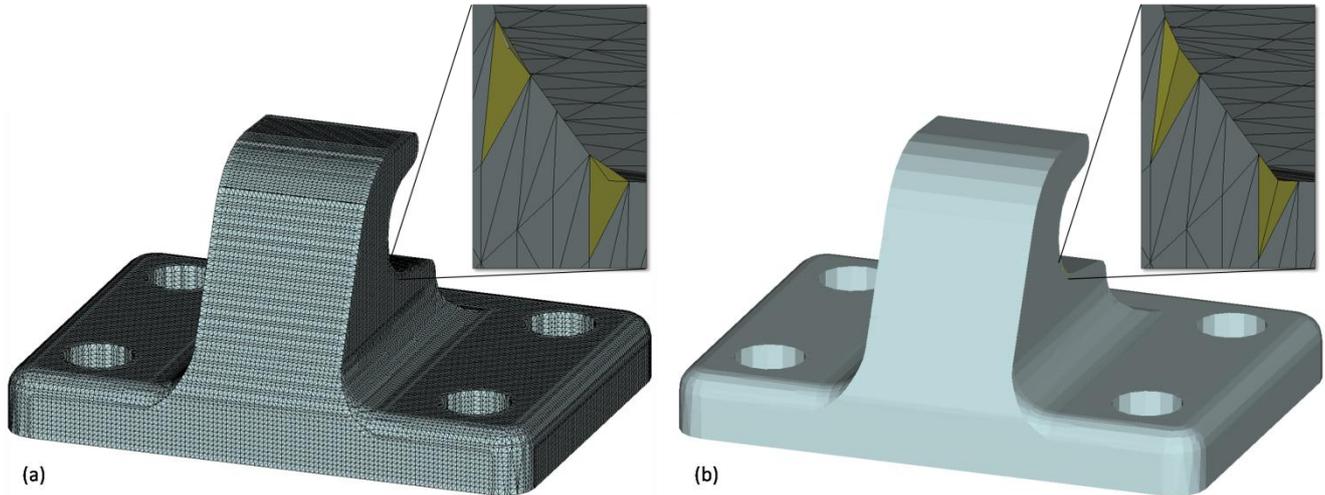


# Intersection-free Dual Contouring on Uniform Grids: An Approach Based on Convex/Concave Analysis

Charlie C. L. Wang

Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong

E-mail: [cwang@mae.cuhk.edu.hk](mailto:cwang@mae.cuhk.edu.hk)



**Figure 1.** Intersection-free dual contouring in uniform grids: (a) triangles intersect each other on the resultant mesh from dual contouring (DC) [2], and (b) our intersection-free dual contouring.

## 1. Introduction

The purpose of this research is to investigate a more efficient approach than [1] for generating intersection-free iso-surface from uniformly sampled grids using dual contouring (DC). Crack-free mesh surface can be generated by DC, and DC is able to reconstruct sharp features when Hermite data is available. However, the surface produced by Dual Contouring is rarely intersection-free. The only work existing in literature to address the problem of geometric intersection on the mesh surface generated by DC is [1], which combines the primal and dual contouring. Some necessary conditions have been derived in [1] to try to reduce the number of triangles generated in DC. However, although the conditions in [1] are sufficient, the detection involves the detection of polygon-edge intersection which is neither as robust nor as efficient as our new approach based on convex/concave detection. Also, a question is left at the end of [1]: whether it is possible to apply Rule 1 along to an intersection-free mesh surface without introducing the edge vertices. Our convex/concave analysis gives the negative answer for this question – in most cases, it is not necessary to introduce the edge vertices so that we can have an intersection-free mesh surface with the same number of triangles as the original DC algorithm (ref.[2]) when working on uniform grids. However, there are some extreme cases where edge vertices are needed.

Traditional contouring techniques, such as the Marching Cubes (MC) method [3] and its variants [4-7], fulfill the intersection-free requirement as they generate triangles enclosed in grid cells and, within each cell, the triangles are not intersecting. However, these methods typically produce much more triangles than DC when contouring the volumetric data sampled at the same grid resolution. In [1], Ju and Udeshi borrowed the idea of enclosing triangles in non-overlapping volumes (called *envelopes*) from MC to a modified DC approach that avoids potential intersections. By their method, intersection-free triangular mesh surfaces can be generated but with more triangles than the mesh surfaces generated by the original DC approach [2]. Here, we derive a convex/concave analysis based approach, which is more robust and efficient. Moreover, heuristic

