

Boundary Recovery for 3D Delaunay Triangulation

Yan LIU^a, S.H. LO^b, Zhen-Qun GUAN^{a,*}, Hong-Wu ZHANG^a

^a *State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics, Dalian University of Technology, Dalian 116024, China*

^b *Department of Civil Engineering, The University of Hong Kong, Pokfulam, Hong Kong, China*

*Corresponding author: Zhen-Qun GUAN

Email: guanzhq@dlut.edu.cn

Telephone: +86 411 84709730

Address: Department of Engineering Mechanics, Dalian University of Technology,
Dalian 116024, China

ABSTRACT

New ideas are presented in this paper for the boundary recovery of 3D Delaunay triangulation. Fully constrained Delaunay triangulations in terms of geometrical and topological integrity on all boundary edges and facets are required in many applications, such as meshing by components, fluid-structure interactions, parallel mesh generation, local remeshing and interface problems, etc. The geometry of boundary edges and facets can be recovered by the introduction of Steiner points. However, for a fully constrained Delaunay triangulation, these Steiner points have to be removed or repositioned towards the interior of the domain to restore the topological integrity of the boundary edges and the facets. It is found that Steiner points on edges could be removed more systematically following a specific sequence in an alternative manner rather than a random selection commonly adopted in practice; whereas for Steiner points on a facet, a weight on the Steiner point adjacency would lead to an optimal order to facilitate their removal. A linear programming technique is also employed to determine the feasible region for the relocation of Steiner points in the interior of the domain. Work examples and industrial applications with details in the boundary recovery are presented to illustrate how the algorithm works on objects with difficult boundary conditions.

Keywords: *Boundary recovery; fully constrained Delaunay triangulation; linear programming; sequence for removal of Steiner points*

1. INTRODUCTION

Efficient and robust automatic mesh generation plays a significant role in computational science and engineering [1-4]. For automatic mesh generation over complex domains, unstructured tetrahedron meshing is widely employed due to its inherent theoretical and implementation merits. Basic meshing algorithms such as Advancing-front [5, 6], Octree [7] and Delaunay triangulation [8-12] have been proposed and implemented over the last three decades. Among these methods, Delaunay triangulation is the most popular and well studied. While the 2D Delaunay triangulation method is mature and fully investigated, in three dimensions, there are still a number of difficulties including the boundary recovery problem.

Taking the triangular surface mesh as input, *boundary recovery* requests all triangles on the input surface mesh be preserved without introducing Steiner points on the boundary. The integrity of the boundary (both geometry and topology) is crucial in meshing by components, parallel mesh generation, partial remeshing and multi-interface problems. With the development of the Finite Element Method and its applications, how to generate a tetrahedral mesh with an integral boundary is increasingly important [13-17].

Generally, the existing boundary recovery approaches can be grouped into two categories: (1) local mesh reconnection; (2) introduction of Steiner points. In the first approach, no Steiner point will exist in the final tetrahedron mesh. However, there is no theoretical basis for the success of such methods due to the existence of Schönhardt configuration. In the second approach, Steiner points are introduced to achieve boundary recovery. Although they could recover the geometry of the missing quantities, as how to remove the non-positive tetrahedral elements so created systematically to ensure topological integrity is still an open issue.

1.1 Local mesh reconnection

As no Steiner points are inserted in the final mesh, approaches by local mesh reconnection attract interest of many researchers. Weatherill and Hassan [18] split the constrained boundary by inserting Steiner points on edges and facets, then suppressed the inserted points by locally remeshing tetrahedra linked to them. However, the remeshing routine could not guarantee a valid boundary-recovered topological structure even though it exists. Liu et al. [16] employed an exhaustive method named small polyhedron reconnection (SPR) to achieve the boundary recovery. For a *small* polyhedron with no more than 20 triangular facets, SPR evaluates all possible topological structures to detect if the missing quantities could be recovered in one of the configurations. Ghadyani et al. [17] carved out a hole in the vicinity of missing boundary facets, and the hole and missing boundary facets formed a polyhedron, which is to be meshed by a method known as LAST RESORT.

