Thickening Freeform Surfaces for Solid Fabrication

Charlie C.L. Wang[1*] Yong Chen[2]

[1]Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, China

[2]Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, USA

Abstract

Purpose: Given an intersection-free mesh surface S, we introduce a method to thicken S into a solid H located at one side of S. By such a surface-to-solid conversion operation, industrial users are able to fabricate a designed (or reconstructed) surface by rapid prototyping.

Design/methodology/approach: In this paper, we first investigate an implicit representation of the thickened solid H according to an extension of signed distance function. After that, a partial surface reconstruction algorithm is proposed to generate the boundary surface ∂H of H, which remains the given surface S on the resultant surface.

Finding: Experimental tests show that the thickening results generated by our method give nearly uniform thickness and meanwhile do not present shape approximation error at the region of input surface S. These two good properties are important to the industrial applications of solid fabrication.

Research limitation/implications: The input polygonal model is assumed to be intersection-free, where models containing self-intersection will lead to invalid thickening results.

Originality/value: A novel robust operation to convert a freeform open surface into a solid by introducing no shape approximation error. A new implicit function that gives a compact mathematical representation, which can easily handle the topological change on the thickened solids. A new polygonization algorithm that generates faces for the boundary of thickened solid meanwhile retaining faces on the input open mesh.

***Keywords*:** Thickening, mesh surface patch, signed distance, surface to solid, rapid prototyping.

1 Introduction

Rapid prototyping (RP) is a very important fabrication tool to help designers to realize the prototype of their product – especially for those products with freeform shape. Common rapid prototyping techniques (ref. [1]) include *selective laser sintering* (SLS), *stereolithography* (SLA), *fused deposition modeling* (FDM), *laminated object manufacturing* (LOM), *3D printing* (3DP), etc. In order to fabricate a designed model (or shape) into a real object by these rapid prototyping methods, the model is usually represented as the boundary representation (B-rep) of a solid in $\Re^3$ (ref. [2]).

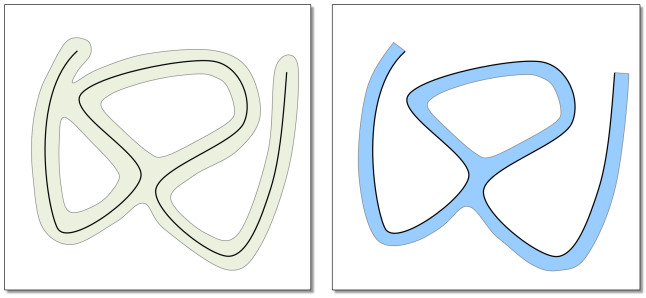

Figure 1: The difference between surface offsetting (left) and thickening (right), where the bolded curve denotes the given freeform surface to be remained on the resultant solid.

In the applications of design and manufacturing, there is a demand for a geometric modeling tool to convert a freeform mesh surface (usually in the form of open surface) into a thin-shell solid that can be fabricated by rapid prototyping machines. The freeform mesh surfaces to be processed could be a surface region trimmed from a designed model. In order to physically check the fitness of the designed model to other objects, a prototype surface of the model needs to be fabricated. Moreover, the surface to be thickened (and then fabricated) could also be a patch reconstructed from the scanned point clouds. Many RP fabrication procedures require the solid model having nearly uniform thickness so that large residual strains will not be generated by the non-uniform shrinkage of materials. The freeform surfaces to be fabricated are usually represented in the form of piecewise-linear mesh surfaces, which are not limited to rectangular or triangular parametric surfaces and could have high genus topology. Therefore, the offsetting techniques for parametric surfaces in the literature (e.g., [3–5]) cannot be directly used here. Moreover, although the offsetting operation (e.g., [6–8]) in solid modeling can compute offsetting shells for general 3D freeform models, the computed shell are located on both sides of a surface patch S that does not satisfy the requirement described below (see the left of Fig.1 for an illustration).

Offsetting the input mesh surface one-side by explicitly moving vertices may lead to self-intersections. For example, as shown in the right of Fig.1, the intersection-free one-

*Corresponding Author; E-mail: cwang@mae.cuhk.edu.hk

side offsetting result in a surface that has different topology to the input surface. It is also not clear how to define the boundary of this "one-side offsetting". Although it is possible to "project" the boundary of input surface onto the offset surface and cut off undesired regions, the projection and cutting off steps involve many numerical predicates and will suffer the robustness problems [9]. In this paper, we define a compact representation for thickened solids, which help on the robust and efficient computation of resultant models in boundary representation.

The mesh surface trimmed from an existing model is expected to be not changed during the thickening. Specifically, the input mesh surface is required to be remained as part of the boundary surface of the resultant solid (i.e., the output of thickening operation), where most of the RP machines use triangular mesh surfaces as the standard representation of input models. The given surface S is remained on the boundary of the resultant solid so that S can be fabricated. Figure 2 shows such an example on fabricating a surface for helmet design. On a head model reconstructed from a scanned point cloud, designers can draw freeform curves (see Fig.2(a) and (b)) and the mesh surface is then trimmed off by these curves into a new freeform surface patch to be fabricated (see Fig.2(c)). After applying our mesh thickening operation, the surface patch is converted into a thin-shell solid (see Fig.2(d)) and fabricated into a real object as shown in Fig.2(e). The physical fitness check can be taken by this prototype and the head of mannequin.

Problem Definition Given a two-manifold mesh surface patch S, a thickening operation generates a triangular mesh that represents the boundary surface ∂H of a solid H, which is located at one side of S and has a user specified thickness r. Meanwhile, $\forall \mathbf{p} \in S$ ($\mathbf{p} \in \Re^3$), the distance between $\mathbf{p}$ and ∂H,

$$dist(\mathbf{p}, \partial H) = \inf_{\forall \mathbf{q} \in \partial H} \|\mathbf{p} - \mathbf{q}\|,$$

must be zero.

To solve this surface thickening problem, we develop a new method in this paper to produce the triangular mesh surface of ∂H in two steps. First, an implicit representation of the thickened solid H according to an extension of signed distance field is defined on an uniform grids with $(2^l+1)\times(2^l+1)\times(2^l+1)$ nodes, where each node stores a binary value to indicate whether the node is inside H (by value '1'). In this step, a hierarchical assigning algorithm is developed to assign the values of grid nodes efficiently. After that, a new partial surface reconstruction algorithm is investigated to generate the surface of ∂H. To satisfy the requirement of $dist(\mathbf{p}, \partial H) = 0$, the given surface patch S is only modified to a new surface patch $\hat{S}$ by splitting the triangles located on the boundary of S if necessary. Other triangles on S are remained, and the reconstructed mesh surfaces for ∂H are stitched to the boundaries of $\hat{S}$.

1.1 Literature review

The related work in literature can be classified into surface offsetting, solid offsetting, and other solid modeling operations with the help of volumetric representations, which are reviewed below.

The thickening operation proposed in this paper closely relates to the offsetting operation, which can be applied to curves, surfaces and solids. In the earlier work of Rossignac and Requicha [6], the mathematical basis for offsetting solids was described. The offset techniques for curves and surfaces have been extensively studied by Pham [11] and Maekawa [12]. For offsetting a 3D surface, the most difficult issue is how to effectively and efficiently remove those self-intersected regions that do not belong to the resultant offset surface. Most of recent surface offsetting approaches (e.g., [3, 4]) focus on solving this issue. However, the input of these approaches are parametric surfaces with rectangular parametric domains (or triangular parametric surface such as [5]), which cannot be directly applied here to the piecewise-linear surface patches. For the applications in CAD/CAM, more and more models are represented by piecewise-linear freeform surface (especially those objects reconstructed from scanned 3D point clouds or scanned volumetric images).