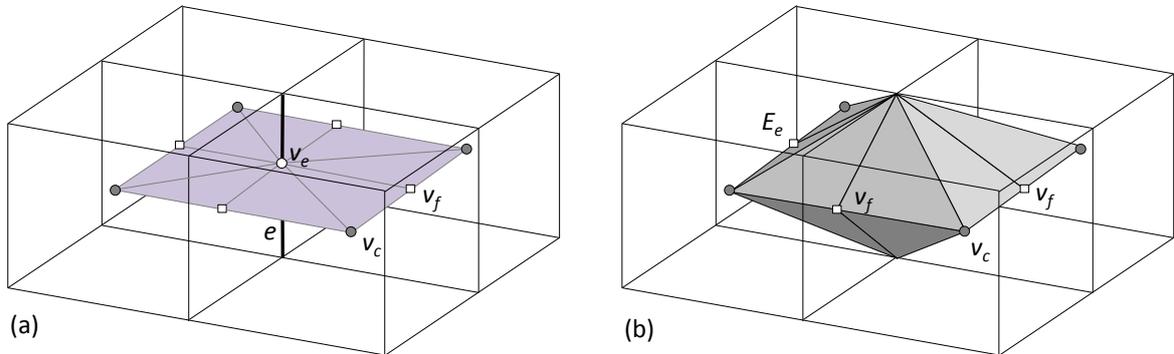
rules are developed in this work for preserving sharp-edges better on the resultant mesh surfaces. Figure 1 shows a result of our approach versus the original DC.

## 2. Preliminary

To be self-contained, the principle of intersection-free DC will be briefed in this section. More details can be found in [1, 2].

Basically, when applying the dual contouring algorithm to volumetric data sampled on uniform grids, the isosurface can be generated by linking vertices in boundary grid cells. A grid cell with its eight grid nodes have inconsistent *inside* (or *outside*) configurations is a boundary grid cell. In each boundary cell  $c$ , a vertex  $v_c$  on the resultant mesh surface is created and located at the position minimizing the Quadratic Error Function (QEF) defined by the samples on the grid edges of  $c$ , where the samples are equipped with normal vectors (i.e., Hermite data). For each cell edge  $e$  the contains a sign change – with one end *inside* but the other *outside*, two triangles will be constructed to linking four vertices in the cells around it. However, the mesh surface constructed in this way is rarely intersection-free.

A hybrid approach of primal/dual methods is developed in [1]. As illustrated in Fig.2, a triangle fan with eight triangles is generated around a cell edge  $e$  with sign change in the hybrid approach, where each triangle connects the edge vertex  $v_e$  on  $e$ , the face vertex  $v_f$  on a face  $f$  sharing  $e$  and the cell vertex  $v_c$  in a cell  $c$  sharing  $f$ . Ju and Udeshi then proved that the hybrid approach generates intersection-free triangles if each cell vertex, face vertex and edge vertex lies interior to the corresponding cell, face or edge. The envelop  $E_e$  at a grid edge  $e$  is defined as the union of eight tetrahedra, each formed by the edge  $e$ , the face vertex  $v_f$  and the cell vertex  $v_c$  (see Fig.2). It has proven that the envelopes  $E_e$  of different edges  $e$  are disjoint, and the triangles generated by the hybrid method around  $e$  are contained in separate tetrahedral of the envelope  $E_e$ . Therefore, the triangulated surface is guaranteed to be intersection-free.



**Figure 2.** Triangle fans generated by the hybrid method at a grid edge  $e$  with sign change (a), and the envelope  $E_e$  of triangles formed by the union of tetrahedra (b).

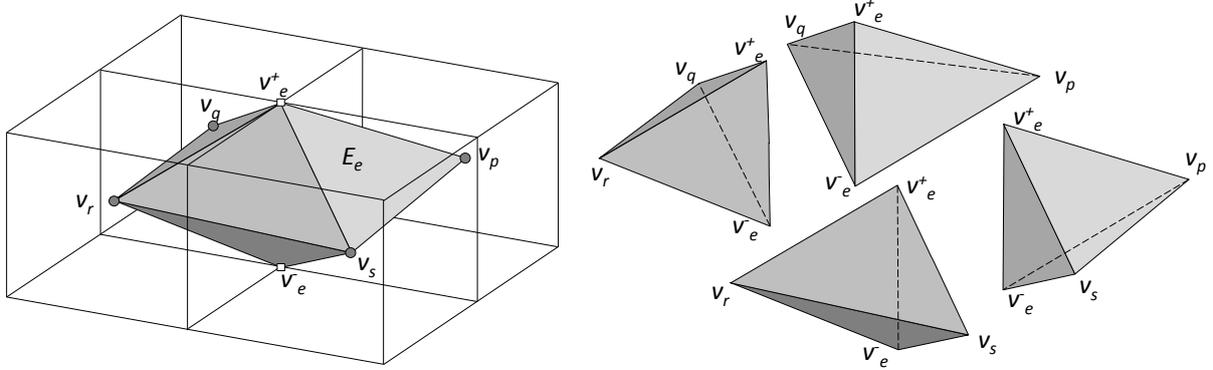
To reduce the number of triangles generated by the hybrid method, three rules are defined in [1] for contouring an Octree grid. The rules are based on the detection of polygon-edge intersections. However, when contouring uniform grids, a simpler detection method based on convex/concave analysis can be developed, which is the major contribution of our work below.

