedron, their connectivity is established. With manifold surfaces, this can be performed using a brief table of edge configurations (*i.e.*, those three or four tetrahedral edges transecting the mid-surface).

With non-manifold and boundary vertices present, however, connectivity is more difficult to establish. In [1] surface normals at vertices are matched to establish a vertex-to-vertex pairing.

The surface normal can be calculated from footpoint vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ as:

$$\boldsymbol{N} = (\boldsymbol{v}_1 + \mathbf{v}_2) \times (\mathbf{v}_1 \times \mathbf{v}_2), \tag{1}$$

for $(\boldsymbol{v}_1 \bullet \boldsymbol{v}_2) / (|\boldsymbol{v}_1||\boldsymbol{v}_2|) > -1$ (otherwise v_1 is opposite $\boldsymbol{v}_2$ so that either can serve as $\boldsymbol{N}$). The unit length normal is given by $\boldsymbol{N}/|\boldsymbol{N}|$.

Reliance solely on surface normals, however, can produce connectivity errors if a contour has significant curvature within a tetrahedron. The present method establishes connectivity through contour-following, except in the few manifold cases (fig. 5) where connectivity is obvious.

Three distinct contour-followers are employed. One is two-dimensional within the face of a tetrahedron; two are three-dimensional and interior to the tetrahedron:

- from each *edge vertex* a 2D contour is followed to another edge vertex or to a *face vertex*,
- from each face vertex a 3D contour is followed along a *non-manifold seam* to another face vertex or to an *inner vertex*, and
- as above but the contour follows a *boundary edge* (not a non-manifold seam).

4.1 2D Follower

The 2D contour follower creates a series of transecting segments along the face of a tetrahedron, until a change in the footpoint vectors is detected. For example, in fig. 9, three contours on the tetrahedron's right face intersect at the red vertex, *s*. The short pink segments crossing the contour connect samples from which green footpoint vectors are drawn. Beyond the intersection but in line with the contours are grey segments whose footpoint vectors do not align with those tracking the contour, indicating a branch point.

The location of the branch point can be computed to arbitrary precision, and serves as the center for a circle of footpoint vector tests that guarantee all contours at the branch point have been followed.[3]

[3] Although most 2D contours proceed from a tetrahedral edge to a branch point on a tetrahedral face, some contours must proceed from the face. For example, in fig. 9 vertices *1* and *2* are on tetrahedral edges; contours are followed from them to the branch point. But vertex *3* is not on an edge and is detected by following the contour from the transecting segment marked *s*.

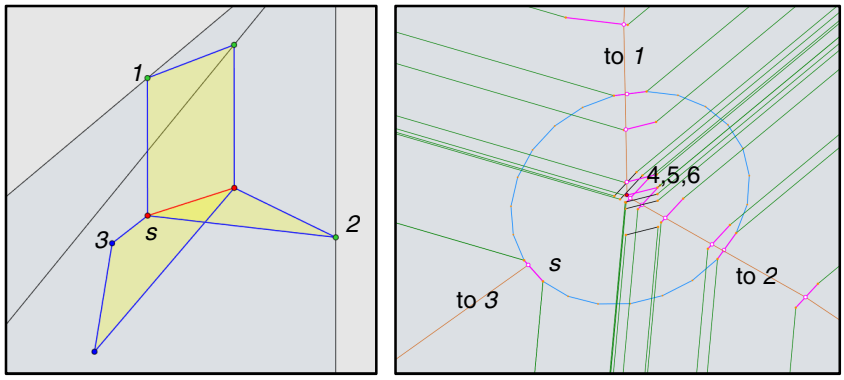

Fig. 9 2D contours from edge vertices meet at a face vertex; three distinct footpoint directions imply three contours

4.2 3D Non-manifold Follower

The three-dimensional non-manifold seam is followed using a series of transverse circles perpendicular to the contour; each circle is expected to intersect the mid-surface in three or more locations, which leads to a straightforward solution for the intersection of the three or more sub-surfaces that impinge on the seam.

For example, in Figure 10, left, the follower takes a step along its previous tangent, shown by the green arrow; this new location (the green dot) is the center for footpoint vectors that yield three transecting segments (with the intersection points shown in white). Each of the three dashed magenta lines represents the intersection of the sub-surface with the plane of the circle. The pairwise intersections of these lines are nearly coincident and their average can be used as the next point of the contour.