1.2 Introduction of Steiner points

With the help of Steiner points, missing edges and facets could be recovered. An edge is represented as broken line segments separated by Steiner points, and facets are

represented by a concatenation of sub-triangles supported on Steiner points. Such meshes are known as semi-constrained Delaunay triangulation; and if topological integrity of boundary edges and facets are not required, the boundary recovery process can stop at this point. However, if a fully constrained Delaunay triangulation including topological integrity is required, all Steiner points have to be removed or repositioned towards the interior of the domain. George et al. [13] improved their previous work [19] to propose a method to mesh an arbitrary polyhedron. By this method Steiner points are inserted on constrained boundary as Weatherill and Hassan [18] did, which are suppressed one by one later in a separate process. For each Steiner point, a Delaunay triangulation on missing facets is constructed and flat elements of zero volume are created to recover the related topological structure. Du and Wang [14] inserted Steiner points on boundaries through a heuristic approach [20] to reduce the number of Steiner points. They used an edge-swap procedure on the missing boundary facets to remove some of the Steiner points. However, there is some drawback in the method proposed by George et al. [13] and Du and Wang [14]: a number of *locked* Steiner points are generated which could not be easily removed. Chen et al. [21] combined the work of Du and Wang [14] and Liu et al. [16]. They employed SPR to reduce the number of Steiner points in the final mesh. However, their method still faces the problem of *locked* Steiner points. Guan et al. [22] proposed a technique named ‘dressing wound’. Compared to the method proposed by George et al. [13] or Du and Wang [14], it introduces more Steiner points. However, this method provides a new perspective for the boundary recovery problem.

In summary, boundary recovery for meshing arbitrary polyhedrons is still difficult. New ideas for the boundary recovery of the 3D Delaunay triangulation are presented in this paper to address some of the difficulties. The method proposed focuses on systematic removal of Steiner points. As a departure from the previous works [13, 14, 21, 22], in the process of removing Steiner points, the proposed approach optimizes the sequence and locations in the removal of Steiner points to reduce the number of *locked* Steiner points as much as possible. Moreover, a linear programming optimization is adopted to determine the feasible region in relocating Steiner points. Compared with Laplacian smoothing based methods, it guarantees finding feasible positions for Steiner points should they exist.

2. INSERTION ALGORITHM AND BOUNDARY RECOVERY

Delaunay triangulation is the geometric dual of a Voronoi tessellation. It was presented in 1934 by Delaunay who found the property of empty circumsphere criterion [23]. In 3D domain, empty circumsphere criterion can be presented as below: for every tetrahedron there are no vertices in its circumsphere except for the four vertices of its own. In 1981, Bowyer [24] and Watson [25] independently proposed the method of point insertion, which is simple and efficient. The kernel of this method is to insert points one by one until all the points are treated. In order to satisfy the empty circumsphere criterion, the topology of the mesh is corrected after each point insertion.

For a polyhedron represented by a set of triangles, the 3D Delaunay triangulation by point insertion will generate the boundary facets of a convex polyhedron, but there is no guarantee for a concave or multi-connected polyhedron. For the boundary integrity of a general polyhedron, additional work needed be done after point insertion. After partial recovery of some edges and facets by local element swaps 2-3, 3-2 and 4-4 [26], Steiner points can then be inserted on the boundary to recover the remaining boundary edges and facets. The Steiner points have to be removed or repositioned to ensure the topological integrity of the boundary. *Algorithm 1* presents the main ideas of the constrained boundary recovery procedure proposed in this paper.

Algorithm 1. Boundary recovery for 3D Delaunay triangulation

Step 1: Take the given polyhedron as input. Set original triangular facets on the boundary as the restricted boundary.

Step 2: Perform Bowyer-Watson algorithm for points on the boundary of the polyhedron, and obtain an initial triangulation of the boundary points.

Step 3: Carry out local element swaps to recover boundary edges and facets.

Step 4: Insert Steiner points to unrecovered boundary edges and facets.

Step 5: Suppress Steiner points on boundary edges.

Step 6: Optimize the tetrahedral mesh, and eliminate flat elements.