## 3. Convex/Concave Condition for Intersection-free Triangulation

A simpler convex/concave condition for intersection-free triangulation can be derived. First of all, as long as all cell vertices are located in their corresponding cell boxes, the intersection between the cell face  $f_{pq}$  of two neighboring cells  $c_p$  and  $c_q$  (containing cell vertices  $v_p$  and  $v_q$ ) and the line segment  $v_p-v_q$  is always in the face  $f_{pq}$  when contouring uniform grids. Thus, the face vertices in the hybrid method of [1] are not needed. By removing face vertices, the envelope  $E_e$  around  $e$  is then formed by four tetrahedra.

**Definition 1** Envelope  $E_e$  around an edge  $e$  that does not have volume overlap with other edge envelopes is formed by four tetrahedra:  $v_p-v_q-v^+_e-v^-_e$ ,  $v_q-v_r-v^+_e-v^-_e$ ,  $v_r-v_s-v^+_e-v^-_e$  and  $v_s-v_p-v^+_e-v^-_e$ , where  $v_p$ ,  $v_q$ ,  $v_r$  and  $v_s$  are cell vertices in cells  $c_p$ ,  $c_q$ ,  $c_r$  and  $c_s$  around  $e$ , and  $v^+_e$  and  $v^-_e$  are two vertices at the end of  $e$ .

Figure 3 gives an illustration of the four tetrahedra that form the intersection-free envelope  $E_e$  around the edge  $e$ . Note that the indices  $p$ ,  $q$ ,  $s$  and  $r$  are circularly used here.

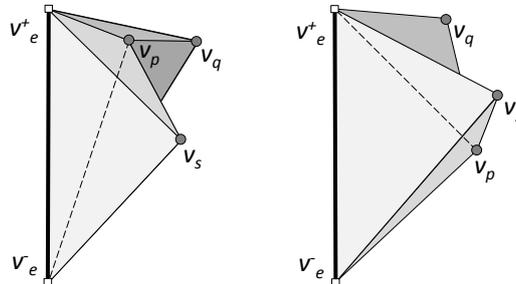


**Figure 3.** Four tetrahedra form the intersection-free envelope  $E_e$  around the edge  $e$ .

**Definition 2** For the cell vertex  $v_c$  forming the intersection-free envelope  $E_e$  around an edge  $e$ , if any of the edges  $v_c-v^+_e$  or  $v_c-v^-_e$  is concave, it is defined as a *concave envelope vertex* of  $E_e$ ; otherwise, it is named as a *convex envelope vertex*.

**Remark 1** The edge  $v_p-v^+_e$  is concave if  $v_p$  is below the plane defined by the oriented triangle  $v_s-v_q-v^+_e$ , and the edge  $v_p-v^-_e$  is concave if  $v_p$  is below the plane defined by the oriented triangle  $v_q-v_s-v^-_e$ .

Figure 4 gives an illustration for Remark 1.



**Figure 4.** Two cases that  $v_p$  becomes a concave envelope vertex: (left) the edge  $v_p-v^+_e$  is concave and (right) the edge  $v_p-v^-_e$  is concave.

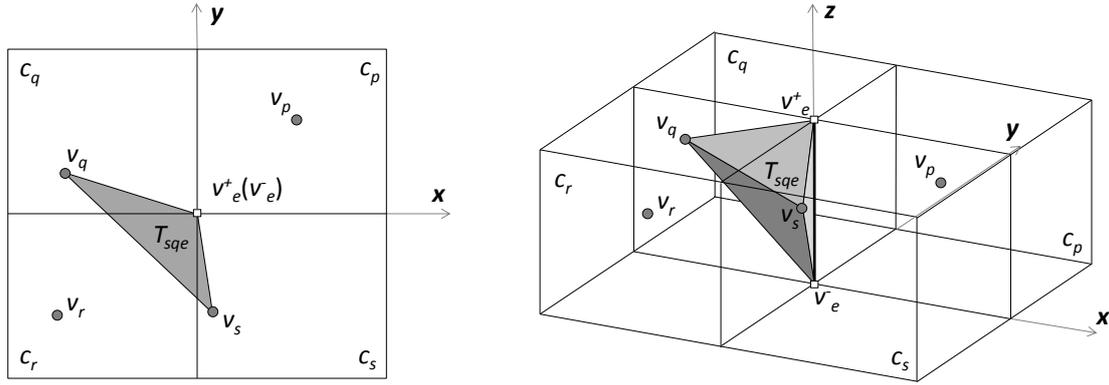
**Proposition 1** For two cell vertices  $v_s$  and  $v_q$  in two non-neighboring cells  $c_s$  and  $c_q$ , the tetrahedron  $T_{sqe}$  formed by its previous cell vertex  $v_s$ , its latter cell vertex  $v_q$  and the edge  $e$  can only intersect three of the four cells around  $e$ .