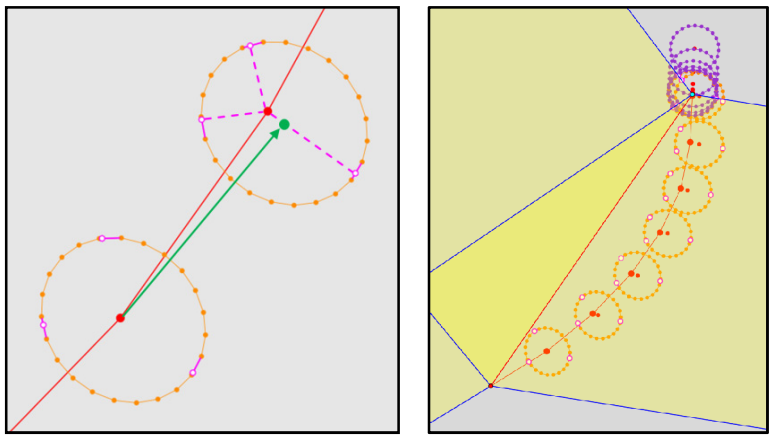

Fig. 10 Contour-following along a non-manifold seam

In Figure 10, bottom, a series of contour steps defines the contour until it reaches its end, in this case a coping element; in the figure, the purple dots are external to the object.

4.3 3D Boundary Follower

The boundary contour requires a more complex mechanism. The transverse circle is expected to intersect the mid-surface in only one location, so that the intersection technique employed for non-manifold seams cannot be applied. Instead, from the one and only intersection, a 2D contour is followed in the plane of the circle, until the boundary edge is located, as shown below.

So that surface branches are not skipped (*e.g.*, hopping over a 'T' intersection), a "sweep" circle is tested for the expected two intersections with the mid-surface.

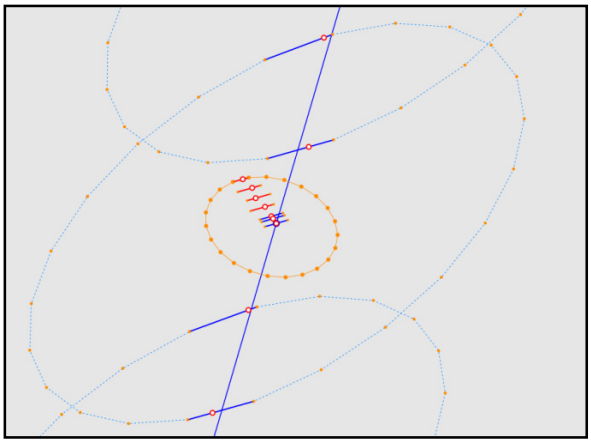

Fig. 11 A contour-following step on a boundary edge; the transverse circle is in orange and the overlapping sweep circles are in light blue

The contour follower is complicated further by the need to accommodate two types of boundary terminations: one caused by a weakening difference between the footpoint vectors, such as the center of a round; and one caused by a coping surface where the footpoint vectors remain divergent but the contour leaves the object interior.

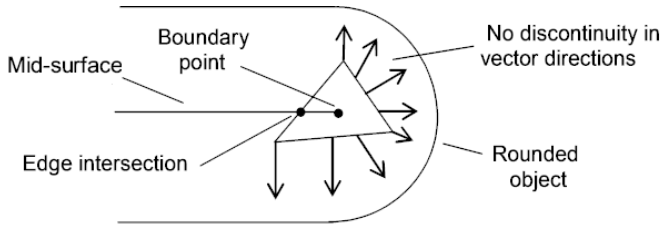

Fig. 12 A tetrahedral face with boundary point due to rounded object

4.4 Vertex Ordering

Each contour, whether from edge vertex to face vertex, or face vertex to inner vertex, inherently specifies one or more pairings between mesh vertices (for example, the non-manifold seam in Figure 9 represents three pairings: 1/4, 2/5, and 3/6). The pairs can be created in any order, but once all contours have been defined for a tetrahedron, the pairs are sequenced so that one pair connects to another (not unlike dominos end to end) until a polygon of three or more sides is formed. The polygon, if 4 or more sides, is decomposed into triangles.