The offsetting operation for 3D surfaces can be extended to compute the offsets of 3D models by first offsetting all surfaces of a model and then trimming (or extending) these offset surfaces to reconstruct a closed 3D model [6, 13, 14]. These earlier approaches first compute a superset of the offset surface by offsetting 1) vertices into spheres, 2) edges into cylinders, and 3) faces into parallel faces. Then, they trim that superset by subdividing its elements at their common intersections and deleting the pieces that are too close to the original solid. This is a very expensive computing process and the trimming at tangential contacted regions is numerically unstable. Although the recent work in solid modeling can remove the self-intersections more robustly and efficiently with the help of other representation (e.g., [7, 8, 15]), simply applying the solid offsetting operator to a mesh surface patch S will generate a solid on both side of S, which does not satisfy the requirement defined in our objective of surface thickening to remain S on the boundary surface of a resultant solid (see Fig.1).

Many offsetting approaches for 3D solids seek the help of volumetric representation of solids to remove the self-intersections on the result of offsetting. Widely used representations in these approaches include voxel-based representation [16, 17], ray-based representation [8, 18, 19], the fast marching method [20, 21], distance-field based representation [22–25], or Binary Space Partition (BSP) tree [26]. Some of them can be applied to solid modeling operations that are more general than offsetting (e.g., Minkowski-sum, general sweeping, etc.). However, to the best of our knowledge, none of these approaches can generate a mesh surface that satisfies the requirement of being coincident with the input surface patch. The proposed mesh thickening approach generates the indication field on uniformly sampled grids with the help of distance-field. Nevertheless, the novelty here is more than using the distance-field to remove self-intersections. Details are discussed in the following subsection.

Lastly, the partial surface reconstruction algorithm is akin

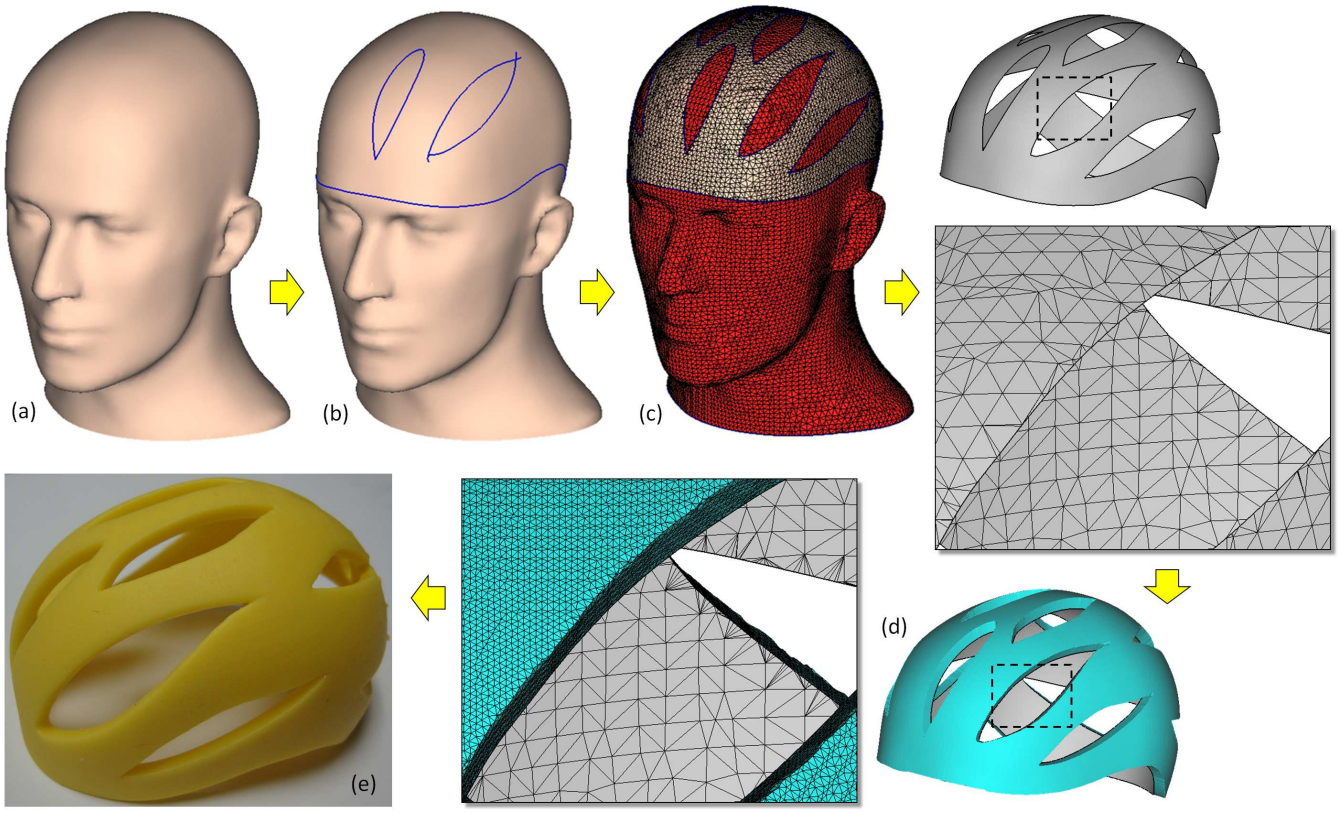

Figure 2: Application of using the thickening operation in the helmet design and prototype fabrication. Given a head model in (a), designers can draw freeform curves on the surface of the head model (see (b)) and cut out the triangular mesh surface patch for modeling the helmet from the head model (see (c)) by using the method in [10]. The thickening operation is then utilized to fabricate a solid model with relatively uniform thickness (see (d)), which is passed to a mask projective SLA machine to produce a prototype of the helmet by resin (see (e) for the result). In (d), the newly reconstructed triangles are displayed in blue color and the triangles coincident with the given surface patch in (c) are displayed in gray color. It is easy to find that only triangles on the boundary of the input surface patch are split while other given triangles are remained on the resultant surface.

to the *dual contouring* (DC) algorithm [27] to convert the implicit representation of a solid into B-rep according to its zero-level isosurface. In the basic DC algorithm, the implicit function for a solid is first sampled on uniform cubic grids. A grid cell with its eight grid nodes having inconsistent *inside* (or *outside*) configurations is named as a boundary grid cell. In each boundary cell, a vertex on the resultant mesh surface is created and located at the position minimizing the *quadratic error function* (QEF) defined by the Hermite data samples on the grid edges of the cell. For each edge that has one end *inside* but the other *outside*, two triangles are constructed to link four vertices in the cells around the edge. The resultant B-rep is formed by these triangles. The basic DC algorithm is recently modified to generate intersection-free mesh surfaces [7, 28] and manifold-preserved surfaces [29]. Differently, in our approach, the triangles reconstructed from the uniformly sampled grids is required to stitch onto the existing triangles in S. We develop an extension of the DC algorithm to achieve this goal by 1) positioning the vertices onto the boundary curve ∂S of S when a cell intersects ∂S and 2) neglecting the construction of triangles at the region occupied by the given surface S. Details will be discussed in Section 4.