**Proof:** As illustrated in Fig.5, without loss of the generality, four cells are located in the four regions split by the  $x$ - $o$ - $z$  and  $y$ - $o$ - $z$  planes. If the tetrahedron  $T_{sqe}$  has intersected with  $c_r$  and  $p_r \in c_r$  is a point inside  $T_{sqe}$ , a point in  $T_{sqe}$  must at the same side of plane  $v_q-v^+_e-v^-_e$  as  $p_r$ . Since the plane  $v_q-v^+_e-v^-_e$  passing through the origin and  $z$ -axis which separate the volumes of  $c_r$  and  $c_p$  into two side of it, it is impossible to find a point from the cell  $c_p$  at the same side of the plane as  $p_r$ .

◇

By Remark 1 and Proposition 1, we can conclude the following remark since all points in  $c_p$  will be above the oriented planes  $v_s-v_q-v^+_e$  and  $v_q-v_s-v^-_e$  when the tetrahedron  $T_{sqe}$  intersects  $c_r$  (see Fig.5).

**Remark 2** A cell vertices  $v_p$  in the cell  $c_p$  will *NOT* be a concave envelope edge if the tetrahedron  $T_{sqe}$  formed by its previous cell vertex  $v_s$ , its latter cell vertex  $v_q$  and the edge  $e$  does not intersect the volume of cell  $c_p$ .



**Figure 5.** The tetrahedron formed by two cell vertices in diagonal cells and the edge  $e$  can only intersect three of the four cells around  $e$ .

**Proposition 2** For four cell vertices  $v_p, v_q, v_r$  and  $v_s$  around an edge  $e$ , if no concave envelope vertex is found, all the four triangles:  $v_p-v_q-v_r, v_p-v_r-v_s, v_p-v_q-v_s$  and  $v_q-v_r-v_s$  are all enclosed by  $E_e$ .

**Proof:** As shown in Fig.5, the vertex  $\bar{v}_e$  is always below the triangles:  $v_e^+-v_p-v_q, v_e^+-v_q-v_r, v_e^+-v_r-v_s$  and  $v_e^+-v_s-v_p$ , and the vertex  $v_e^+$  is always above the triangles:  $\bar{v}_e-v_p-v_q, \bar{v}_e-v_q-v_r, \bar{v}_e-v_r-v_s$  and  $\bar{v}_e-v_s-v_p$ . Therefore, the edges  $v_p-v_q, v_q-v_r, v_r-v_s$  and  $v_s-v_p$  are all convex. Since all edges are convex, the envelope  $E_e$  is a convex hull. Linear combination of three vertices on the convex hull leads to a point inside  $E_e$ . Thus, the triangles  $v_p-v_q-v_r, v_p-v_r-v_s, v_p-v_q-v_s$  and  $v_q-v_r-v_s$  are all enclosed by  $E_e$ .  $\diamond$

**Proposition 3** For four cell vertices  $v_p, v_q, v_r$  and  $v_s$  around an edge  $e$ , if only  $v_p$  is a *concave envelope vertex*, triangles  $v_p-v_q-v_r$  and  $v_p-v_r-v_s$  will be enclosed by the envelope  $E_e$ , but the triangles  $v_p-v_q-v_s$  and  $v_q-v_r-v_s$  will have some part *NOT* enclosed by  $E_e$ .

**Proof:** If only  $v_p$  is a concave envelope vertex and the edge  $v_e^+-\bar{v}_e$  has no intersection with the triangle  $v_p-v_q-v_r$ , the edge  $v_p-v_r$  will be a convex edge for the region enclosed by two tetrahedra  $v_r-v_s-v_e^+-\bar{v}_e$  and  $v_s-v_p-v_e^+-\bar{v}_e$ . This region becomes a convex hull. Thus the triangle  $v_p-v_r-v_s$  is inside  $E_e$ . The region formed by the tetrahedra  $v_p-v_q-v_e^+-\bar{v}_e$  and  $v_q-v_r-v_e^+-\bar{v}_e$  can be split into other two tetrahedra  $v_p-v_q-v_r-v_e^+$  and  $v_p-v_q-v_r-\bar{v}_e$ , which encloses the triangle  $v_p-v_q-v_r$ .