Step 7: Suppress Steiner points on boundary facets.

Step 8: Optimize the tetrahedral mesh, and eliminate flat elements.

Step 9: Delete tetrahedra outside the model to obtain the final tetrahedral mesh.

3. DETAILED ALGORITHM FOR BOUNDARY RECOVERY

The recovery of the geometry of a triangulated boundary has been studied by many researchers. Effective methods [15, 18, 20] were proposed to resolve this problem. For the geometric boundary recovery, a method similar to that proposed by George et al. [13] is adopted in this paper, in which intersection points on the boundary facets are taken as Steiner points, which are inserted by means of an enhanced Delaunay insertion to preserve the existing boundary facets. Once the boundary facets are recovered, the next phrase is to remove the Steiner points which have been inserted onto the boundary facets to assist their recovery. Points on the boundary can be removed through local reconnections and the creation of flat elements of zero volume [13, 14], but no systematic procedure is given as how these elements could be effectively opened up. Hence, the focus of this paper is to devise a systematic procedure to remove the elements so created in the boundary recovery process.

Before we could present a detailed account of the proposed method, the notion and the definition of *locked* points have to be clarified.

Locked point: A point will be said to be *locked* if a movement in any direction will render one or more tetrahedral element(s) connected to it to be negative.

As shown in Figure 1, S_1 is a *locked* point and S_2 is not. Specifically, in Figure 1, ACB and ABD are boundary facets, tetrahedra ACS_1S_2 , DAS_1S_2 , $ACBS_1$ and $ABDS_1$ are non-positive elements. To remove S_1 and S_2 from the boundary facets, the possible direction to reposition the points is along the direction normal to triangle BAD . There is no problem to relocate S_2 as both tetrahedra ACS_1S_2 and DAS_1S_2 will be opened up to have a positive volume. However if S_1 is to move upwards, tetrahedra ACS_1S_2 and DAS_1S_2 will be inverted to have a negative volume; thus point S_1 is *locked*.

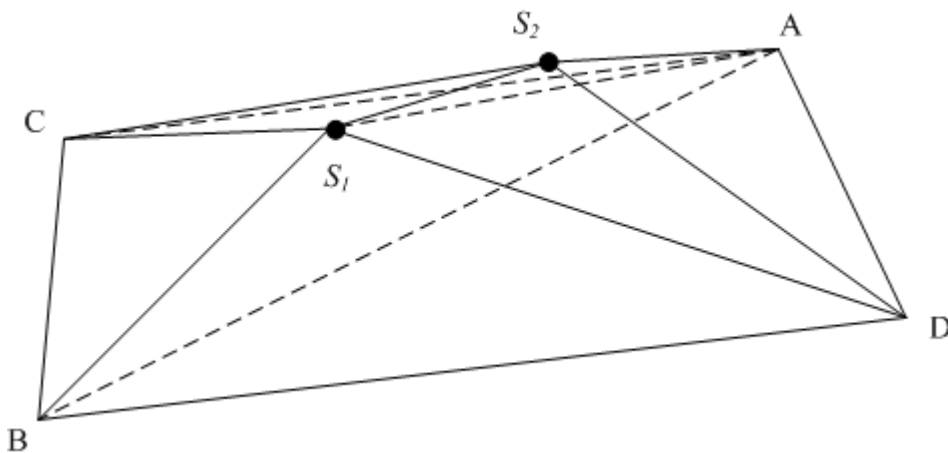


Figure 1. S_1 is a *locked* point and S_2 is free to move upwards.

As the movement of *locked* points is restricted, and they will significantly increase the difficulty in the removal of non-positive elements, an effective boundary recovery procedure should produce as few *locked* points as possible in the recovery process.

3.1 Suppressing Steiner points to recover boundary edges

Steiner point removal consists of lifting point on the facets and edges of the boundary. When an edge is broken up by several Steiner points, the order of point removal is crucial for reducing the number of *locked* points.