1.2 Contributions

Major technical contributions of our work fall in three aspects.

- A novel mesh thickening operation for solid fabrication is proposed in this paper. In literature, there is no such thickening operation available for inputs with general piecewise-linear surfaces.

- The new thickening operation will generate a new solid H lying at one side of the given surface S, where an implicit representation of this solid is defined by extending signed distance functions. Efficient filters are developed to evaluate the results of point-membership classification on the solid.

- To obtain the boundary surface, ∂H, of H, a partial surface extraction algorithm is investigated for generating the B-rep by remaining S on the resultant surface. It is very important for solid fabrication that no shape approximation error is generated on the surface to be fabricated.

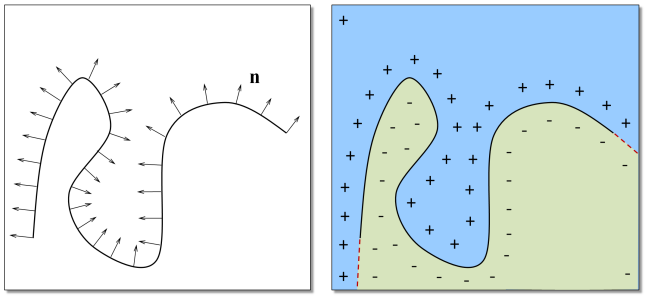

Figure 3: The signed distance-field of a given oriented surface S (left) classifies the $\Re^3$ space into three regions (see right): 1) $sDist(\mathbf{p}, S) > 0$ (the region in blue color), 2) $sDist(\mathbf{p}, S) < 0$ (the region in green color), and 3) $sDist(\mathbf{p}, S) = 0$ (the black solid curve and the red dashed curves).

Rest of the paper are organized as follows. After giving the mathematical definition of solids produced by our thickening operation in Section 2, a hierarchical filtering method is presented in Section 3 to efficiently evaluate the solid on uniformly sampled grid nodes. Section 4 concentrates on the partial surface reconstruction algorithm that generates the B-rep of ∂H. Several experimental tests are given in Section 5 to demonstrate the function of our approach, and our paper finally ends with the conclusion section.

2 Shape Representation

This section discusses the mathematical representation of solids generated by our mesh thickening operation.

Definition 1 For a given two-manifold surface patch S, the signed distance from a point $\mathbf{p} \in \Re^3$ to S is defined as

$$sDist(\mathbf{p}, S) = \begin{cases} \frac{(\mathbf{p}-\mathbf{c})\cdot\mathbf{n_c}}{|(\mathbf{p}-\mathbf{c})\cdot\mathbf{n_c}|}\|\mathbf{p}-\mathbf{c}\| & ((\mathbf{p}-\mathbf{c})\cdot\mathbf{n_c} \neq 0) \\ 0 & ((\mathbf{p}-\mathbf{c})\cdot\mathbf{n_c} = 0) \end{cases} \quad (1)$$

where $\mathbf{c}$ is the closest point of $\mathbf{p}$ on S as

$$\mathbf{c} = \arg \inf_{\forall \mathbf{q} \in S} \|\mathbf{p}-\mathbf{q}\| \quad (2)$$

and $\mathbf{n_c}$ is the surface normal[1] of S at $\mathbf{c}$.

Different from the signed distance field for solids with closed boundary surfaces, the signs for distances to an open surface are defined according to which side of S the query point $\mathbf{p}$ is located. Specifically, the sign of inner product between the vector $(\mathbf{p}-\mathbf{c})$ and the normal vector, $\mathbf{n_c}$, of surface S at $\mathbf{c}$. This is an extension of signed distance fields. The sign of distance to the surface S partitions the $\Re^3$ space into three regions (see Fig.3 for an illustration). Note that, the red dashed curves (in the right of Fig.3) represent the region of points $\mathbf{p}$ with $sDist(\mathbf{p}, S) = 0$ but $\mathbf{p}$ not on S.

[1]The input two-manifold surface S is oriented, thus its normal vectors point to one side of S.

By the signed distance function in Definition 1, we can define the point set of the thickened solid $H(S)$ for S as follows.

Definition 2 The point set of the thickened solid $H(S)$ having the thickness r for a given two-manifold surface patch S is defined as

$$H(S) = \{\mathbf{p} \mid sDist(\mathbf{p}, S) \in [-r, 0],\ \forall \mathbf{p} \in \Re^3\} \backslash B(S) \cup \partial S, \quad (3)$$

where $B(S)$ is

$$B(S) = \{\mathbf{p} \mid \arg \inf_{\forall \mathbf{q} \in S} \|\mathbf{p}-\mathbf{q}\| \in \partial S,\ \inf_{\forall \mathbf{q} \in S} \|\mathbf{p}-\mathbf{q}\| \leq r,\ \forall \mathbf{p} \in \Re^3\}, \quad (4)$$

and '\' and '$\cup$' denote the subtraction and the union of point sets respectively.

In our definition of the thickened solid, the point set of a thickened solid excludes the points whose closest points are on the boundary of S (i.e., all points in $B(S)$). As shown in Fig.4, if these points are included in the point sets for thickened solids, the given surface patch, S, will be smoothly extrapolated by the boundary surface of the thickened solid. This results in a solid H that the given surface S cannot be easily identified on the resultant ∂H. Industrial applications prefer to generate a solid on which the given surface patch, S, can be easily identified (as shown in Fig.4(d)). The points in $B(S)$ must be excluded from $H(S)$; however, the boundary of S, ∂S, must be included in $H(S)$ to result in a *regularized* solid (ref. [2]).

Note that, the solid defined in Eq.(3) is located at the 'interior' side of the given surface patch (i.e., the non-positive region in $\Re^3$ defined by $sDist(\mathbf{p}, S)$ – the green region in Fig.3). Similarly, a thickened solid in the non-negative region can be defined by $sDist(\mathbf{p}, S)$ as

$$\{\mathbf{p} \mid sDist(\mathbf{p}, S) \in [0, r],\ \forall \mathbf{p} \in \Re^3\} \setminus B(S) \cup \partial S.$$

Or to obtain such a solid by flipping the orientation of all triangles on the given surface patch S. Therefore, in the rest of our paper, we only discuss about the boundary surface generation method for the solid defined in Eq.(3).

3 Fast Evaluation of Grid Nodes

The definition of thickened solid $H(S)$ in Eq.(3) is an implicit representation, which is going to be converted into the mesh surface of ∂H. We sample the solid H on uniform grids, where each grid node stores its signed distance to S. The sampling distance (i.e., the width, w, of cubic grid cell) is a parameter that can be selected by users. However, selecting a value of w greater than half of r (i.e., the specified thickness of H) may let the newly created boundary surface ($\partial H \setminus S$) fall in the same grid cell which holds the original surface S. Such a case will result in a poor mesh surface according to the limitation of DC algorithm that each cell will generate only one vertex on the resultant mesh.

Remark 1 The width of uniformly sampled grids, w, should be less than $r/2$.