If the edge  $v_e^+-\bar{v}_e$  has an intersection with the triangle  $v_p-v_q-v_r$ , the edge  $v_p-v_r$  will be a convex edge for the region enclosed by two tetrahedra  $v_p-v_q-v_e^+-\bar{v}_e$  and  $v_q-v_r-v_e^+-\bar{v}_e$ . Thus, the region is a convex, and the triangle  $v_p-v_q-v_r$  is inside. The region formed by the tetrahedra  $v_r-v_s-v_e^+-\bar{v}_e$  and  $v_s-v_p-v_e^+-\bar{v}_e$  can be split into other two tetrahedra  $v_q-v_r-v_s-v_e^+$  and  $v_q-v_r-v_s-\bar{v}_e$ , which encloses the triangle  $v_q-v_r-v_s$ .

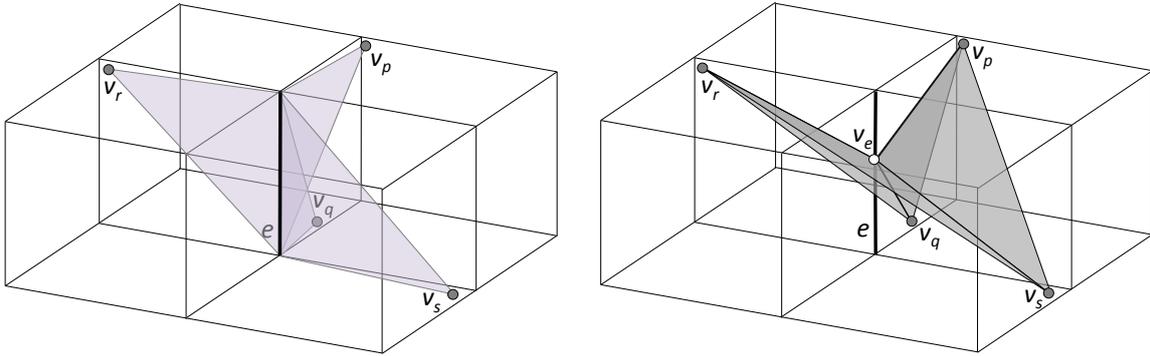
Since  $v_p$  is a *concave envelope vertex*, the edge  $v_p-v_s$  is outside  $E_e$ . Therefore, the triangles  $v_p-v_q-v_s$  and  $v_q-v_r-v_s$  will have part *NOT* enclosed by  $E_e$ .  $\diamond$

**Remark 3** For four cell vertices  $v_p, v_q, v_r$  and  $v_s$  around an edge  $e$ , only one or two vertices can be *concave envelope vertices*, and these two vertices must be in the adjacent cells.

If  $v_p$  is a *concave envelope vertex*, the tetrahedron  $T_{sqe}$  formed by  $v_s, v_q$  and the edge  $e$  can only intersect  $c_p, c_s$  and  $c_q$  but not the cell  $c_r$  (by Proposition 1). Therefore,  $v_r$  cannot be a concave envelope vertex (by Remark 2). Also, if  $v_q$  is a *concave envelope vertex*,  $v_s$  cannot be a concave envelope vertex.

**Remark 4** if two vertices are *concave envelope vertices*, none of the triangles  $v_p-v_q-v_r, v_p-v_r-v_s, v_p-v_q-v_s$  and  $v_q-v_r-v_s$  is enclosed by  $E_e$ .

If  $v_p$  is a concave envelope vertex (or  $v_r$  is a concave envelope vertex), the edge  $v_s-v_q$  is outside  $E_e$ ; and the edge  $v_p-v_r$  is outside  $E_e$  when  $v_q$  (or  $v_s$ ) is outside  $E_e$ . Figure 6 shows an example for that. If two vertices are concave envelope vertices, we have to further split them into four triangles by adding an edge vertex  $v_e$  on  $e$  to ensure that  $E_e$  encloses all the triangles.