3.1.1 Removal of one Steiner point

As shown in Figure 2a, P is a Steiner point on the edge shared by facets F_1 and F_2 . To suppress P , triangles on F_1 and F_2 are reconnected through edge swaps as shown in Figures 2b and 2c. Then triangles acb and abg are formed respectively on F_1 and F_2 to recover the missing edge. In the reconnection process, elements $decP$, $cebP$, $cbaP$, $abgP$ and $bfgP$ are created and P is repositioned upwards normal to facets F_1 and F_2 as shown in Figure 2d.

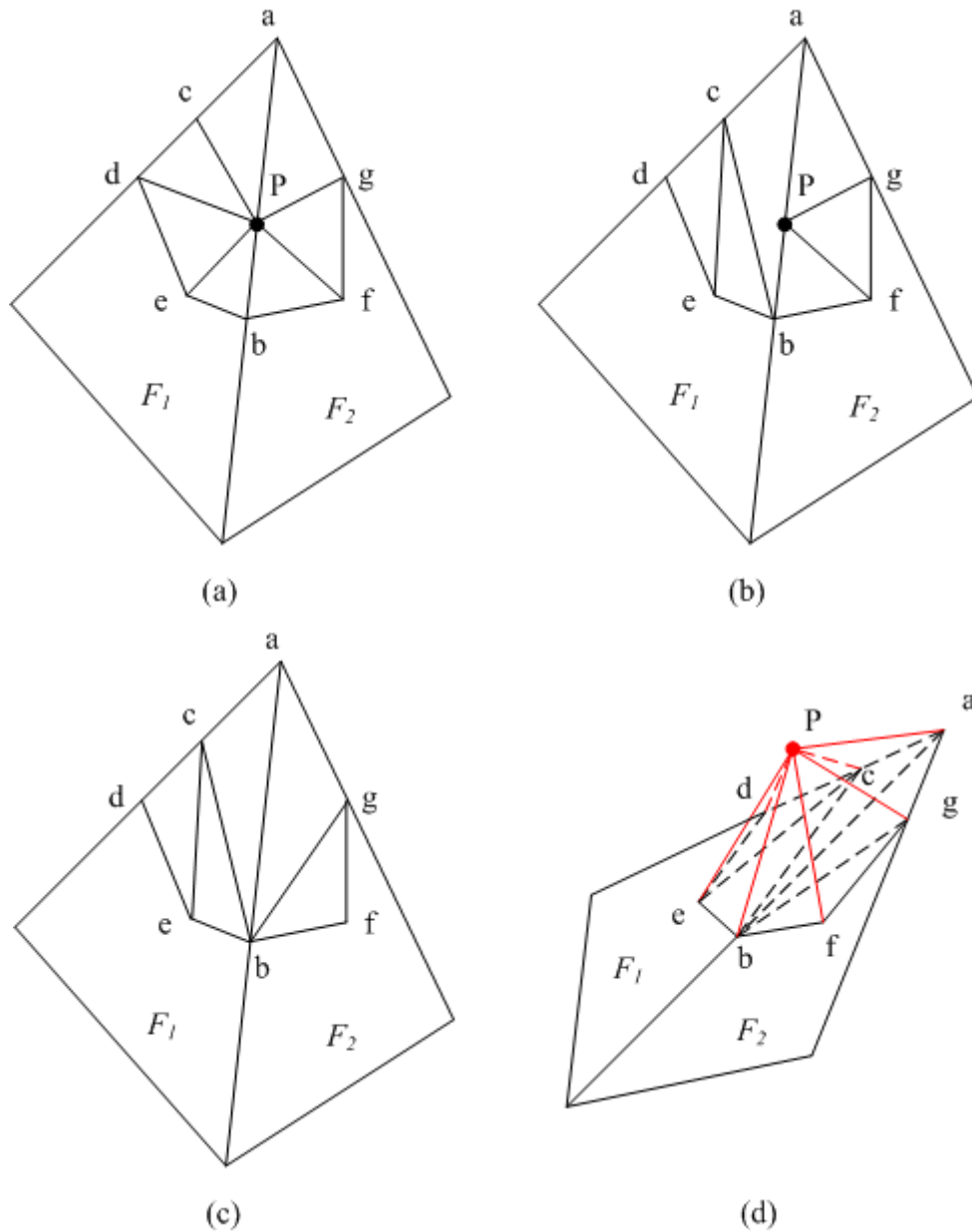


Figure 2. Removal of Steiner point on a boundary edge:
 (a) Steiner point P on boundary edge; (b) Suppressing P on F_1 ;
 (c) Suppressing P on F_2 ; (d) segment ab is recovered.