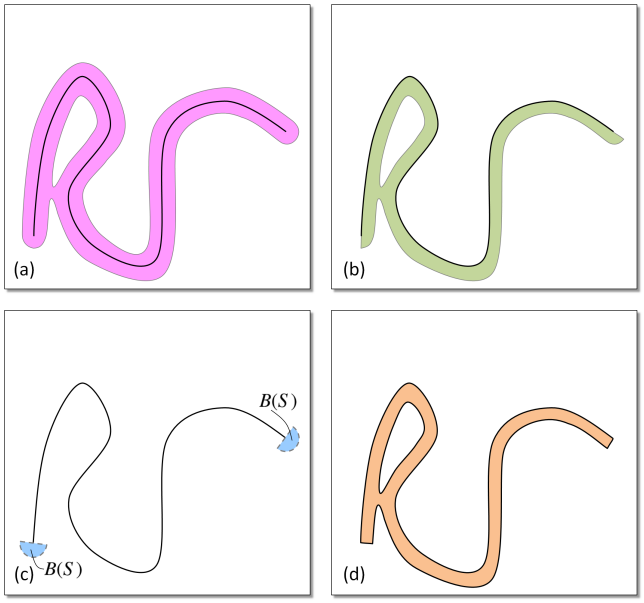

Figure 4: The illustration for the definition of a thickened solid H: (a) the given surface S and its offsetting result, (b) the solid including the points with the closest points on the surface boundary ∂S, (c) the open point set $B(S)$ as defined in Eq.(4) must be excluded, and (d) the thickened solid $H(S)$ as defined in Eq.(3). It is easy to find that the boundary of given surface is not clearly presented on the resultant solid if the points with closest points on the surface boundary ∂S are included (e.g., as the solid shown in (b)).

The space Γ that bounds a thickened solid H can be estimated by enlarging the bounding box of S with r. Therefore, the signed distance-field to S (according to Definition 1) is sampled in Γ on uniform grids with width w. To efficiently search the closest point $\mathbf{c_q}$ on S according to a query point $\mathbf{q}$, the swept sphere volume hierarchy (SSVH) of triangles in [30] is adopted here to determine the information that needs to be stored on grid nodes. As analyzed in [7], though the closest point search based on SSVH is fast, using it to determine the signed distance value for all nodes on grids with moderate density (e.g., $257 \times 257 \times 257$) may take more than several hours. However, such comprehensive search is not necessary as the surface reconstruction only considers cells intersecting ∂H. To speed up the node value assignment, we classify the grid nodes into *valid* and *invalid* ones. The definition ensures that the minimal box in the sampled distance field has all nodes with valid distance values if the box intersects the boundary of solid H.

Definition 3 For any grid node, if its distance to the boundary of solid H is less than $\sqrt{3}w$ with w being the width of grid cells, this grid node is defined as a *valid grid node*; otherwise, it is called an *invalid grid node*.

By this definition, the grid cells intersect ∂H only have valid grid nodes. Therefore, a good strategy for generating a mesh surface of ∂H is to compute a narrow signed distance field only near the valid grid nodes. However, the boundary of H consists of several parts, the construction of narrow distance-field around ∂H is more difficult than the narrow distance-field for solid offsetting in [7]. A looser bound for the set of valid grid nodes is given by introducing the *candidate region* of valid grid nodes as follows.

Definition 4 *Candidate region* Ω in $\Re^3$ of valid grid nodes is defined as a set of points, where any point $\mathbf{q} \in \Omega$ must satisfy $|Dist(\mathbf{q}, S)| \leq r + \sqrt{3}w$.

The candidate region of valid nodes defined a superset of valid grid nodes, which is actually an offset solid of S with the offset value $(r + \sqrt{3}w)$.

Remark 2 For a sphere centered at $\mathbf{o}$ with diameter d, this sphere has no intersection with the candidate region if $|sDist(\mathbf{o}, S)| > d + r + \sqrt{3}w$.

This remark is used to develop an hierarchial assigning algorithm for grid nodes. Without loss of generality, we assume that there are $(2^l + 1) \times (2^l + 1) \times (2^l + 1)$ grid nodes to be assigned for the thickened solid $H(S)$ (with l being an integer). Starting from the bounding box Φ of all these grid nodes, we recursively subdivide the boundary box into eight sub-boxes. For a sub-box, if the distance from the center $\mathbf{o}$ of its circumsphere to the given surface S is greater than $d + r + \sqrt{3}w$ (according to Remark 2), the subdivision is stopped and all grid nodes in this sub-box are assigned as *not* belonging to the candidate region. Otherwise, the sub-box is further subdivided until a sub-box with only eight grid nodes is obtained, which cannot be further refined. When reaching the finest level of the hierarchy, the signed distance from a gird node to S will be determined by [30] and stored; meanwhile, whether this grid node is inside the solid H will be determined according to Definition 2. Figure 5 illustrates the hierarchical structure (i.e., an octree) for assigning the value on grid nodes.

4 Partial Surface Reconstruction

In this section, we present the partial surface reconstruction algorithm that generates a mesh surface M for ∂H, which remains the given surface patch S as part of it. Our reconstruction algorithm is an extension of the *dual contouring* (DC) algorithm [27] conducted on the uniformly sampled grid nodes. Briefly, in DC algorithm, every grid-box with some grid node *inside* a solid H while other nodes *outside* will generate a vertex for the resultant mesh surface M. The position of vertex is determined by minimizing the *quadratic error functions* (QEF) for obtaining a good shape approximation (ref. [27]). For each grid edge having one ending node *inside* and *another* outside, a quadrangle is constructed by linking the vertices in the four grid boxes around this grid edge. In general, these four vertices are not coplanar; therefore, the quadrangle is split into two triangles to have a deterministic representation of M.

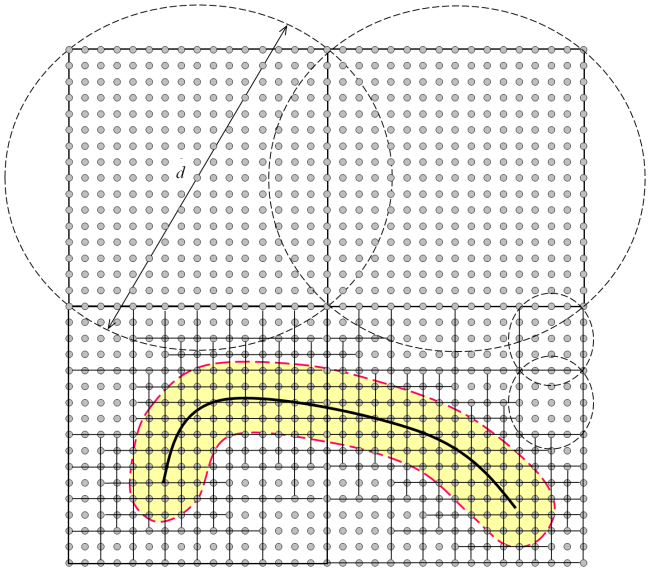

Figure 5: An example hierarchical structure which can efficiently detect the grid nodes not belonging to the candidate region (in yellow) of a given surface (illustrated by the bold black curve).

4.1 Boundary tracking and processing

First of all, in order to let the reconstructed triangles stitch to the given surface patch S, the grid cells that intersect with the boundary curves of S are determined (called *boundary-curve cells*). Different from the original dual contouring algorithm, the vertices generated in these boundary-curve cells are located in a different way. Moreover, the boundary curves ∂S of the given surface patch S are processed to eliminate the gap between the newly reconstructed surface region and S by a method similar to the boundary stitching in [31].