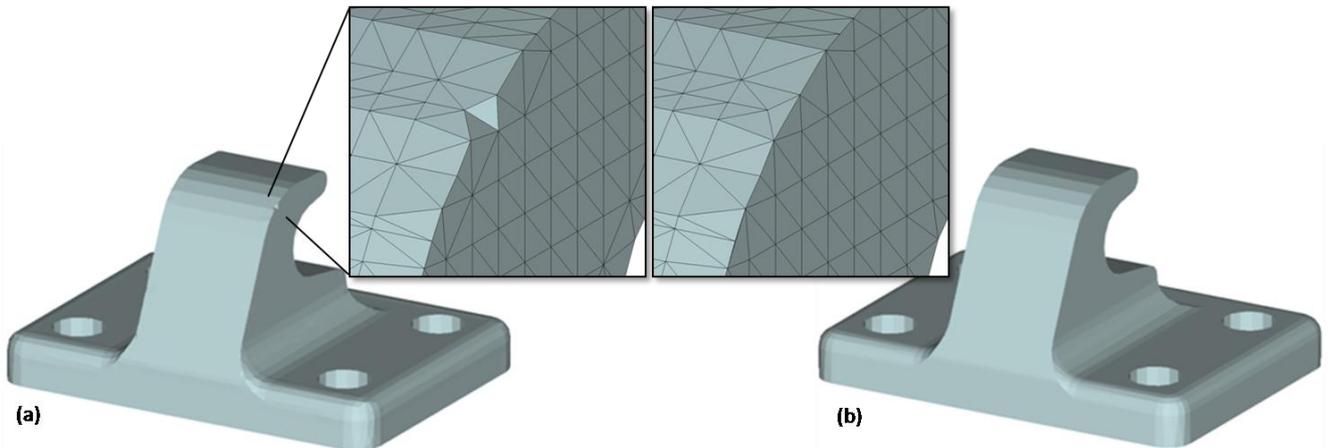
**Figure 6.** An example with two concave envelope vertices: (left) location of vertices around edge  $e$  and (right) the triangulation by inserting an edge vertex  $v_e$ .

**Analysis of Robust and Efficient Implementation:** Our convex/concave analysis based approach is more robust and efficient comparing the intersection-free approach presented in [1], which is based on the edge-polygon intersection tests. In general, 4 to 6 tests of edge-triangle intersection are needed for the triangles around a sign change edge  $e$ . By the fastest algorithm of edge-triangle intersection test in literature (ref. [8]), each edge-triangle intersection test needs 3 cross-product and 3 (or 4) dot-products. Therefore, totally 18 cross-product and 24 dot-products are required in the worst case. In our convex/concave analysis, we only need to detect whether the four vertices are concave envelope vertices. Specifically, when detecting whether a vertex  $v_p$  is below the oriented triangle  $v_s-v_q-v_e^+$  can be efficiently detected by checking if the scalar triple product below satisfies

$$(v_p - v_e^+) \cdot ((v_s - v_e^+) \times (v_q - v_e^+)) < 0.$$

In the worst case, 8 cross-products and 8 dot-products are needed. When computing scalar triple product by determination, the computation can be further reduced. Moreover, when find  $v_p$  is a concave envelope vertex, the detection of  $v_r$  can be neglected (by Remark 3). The detection on  $v_q$  and  $v_s$  can be simplified in the same way.

In the aspect of robustness, the detection based on edge-polygon intersection test is suffered by the extreme cases (e.g., the intersection occurs on the edge or vertex of the triangle). Obviously, our detection based on convex/concave analysis does not have such problem.



**Figure 7.** Sharp feature preservation: (a) the result without applying the heuristic rules and (b) the result after applying the heuristic rules for sharp feature preservation.

#### 4. Heuristic Rules for Sharp-Edge Preservation

The position of a vertex  $v_c$  in a cell  $c$  is determined by minimizing the Quadratic Error Function (QEF) defined by the Hermite samples on the grid edges of  $c$ . Although this can move vertices on the resultant mesh surface to a position on the sharp features, the subsequent triangulation on four vertices  $v_p, v_q, v_r$  and  $v_s$  around the edge  $e$  may destroy the sharp edges (see Fig.7(a) as an example). Some heuristic rules are developed in this section to preserve sharp-edges when triangulating the vertices  $v_p, v_q, v_r$  and  $v_s$  around the edge  $e$  with sign change.