3.1.2 Optimization of the order in removing Steiner point

Suppose boundary edge E is broken up by a series of Steiner points $\{S_i : i = 1, 2, \dots, n\}$ counting from one end of E to the other end. To recover edge E , we first deal with points $S_1, S_3, \dots, S_{2k-1}$ $k = (n+1)/2$, and then the rest of the Steiner points S_2, S_4, \dots, S_{2k} $k = n/2$. By dividing the Steiner points into two groups, only half of the points, namely, S_2, S_4, \dots, S_{2k} are *locked* points. A boundary edge AB with three Steiner

points S_1 , S_2 and S_3 as shown in Figure 3 is taken as an example, and the result will be compared with the procedure by random point selection.

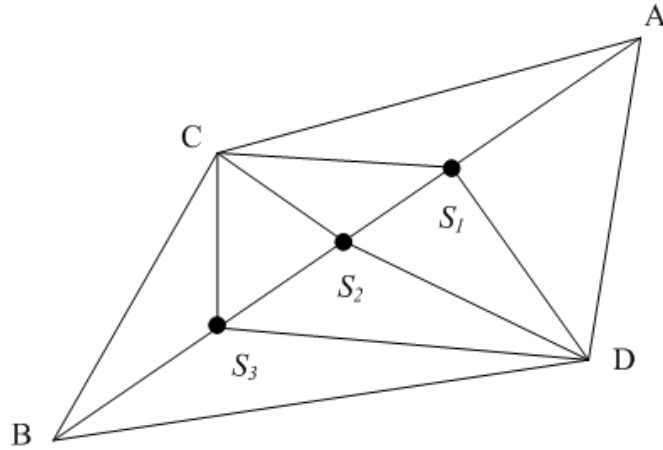


Figure 3. Boundary edge with three Steiner points.

The proposed method removes the Steiner points in the order of S_1 , S_3 and S_2 . The results are depicted in Figure 4a and 4b. In the process of recovering edge AB , flat elements AS_1S_2C , AS_1S_2D , BS_2S_3C , BS_2S_3D , $ABCS_2$ and $ABDS_2$ are created. S_2 is *locked* as wherever S_2 is repositioned at least one of the above six elements will be made negative. S_1 and S_3 are not *locked* as they could move upwards normal to facet CBA or BAD without producing any negative tetrahedral element(s).

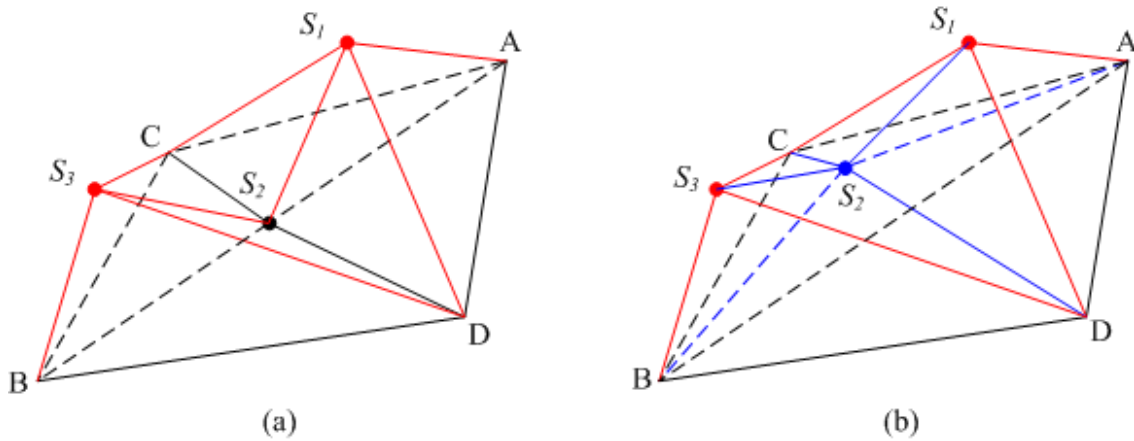


Figure 4. Edge recovered through the proposed sequence. Colours indicate different steps of node reposition.