Instead of detecting the intersection between the boundary edges on S and all the grid cells, which is neither efficient nor robust, we construct the boundary-curve cells and track the boundary edges in them by a top-down detection algorithm. Using the bounding box Φ of all the grid nodes as a root, an octree based hierarchy is constructed. Each node on the octree spans a cubical space in $\Re^3$ and stores the boundary edges that fall in this cubical space. When constructing the octree, the refinement of a node is stopped if no boundary edge falls in the space spanned by this node. Leaf nodes of the octree at the finest level are the boundary-curve cells which intersect ∂S (see the cubes shown in Fig.6).

Unlike other boundary grid cells in the DC algorithm, vertices in the boundary-curve cells for the mesh reconstruction are generated and located in a different way. First of all, if there are more than one boundary vertices of S in a boundary-curve cell c_{bnd}, the vertex closest to the center of c_{bnd} is selected as this cell's vertex, which will be connected by triangles to form the resultant mesh, M. If there is no boundary vertex in a boundary-curve cell c_{bnd}, a new vertex will be created on one of the boundary edges in c_{bnd} and located at the place closest to the center of c_{bnd}. After constructing vertices in all the boundary-curve cells, a boundary

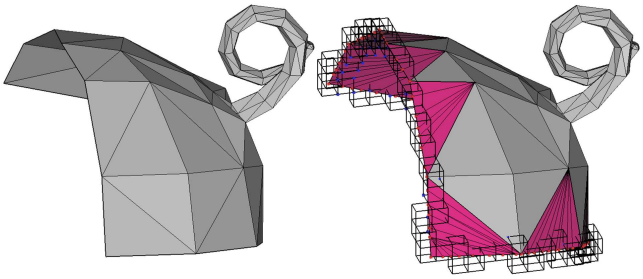

Figure 6: Boundary tracking and processing: (left) the given surface patch S and (right) the boundary-curve cells (displayed in bold black wire-frame) and the subdivided triangles adjacent to the the boundary of S (in pink color).

edge on the given mesh surface S may have several newly created vertices attached. Each triangle T_{bnd} on S adjacent to these edges will be replaced by a set of new triangles connecting the vertices associated with T_{bnd} and its edges. This can be implemented by sorting all vertices along the triangular edges and applying a minimal area triangulation (ref. [32]). See the re-triangulated faces in the right of Fig.6, which are displayed in different colors. The mesh surface $\hat{S}$ after this re-triangulation is coincident to the input surface patch S.

4.2 Face generation

This subsection discusses the method to generate polygonal faces for the surface $\partial H \setminus \hat{S}$.

Definition 4 A grid edge with one end node *inside* the solid H and another end node *outside* is an edge that intersects the boundary surface ∂H – called *boundary grid edge*.

The original DC algorithm generates a closed mesh surface by constructing a quadrilateral face for each of the boundary grid edges. However, as part of the surface ∂H has already existed in $\hat{S}$ and must be exactly remained, the face generation step must neglect the construction of faces in these regions.

Remark 3 For a boundary grid edge with two end nodes having signed distances to S as d_s and d_e, it has $d_s \geq 0$ or $d_e \geq 0$ if this edge intersects the given surface patch S.

Therefore, either $d_s \geq 0$ or $d_e \geq 0$ is a necessary condition for neglecting the face construction on a boundary grid edge.

Definition 5a A *cross-section* region on the boundary surface ∂H of H is defined as

$$\{\mathbf{p} \mid \arg \inf_{\forall \mathbf{q} \in S} \|\mathbf{p} - \mathbf{q}\| \in \partial S,\ \forall \mathbf{p} \in \partial H\}. \qquad (5)$$

Definition 5b A set of points on the boundary surface ∂H of H satisfying $sDist(\mathbf{p}, S) \neq 0$ and with their closest surface point not on ∂S is defined as the *inner-shell* surface region.

Remark 4 For a boundary grid edge intersecting ∂H at $\mathbf{p}$, if the closest point $\mathbf{c}_p$ of $\mathbf{p}$ on S satisfies $\mathbf{c}_p \in \partial S$, the face

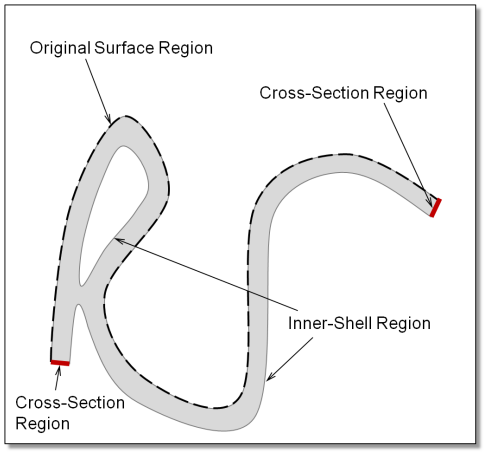

Figure 7: The boundary surface ∂H of H consists of three parts: the *inner-shell* region, the *cross-section* region and the *original surface* region.

constructed according to this boundary edge is in the cross-section region.

The boundary surface ∂H of H consists of three parts including the *cross-section* region, the *inner-shell* region and the *original surface* region (see Fig.7 for an illustration).

In our modified dual contouring algorithm, faces are only constructed on the inner-shell regions and the cross-section regions. Specifically, after constructing the boundary-curve grid cells and processing the boundary triangles on S (by the method in Section 4.1), the polygonal faces for M are constructed on the grid edges intersecting the *inner-shell* region and the *cross-section* region in three steps.

- Firstly, we obtain the intersection point between all boundary grid edges and ∂H by the bisection method. The closest point $\mathbf{c_q}$ of an intersection point $\mathbf{q}$ is then searched by the method of [30]. The Hermite data $(\mathbf{q}, \frac{\mathbf{q}-\mathbf{c_q}}{\|\mathbf{q}-\mathbf{c_q}\|})$ is stored on the grid edge for positioning vertices in the next step.
- Secondly, the vertex in each boundary grid cell is created and located at the average position of all intersection points on the grid edges of this cell. For the boundary-curve grid cells, as the vertices have been created and located on the existing boundary edges of S, no new vertex will be generated and the positions of existing vertices will not be changed.
- Thirdly, if the ends of a boundary grid edge have different signs for their signed distances to S, this edge is possible to intersect the original surface region (according to Remark 3). Therefore, we further check in which region the intersection point between this edge and ∂H is located. If the intersection point is in the original surface region, this boundary grid edge is neglected. Otherwise, a quadrilateral face will be created by connecting the vertices in four grid cells adjacent to this boundary grid edge. The Hermite data of a boundary grid edge is also stored in the face constructed on it, which will be used in the next step of shape optimization.

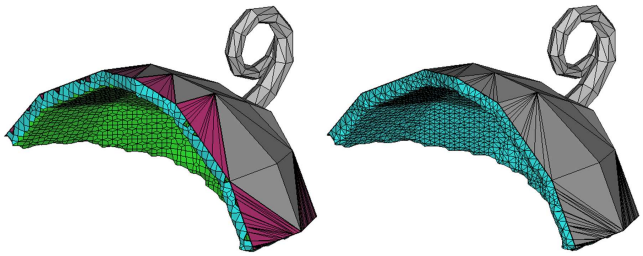

Figure 8: Face generation for the resultant mesh surface M: (left) the quadrilateral faces are generated on both the inner-shell region (in green) and the cross-section region (in blue), and (right) the holes are filled and quadrilateral faces are split into triangles.