First of all, whether a vertex  $v_c$  is on (or near) sharp feature is detected by the normal voting tensor constructed by the normal vectors of Hermite samples on the grid edges of the cell  $c$ . For all Hermite samples  $(p_i, n_i)$  ( $i=0, \dots, m$ ), the normal voting tensor is

$$F_n = \sum n_i \cdot n_i^T$$

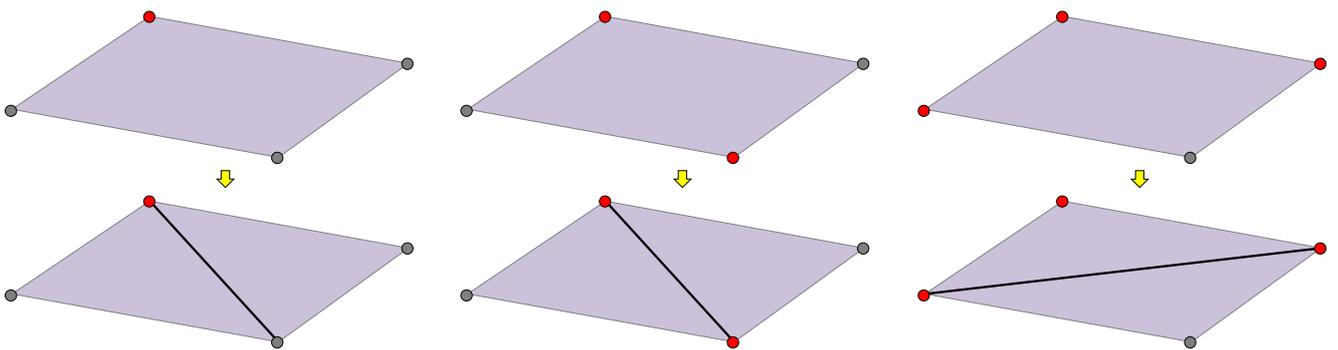
which is a  $3 \times 3$  matrix. After computing the eigen-values ( $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq 0$ ) of the normal voting tensor  $F_n$ , if there are more than two non-zero eigen-values, the vertex  $v_c$  is classified into a vertex near sharp features – named as *sharp-feature vertex*. Ideally, if  $v_c$  is on a corner, none of the eigen-values will be zero; when  $v_c$  is on a sharp edge, two eigen-values will be non-zero – see the analysis in [9]. To be robust, we consider  $|\lambda_1| < \epsilon$  as a zero eigen-value with  $\epsilon = 0.15$ . According to observation, we conclude the following rules for triangulation four cell vertices  $v_p, v_q, v_r$  and  $v_s$  around an edge  $e$  with sign change.

**Rule 1a** When triangulating four cell vertices  $v_p, v_q, v_r$  and  $v_s$  around an edge  $e$  with sign change, if only one vertex among them is sharp-feature vertex, the diagonal of triangulation must link to the sharp-feature vertex.

**Rule 1b** When there are two sharp-feature vertices and they do not belong to adjacent cells, the diagonal of triangulation must link to these two sharp-feature vertices.

**Rule 1c** When there are three sharp-feature vertices, the diagonal of triangulation must link those two sharp-feature vertices that are not in the adjacent cells.

Figure 8 shows an illustration for these rules. By applying these rules, sharp features are better preserved on the resultant mesh surfaces – Fig.7(b) shows an improved sharp edge on the resultant mesh surface. As the intersection-free triangulation has higher priority than preserving sharp features, these rules are only applied when *none* of the four cell vertices  $v_p, v_q, v_r$  and  $v_s$  are concave envelope vertices. Otherwise, the triangulation is determined by Proposition 3.