When Steiner points are removed by a random selection, there are six possible suppression orders: $S_1S_2S_3$, $S_1S_3S_2$, $S_2S_1S_3$, $S_2S_3S_1$, $S_3S_1S_2$ and $S_3S_2S_1$. And four of them (i.e. $S_1S_2S_3$, $S_2S_1S_3$, $S_2S_3S_1$ and $S_3S_2S_1$) will produce two *locked* points. In other words, methods by random point selection have 66.6% chance to create one extra *locked* point in this 3-Steiner-point edge recovery. In removing Steiner points S_1 , S_2 and S_3 , flat elements AS_1S_2C , AS_1S_2D , AS_2S_3C , AS_2S_3D , $ABCS_3$ and $ABDS_3$ are created, which

makes S_2 and S_3 *locked*. In this case, node reposition takes three steps to open up all these flat elements as shown in Figures 5a, 5b and 5c.

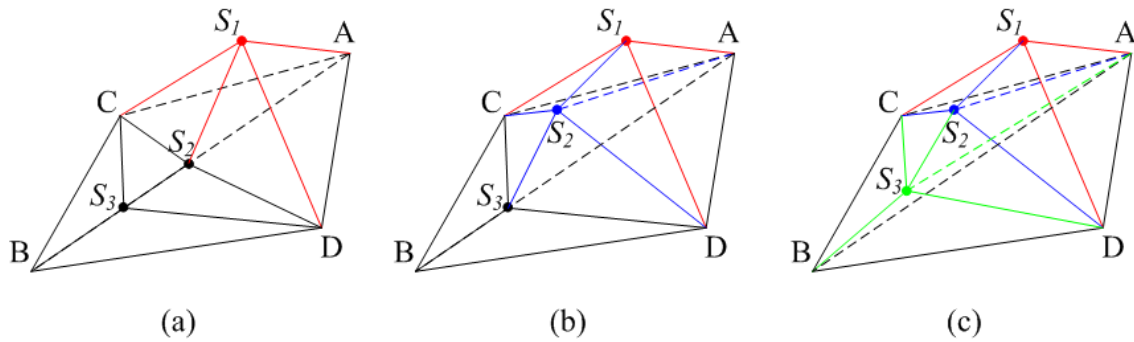


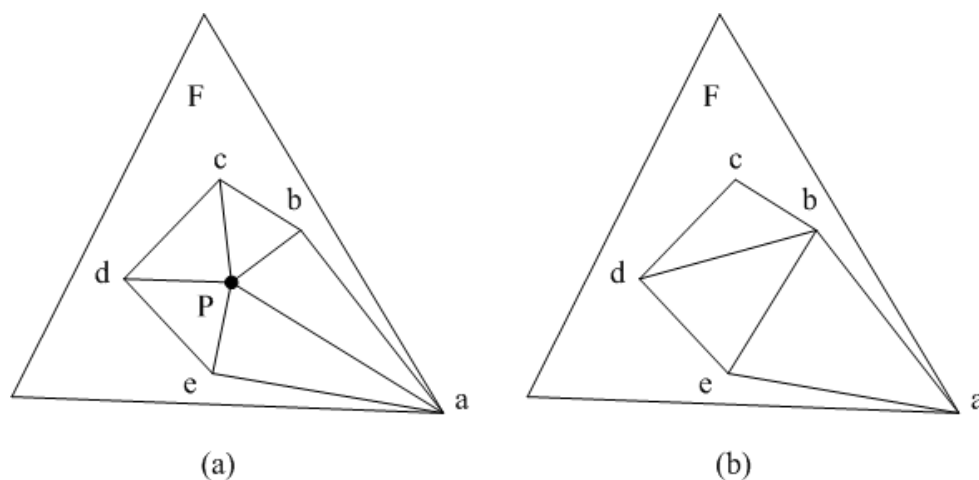
Figure 5. Edge recovered by random order. Colours indicate different steps of node reposition.