The reconstructed quadrilateral faces will cover the inner-shell region and the cross-section region meanwhile connecting to the boundary edges of $\hat{S}$ (see Fig.8 for an example). Specifically, the faces for the inner-shell region and the cross-section region are constructed simultaneously. The faces in the cross-section region can be classified by Remark 4. Different from the original DC algorithm, the faces overlap the input surface S are neglected in this modified face generation step.

On the mesh surface generated by our extended dual contouring algorithm, a few holes may be generated near the boundary curves of the given surface patch S. Some of these holes are generated in a boundary-curve cells holding more than one boundary vertices, where only one vertex is used in the face generation step. Other holes are caused by neglecting the face generation on a boundary grid edge in mistake, which may happen because of round-off errors. Such holes can be easily filled by the minimal area triangulation method in [32]. Moreover, the non-manifold entities generated in DC algorithm can be eliminated (or repaired) by the method presented in [31]. Note that the triangles on $\hat{S}$ must not be changed. An example of the repaired mesh surface is shown in the right of Fig.8.

4.3 Shape processing

In DC based algorithms, the faces generated according to the boundary grid edges do not always interpolate the Hermite sample data stored on the boundary grid edges. The faces generated by our method above also have this problem. As shown in the left of Fig.9, the shape of reconstructed surface near the boundary between the cross-section region and the inner-shell region is not very smooth. We process the shape of reconstructed faces iteratively.

- Step 1). Apply a Laplacian operator to smooth the normal vectors of the faces located at the cross-section region.
- Step 2). Update the normal of the Hermite data stored in a face at the cross-section region by the smoothed normal vector of this face[2].

[2]This is because that the normal vector of a point $\mathbf{q}$ at the cross-section region is not $\frac{\mathbf{q}-\mathbf{c_q}}{\|\mathbf{q}-\mathbf{c_q}\|}$.

- Step 3). Compute the optimal position $\mathbf{o}_v$ of a reconstructed vertex v (i.e., not including the vertices on $\hat{S}$) by minimizing the QEF defined by the Hermite data stored in all faces adjacent to v, and move v to a new position $\mathbf{p}_v^{new} = (1-\alpha)\mathbf{p}_v + \alpha\mathbf{o}_v$ with $\mathbf{p}_v$ being the current position of v and α being a blending factor ($\alpha = 0.25$ is selected in all examples of this paper).
- Step 4). Update the normal vectors of all reconstructed faces (i.e., faces not on $\hat{S}$).

Repeatedly running these steps for 10 to 20 times will generate a smoother surface and form sharp edges between the cross-section region and the inner-shell region (see the right of Fig.9 for an example).

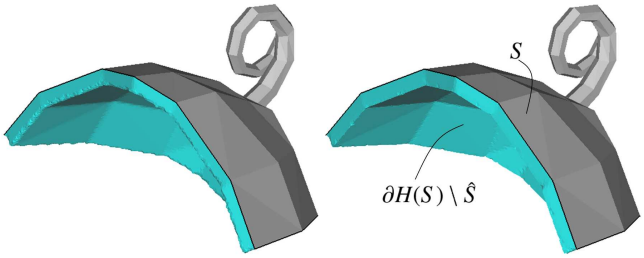

Figure 9: The mesh surface of a thickened solid before (left) vs. after (right) shape optimization. From the silhouette of the blue region, we can find the improvement of shape on the processed mesh (right). The blue region is the reconstructed boundary, $\partial H(S) \setminus \hat{S}$, of H, and the gray region is the input mesh surface.

5 Implementation Details and Results

We have implemented the proposed mesh thickening operation in a program by C++ and OpenGL. All the results shown in this paper are generated on a laptop PC with Inter Core i7 Q740 CPU plus 4GB RAM.

According to our experimental tests, the most time-consuming step in our mesh thickening algorithm is the evaluation of values on grid nodes. Even after applying the fast evaluation method presented in Section 3, the step to generate the narrow signed distance-field does still dominate the computational time of our algorithm (especially when the resolution of grid nodes is high). More than 60% of the time is spent on this step. However, the computation in this step can be easily parallelized to run on the multi-cores of CPU. Our primary implementation by using OpenMP can reduce the time in this step into around 1/4 of the original time when running on the above laptop PC with 8 cores. We test our mesh thickening operation on several examples. Table 1 gives the computational statistics of our algorithm on different examples and in different resolutions of grid nodes. It is easy to find that our algorithm can efficiently generate the mesh surface of the thickened solid from a give surface patch. Figure 10 shows an application of using our thickening operation to build a stand from a scanned surface patch of human face. Figure 11 demonstrates the application in biomedical

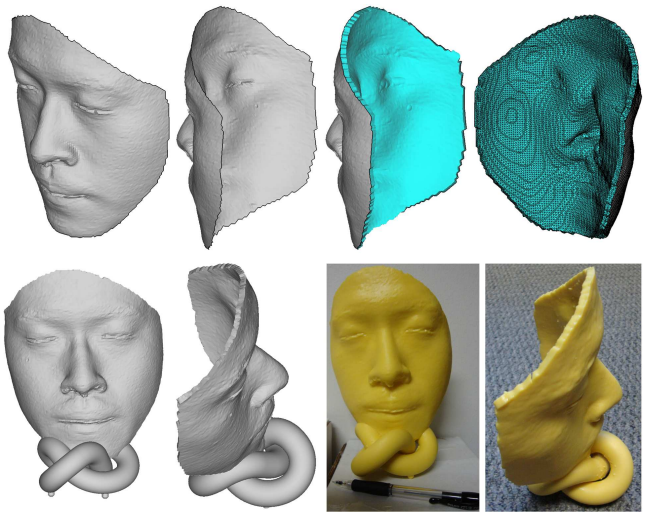

Figure 10: (Top row) The thickening result on a face model which is obtained by scanning the face of an individual. (Bottom row) The thickened solid model can be used to build a stand by solid fabrication.

engineering. Comparing to build the whole femur model, significant less processing time and materials are needed to fabricate a physical model for fitness checking of the matching faces (notice that they are not changed in our approach). Physical models in both these examples are fabricated by a mask projective SLA machine.

Notice that the method given in Definition 1 to assign signs for distance-fields requires the input mesh surface to be intersection-free. Detailed analysis about evaluating signs for the distance-field to a given mesh surface can be found in [33]. If a self-intersected mesh surface is given, incorrect sign may be given to a grid node therefore leads to an incorrect representation for the thickened solid. As shown in the left of Fig.12, unwanted solid will be generated (see the separated blue bump above the shoulder) when self-intersection occurs on the input surface patch. After removing the self-intersected triangles, a correct solid will be obtained according to Definition 1 and 2 (see also Fig.12). For the mesh surface generated by our modified dual contouring algorithm, the method presented in [7] is used to prevent the self-intersection.

When being applied to a closed mesh surface, the thickening result will be the same as "one-side" offsetting. Using a plane to clip the "one-side" offsetting shows different shape comparing to the model obtained from 1) clipping the given closed mesh model and 2) thickening the clipped open surface. See Fig.13 for an example to illustrate this difference.