5 Results

The mid-surface below was generated from approximately 8500 cubic cells.

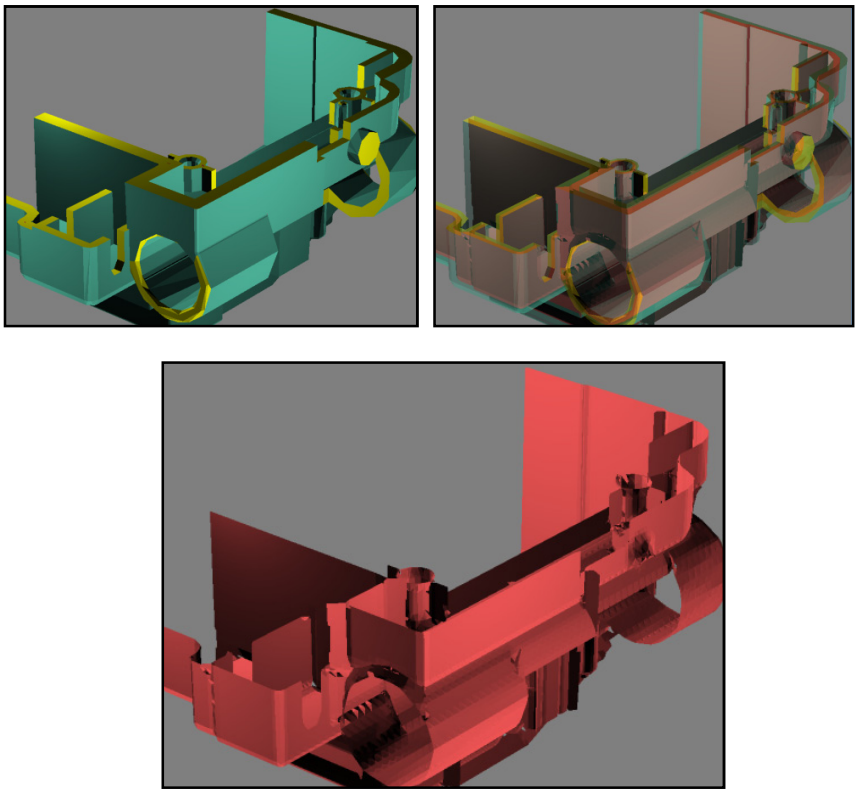

Fig. 13 Upper left: original thin-wall object with yellow coping elements, upper right: mid-surface within semi-transparent object, and bottom: mid-surface

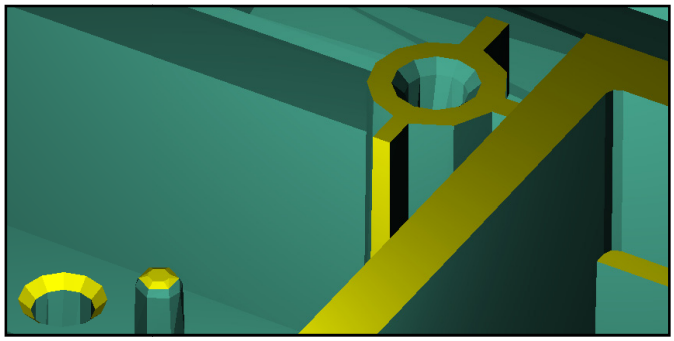

Fig. 14A Object detail

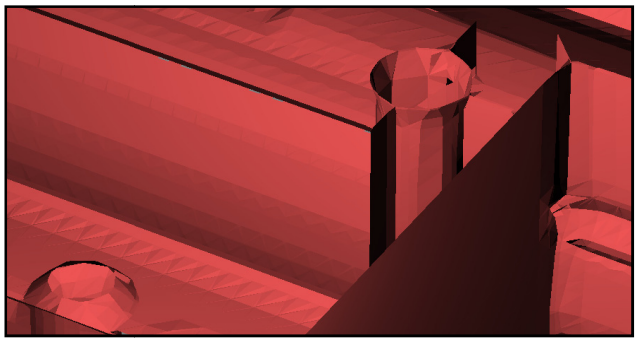

Fig. 14B Mid-surface details

6 Post-processing

There is, necessarily, an overlap between the minimum angle between footpoint vectors needed to ensure connection of component surfaces and the higher threshold needed to prevent unwanted branching. This may produce unwanted "fins" as in the figure below. A brief, manual pruning be necessary.