3.2 Suppressing Steiner points to recover boundary facets

Boundary face recovery is performed after all boundary edges are recovered and all flat elements are opened up. Steiner points, which are connected locally to form a valid topological structure with possibly some flat elements, are then lifted from the boundary faces by means of an optimization process. Two special cases for boundary face recovery are discussed in details to elucidate the idea.

3.2.1 Removal of one Steiner point

Edges connected to Steiner point P on the boundary facet are removed by proper element swaps. In Figure 6a, Steiner point P is on triangular facet F . P could be suppressed by swapping Pa with be and Pc with bd as shown in Figure 6b. As shown in Figure 6c, flat elements $Pabe$, $Pbde$ and $Pbcd$ are created to recover the correct topological structure.



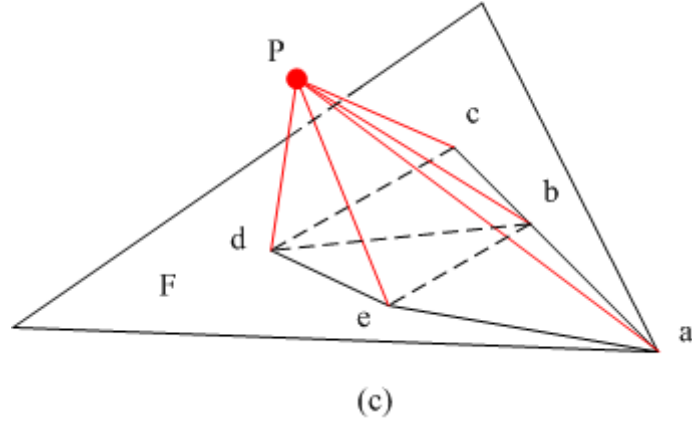


Figure 6: Point removal and topology recovery for a boundary facet: (a) Steiner point P on a triangular facet; (b) Removing P on F ; (c) Topology recovered

3.2.2 Optimization of the order of Steiner point removal

When there are more than one Steiner points on a facet, the order of point removal is crucial. A proper sequence could significantly reduce the number of *locked* points. The central idea in optimizing the order of suppression is to give priority to the Steiner point with the least number of Steiner point neighbors. In other words, for each Steiner point on a facet we take the number of its Steiner point neighbors as weight; and Steiner points are removed following the attached weight in an ascending order.

To be more specific, the operations of a boundary facet with four Steiner points are discussed in details as shown in Figure 7a. The weights of Steiner points S_1 , S_2 , S_3 and S_4 are 3, 2, 2 and 3 respectively. According to the proposed scheme, the order for recovery is S_2 , S_3 , S_4 and S_1 and elements $S_1CS_4S_2$, S_1BCS_2 , S_1ABS_3 , $S_1S_4AS_3$, S_1CAS_4 and $ABCS_1$ are created as shown in Figure 7b. The above weighted method produces two *locked* points S_1 and S_4 . On the other hand, if Steiner points are removed from the boundary facet by a random selection, there are totally 24 possible suppression orders, and 20 (83.3%) of them will create three *locked* points. For instance, one of the orders is to suppress S_1 , S_2 , S_3 and S_4 sequentially. As shown in Figure 7c, the elements created are $S_2S_4BS_1$, $S_3BS_4S_1$, CS_4BS_2 , ABS_4S_3 and $ABCS_4$. Consequently, three points instead of two points S_2 , S_3 and S_4 become *locked*.