Our last experimental test is to check the shape approximation error presented on the mesh surface of a thickened solid at the original surface region. Figure 14 shows analysis on the helmet example by using the publicly available PolyMeCo [34] to visualize the geometric error. It is not difficult to find that no error is presented between the given surface patch and the original surface region on the resultant mesh. The analysis of other examples in this paper gives the similar

Table 1: Computational Statistics

Model	Fig.	Thickness+	Input Trgl. No.	Time for Res. $128 \times 128 \times 128$ (sec.)			Time for Res.: $256 \times 256 \times 256$ (sec.)		
				Grid Eva.	Face Gene.	Shape Opt.	Grid Eva.	Face Gene.	Shape Opt.
Helmet	2	-2.5	10,822	1.03	1.16	1.33	6.07	4.93	5.37
Pig-Tail	9	0.5	204	0.310	0.180	0.460	1.87	0.980	1.95
Face	10	2.5	23,154	0.983	0.499	0.734	5.74	2.06	2.79
Femur	11	2.5	8,856	1.28	0.552	0.828	8.04	2.52	3.31
Sculpture	12	3.0	57,396	0.954	0.538	0.479	5.15	1.92	1.59
Repaired	13	3.0	134,136	0.850	0.654	0.743	4.29	2.21	2.33

+The input thickness is reported as a value with reference to the average edge length of the given mesh surface patch.

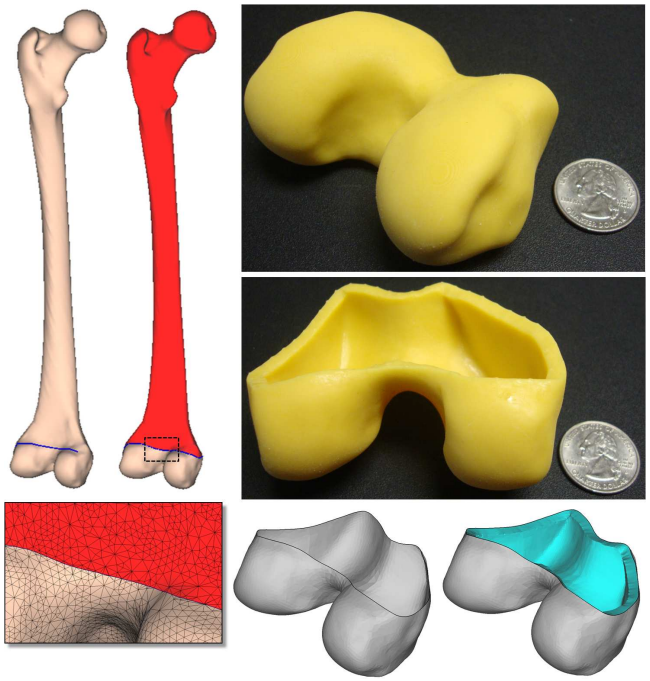

Figure 11: Application in biomedical engineering: the lower part surface of a femur model is selected to be thickened and then fabricated into a physical model which can be used in surgical planning and physical check of products such as implant or prosthetic limp.

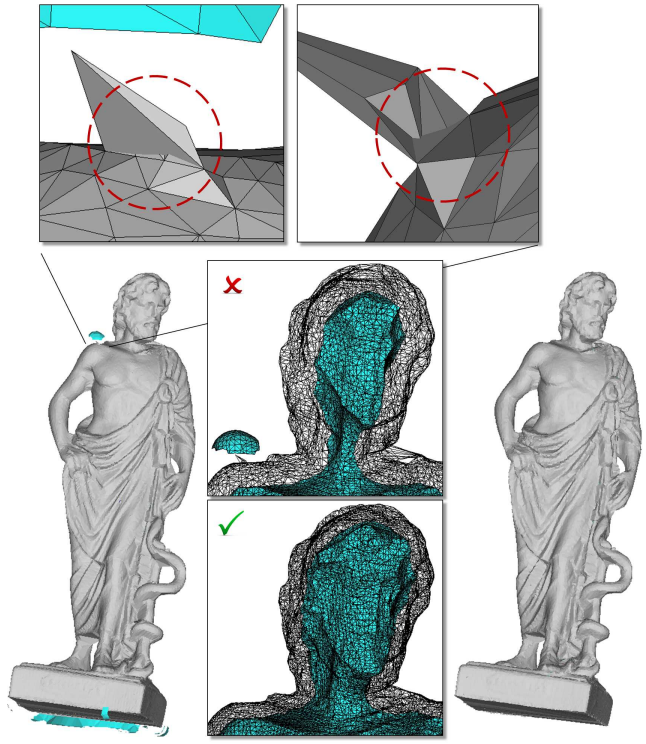

Figure 12: Input mesh model is required to be intersection-free; otherwise, incorrect result will be generated (left and top) – see the separated blue bump above shoulder. A correct result will be generated (right and bottom) after removing the self-intersected triangles on the input surface patch.

results.

6 Conclusion

In this paper, we develop a novel thickening operation to convert a given intersection-free mesh surface patch S into a solid located at one side of S for solid fabrication. The solid is represented by an implicit function defined on an extension of signed distance-fields. We developed a partial surface reconstruction algorithm to generate the boundary surface of the thicken solid, which remains the given surface S on the resultant model without introducing any shape approximation error. Moreover, the model generated by the proposed thickening operation has nearly uniform thickness. The mesh thickening operation presented in this paper is a very useful tool for solid fabrication.

Acknowledgments

The research presented in this paper was partially supported by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF): CUHK/417508 and CUHK/417109. The second author is supported by the National Science Foundation grant CMMI-0927397. Some models used in this paper are downloaded from the AIM@SHAPE Shape Repository.

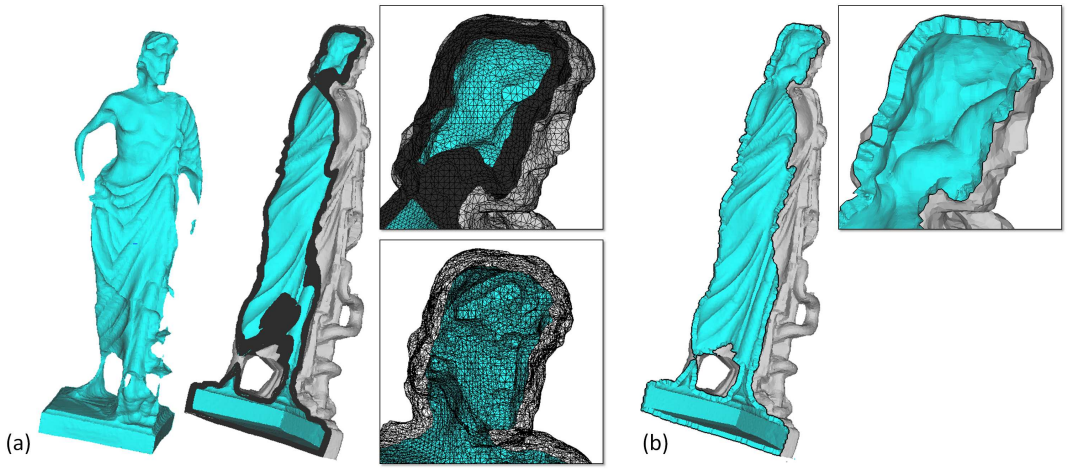

Figure 13: Hollowing vs. thickening: (a) our mesh thickening operation can also be directly applied to closed mesh surfaces to obtain the hollowing results – the left most figure shows the resultant inner mesh surface, and (b) as a comparison, the Greek sculpture model is also clipped into open surface and then conduct the thickening operation.