Fig. 15 Object, top, and unwanted fins, bottom (one fin points up and to the left, the other points down and to the right)

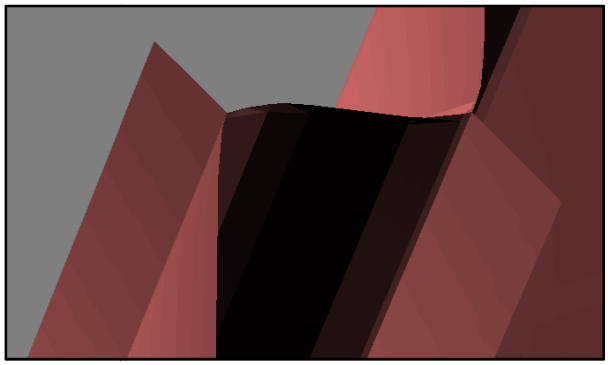

Fig. 15 (*continued*)

7 Future Work

Several areas of future work are promising.

7.1 Parallelization

The solutions within a lattice cell are local and thus amenable to parallelization, as are a number of specific operations. If the software were ported to execute on multiple processors, such as afforded by a modern GPU, the execution time (presently about one hour for fig. 13) could be significantly reduced.

7.2 Mesh Reduction

The intersection of the mid-surface with lattice tetrahedra yields thin and small triangles. A technique introduced in [4] processes the lattice to collapse small or thin triangles, reducing triangle count and improving triangle shape. Its use with non-manifold triangulations may be problematic, however.

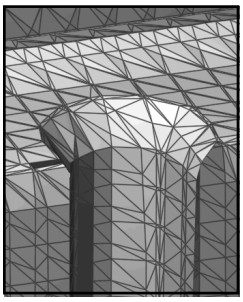

Fig. 16 Mid-surface detail showing thin and tiny triangles

7.3 Coping Detection

The automatic tagging of coping elements should be possible, although a solution is not necessarily straight-forward.

8 Conclusion

This paper has presented a method to generate a triangulated mid-surface mesh from an arbitrary thin-wall object. Aside from minor preparation and post-process pruning, the method is automatic. Results are computed with arbitrary precision and appear robust for complex objects.

Techniques developed in this paper are protected by US patent 6956565 and others pending.

References

1. Bloomenthal, J., Lim, C.: Skeletal Methods of Shape Manipulation. In: Shape Modeling International, Japan (March 1999)
2. Fischer, A., Wang, K.K.: A Method for Extracting and Thickening a Mid-Surface of a 3D Thin Object Represented in NURBS. J. Manuf. Sci. Eng. 119(4B), 706–712 (1997)
3. Lockett, H., Guenov, M.: Graph-Based Feature Recognition for Injection Molding based on a Mid-Surface Approach. Computer-Aided Design 37(2), 251–262 (2005)
4. Moore, D., Warren, J.: Compact Isocontours from Sampled Data. In: Kirk, D. (ed.) Graphics Gems III. Academic Press, New York (1992)
5. Quadros, W., Shimada, K.: Hex-Layer: Layered All-Hex Mesh Generation on Thin Section Solids via Chordal Surface Transformation. In: 11th International Meshing Roundtable, Ithaca (September 2002)
6. Ramanathan, M., Gurumoorthy, B.: Generating the Mid-Surface of a Solid using 2D MAT of its Faces. Computer-Aided Design and Applications 1, 665–674 (2004)
7. Sheen, D.P., Son, T., Ryu, C., Lee, S., Lee, K.: Dimension Reduction of Solid Models by Mid-Surface Generation. International Journal of CAD/CAM 7(1) (2007)
8. Sheen, D.P., Son, T., Myung, D.K., Ryu, C., Lee, S.H., Lee, K., Yeo, T.J.: Transformation of a Thin-Walled Solid Model into a Surface Model via Solid Deflation. Computer-Aided Design 42, 720–730 (2010)