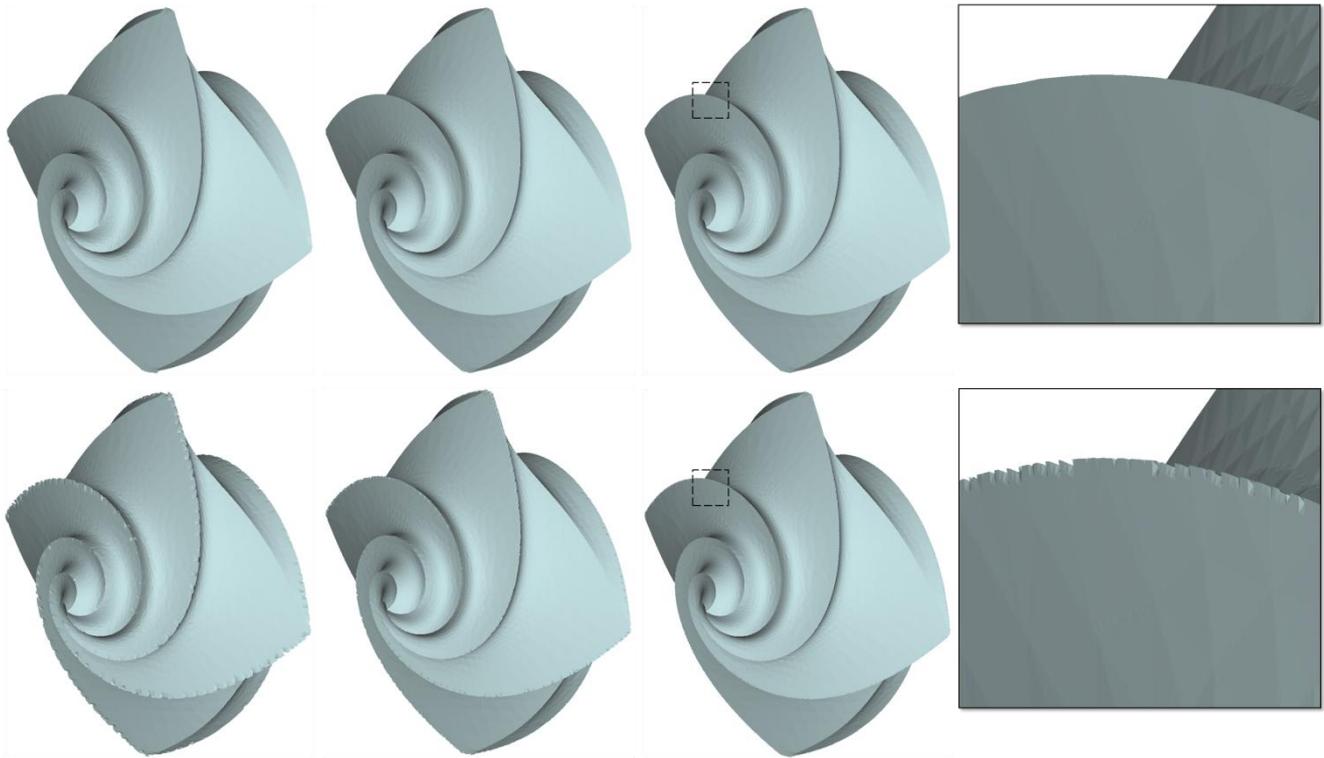


**Figure 8.** Illustration for the three rules of sharp feature preservation, where the vertices in red color are sharp-feature vertices.

#### 4. Discussion

The major drawback of this approach is that the sharp features are not well preserved after adding the constraints for enclosing the vertices and triangles in their corresponding envelope (especially when the resolution is low). For example the model in Fig.9, when not constraining the vertices and faces in their corresponding envelopes, the sharp features are well reconstructed even when the resolution is  $128 \times 128 \times$

128. After adding the intersection-free constraints, even in the resolution of 512 x 512 x 512 the thin-sharp features do still embed aliasing. Adaptive sampling and contouring can be employed to improve the quality of reconstructed thin-sharp features on the resultant mesh surfaces.



**Figure 9.** The thin-sharp features are not well preserved by adding the intersection-free constraints: (top-row) without the intersection-free constraints and (bottom-row) with the intersection-free constraints. The results are shown in different resolutions (left-column) 128 x 128 x 128, (middle-column) 256 x 256 x 256, and (right-column) 512 x 512 x 512.

#### References:

- [1] T. Ju, and T. Udeshi, "Intersection-free contouring on an octree grid", *Proc. of Pacific Graphics 2006*, IEEE Computer Society.
- [2] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data", *Prof. of SIGGRAPH 2002*, pp.339-346, 2002.
- [3] W.E. Lorensen, and H. E. Cline, "Marching cubes: a high resolution 3d surface construction algorithm", *Proc. of SIGGRAPH 87*, pp.163-169, Anaheim, California, July 1987.
- [4] J. Bloomenthal, "Polygonization of implicit surfaces", *Computer Aided Geometric Design*, vol.5, no.4, pp.341-356, 1988.
- [5] G.M. Nielson, and B. Hamann, "The asymptotic decider: resolving the ambiguity in marching cubes", *Proc. of IEEE Visualization 1991*, pp.83-93, 1991.
- [6] C. Andujar, P. Brunet P, A. Chica, I. Navazo, J. Rossignac, and A. Vinacua, "Optimizing the topological and combinatorial complexity of isosurfaces", *Computer-Aided Design*, vol.37, pp.847-857, 2005.
- [7] L.P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel, "Feature-sensitive surface extraction from volume data", *Proc. of SIGGRAPH 2001*, pp.57-66, August 2001.
- [8] N. Chirkov, "Fast 3D line segment-triangle intersection test", *Journal of Graphics, GPU, & Game Tools*, vol.10, no.3, pp.13-18, 2005.
- [9] H.S. Kim, H.K. Choi, and K.H. Lee, "Feature detection of triangular meshes based on tensor voting theory", *Computer-Aided Design*, vol.41, pp.47-58, 2009.