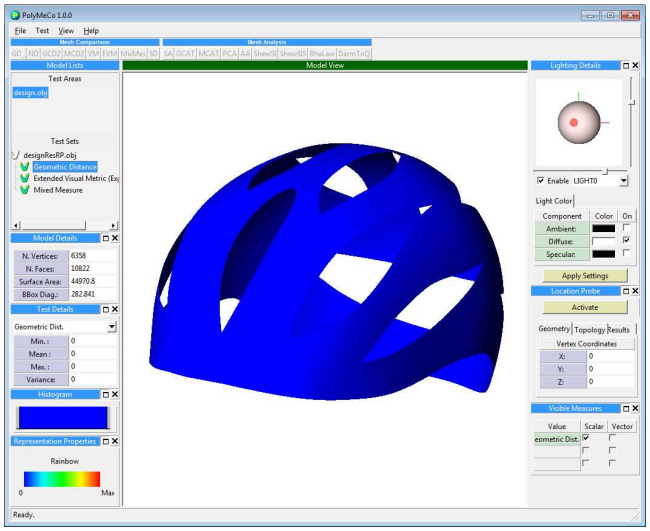

Figure 14: Using the publicly available PolyMeCo [34] to analyze and visualize the geometric error – no shape approximation error is generated on our result.

References

[1] A. Kamrani, E. A. Nasr, Rapid prototyping: theory and practice, New York: Springer, 2006.

[2] M. E. Mortenson, Geometric modeling, New York: Wiley, 1985.

[3] J.-K. Seong, G. Elber, M.-S. Kim, Trimming local and global self-intersections in offset curves/surfaces using distance maps, Computer-Aided Design 38 (2006) 183–193.

[4] D. Pekerman, G. Elber, M.-S. Kim, Self-intersection detection and elimination in freeform curves and surfaces, Computer-Aided Design 40 (2) (2008) 150 – 159.

[5] B. Bastla, B. Jüttlerb, J. Kosinkab, M. Lávičkaa, Computing exact rational offsets of quadratic triangular Bézier surface patches, Computer-Aided Design 40 (2) (2008) 197 – 209.

[6] J. R. Rossignac, A. A. G. Requicha, Offsetting operations in solid modelling, Computer Aided Geometric Design 3 (1986) 129–148.

[7] S. Liu, C. C. L. Wang, Fast intersection-free offset surface generation from freeform models with triangular meshes, IEEE Transactions on Automation Science and Engineering 8 (2) (2011) 347–360.

[8] Y. Chen, C. C. L. Wang, Uniform offsetting of polygonal model based on layered depth-normal images, Computer-Aided Design 43 (2011) 31–46.

[9] C. M. Hoffmann, Robustness in geometric computations, ASME Journal of Computing and Information Science in Engineering 1 (2001) 143–156.

[10] C. C. L. Wang, Y. Zhang, H. Sheung, From designing products to fabricating them from planar materials, IEEE Computer Graphics and Applications 30 (2010) 74–85.

[11] B. Pham, Offset curves and surfaces: a brief survey, Computer-Aided Design (1992) 223–229.

[12] T. Maekawa, An overview of offset curves and surfaces, Computer-Aided Design 31 (3) (1999) 165 –173.

[13] R. T. Farouki, Exact offset procedures for simple solids, Computer Aided Geometric Design 2 (1985) 257–279.

[14] M. Forsyth, Shelling and offsetting bodies, in: Proceedings of the third ACM symposium on Solid modeling and applications, SMA '95, 1995, pp. 373–381.

[15] J.-M. Lien, Covering minkowski sum boundary using points with applications, Computer Aided Geometric Design 25 (2008) 652–666.

[16] Y. Chen, Non-uniform offsetting and its applications in laser path planning of stereolithography machine, in: Proceedings of Solid Freeform Fabrication Symposium, Austin, Texas, USA, 2007, pp. 174–186.

[17] W. Li, S. McMains, A gpu-based voxelization approach to 3d minkowski sum computation, in: Proceedings of the 14th ACM Symposium on Solid and Physical Modeling, 2010, pp. 31–40.

[18] W. K. Chiu, S. T. Tan, Using dexels to make hollow models for rapid prototyping, Computer-Aided Design 30 (1998) 539–547.

[19] C. C. L. Wang, Computing on rays: A parallel approach for surface mesh modeling from multi-material volumetric data, Computers in Industry 62 (7) (2011) 660–671.

[20] D. E. Breen, S. Mauch, Generating shaded offset surfaces with distance, closest-point and color volumes, in: Proceedings of the International Workshop on Volume Graphics, 1999, pp. 307–320.

[21] D. E. Breen, S. Mauch, R. Whitaker, 3d scan conversion of csg models into distance volumes, in: Proceedings of the 1998 IEEE symposium on Volume visualization, 1998, pp. 7–14.

[22] X. Zhang, Y. J. Kim, D. Manocha, Reliable sweeps, in: 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling, 2009, pp. 373–378.

[23] D. Pavic, L. Kobbelt, High-resolution volumetric computation of offset surfaces with feature preservation, Computer Graphics Forum 27 (2008) 165–174.

[24] G. Varadhan, D. Manocha, Accurate minkowski sum approximation of polyhedral models, Graphical Models 68 (2006) 343–355.

[25] Y. J. Kim, G. Varadhan, M. C. Lin, D. Manocha, Fast swept volume approximation of complex polyhedral models, in: Proceedings of the eighth ACM symposium on Solid modeling and applications, 2003, pp. 11–22.

[26] M. Campen, L. Kobbelt, Polygonal boundary evaluation of minkowski sums and swept volumes, Computer Graphics Forum 29 (2010) 1613–1622.

[27] T. Ju, F. Losasso, S. Schaefer, J. Warren, Dual contouring of hermite data, ACM Transactions on Graphics 21 (2002) 339–346.

[28] T. Ju, T. Udeshi, Intersection-free contouring on an octree grid, in: Proceedings of Pacific Graphics 2006, 2006.

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

[30] E. Larsen, S. Gottschalk, M. C. Lin, D. Manocha, Fast proximity queries with swept sphere volumes, in: Proceedings of International Conference on Robotics and Automation, 2000, pp. 3719–3726.

[31] C. C. L. Wang, Approximate boolean operations on large polyhedral solids with partial mesh reconstruction, IEEE Transactions on Visualization and Computer Graphics 17 (2011) 836–849.

[32] G. Barequet, M. Sharir, Filling gaps in the boundary of a polyhedron, Computer Aided Geometric Design 12 (1995) 207–229.

[33] J. A. Baerentzen, H. Aanaes, Signed distance computation using the angle weighted pseudonormal, IEEE Transactions on Visualization and Computer Graphics 11 (2005) 243–253.

[34] S. Silva, PolyMeCo: Polygonal Mesh Analysis and Comparison Tool, http://www.ieeta.pt/polymeco/, 2008.