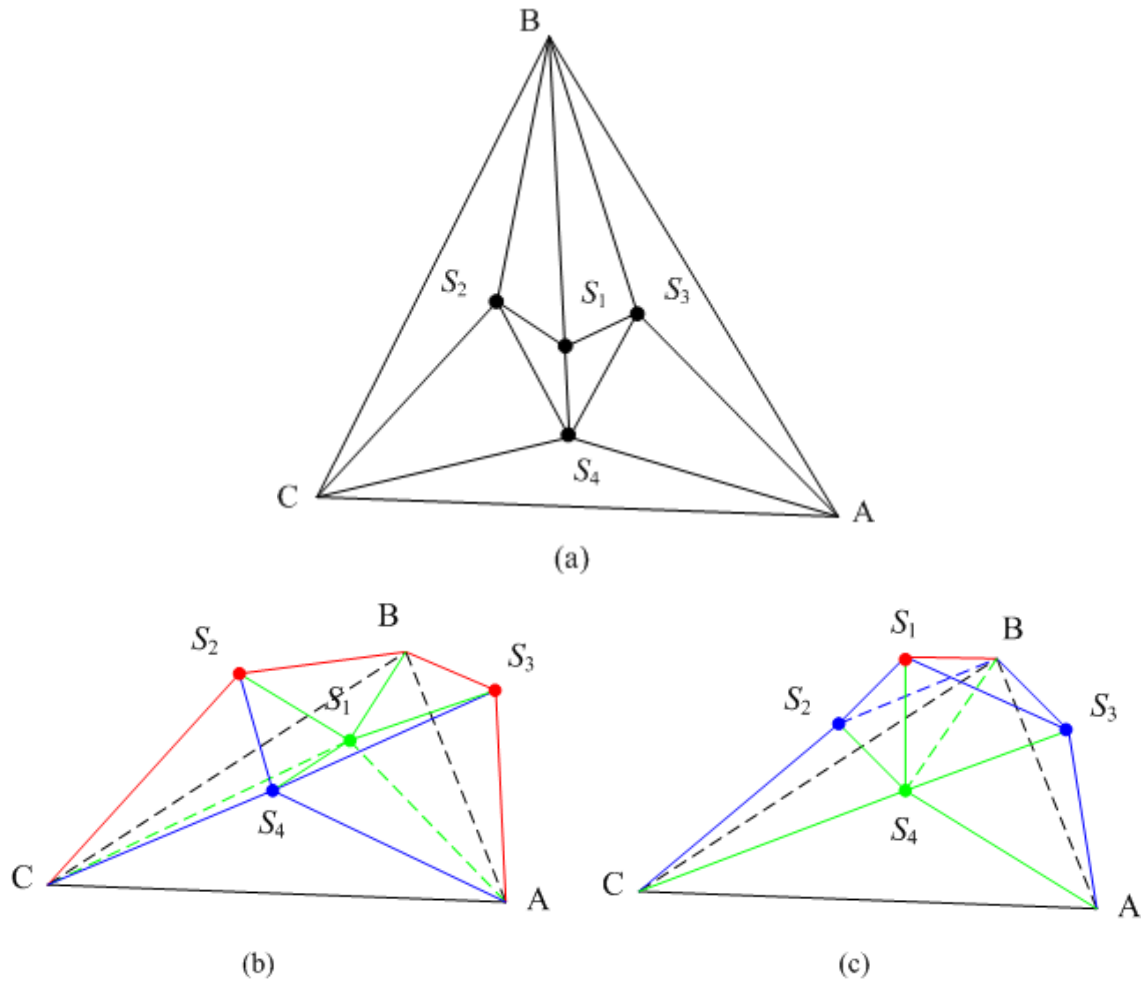


Figure 7. Recovering a boundary facet with four Steiner points: (a) Missing facet; (b) Sequence S_2, S_3, S_4, S_1 ; (c) Sequence S_1, S_2, S_3, S_4 . Colours indicate different steps of node reposition.

Solutions for special cases of equal weighting

In general, if the weights (number of Steiner point neighbors) of the Steiner points are the same as shown in Figure 8a and 9a, the proposed weighted method will not reduce the number of *locked* points. However, under this circumstance, an extra Steiner point could be introduced to reduce the *locked* points. Recovery procedures for the cases of 3 Steiner points and 4 Steiner points are described as follows, where points not *locked* are moved slightly in the figures.

Case I: 3 Steiner points

As shown in Figure 8b, an extra Steiner point P is created at the barycenter of S_1, S_2 and S_3 . The missing boundary facet ABC is recovered through the creation of elements $PABC, PS_1S_2S_3, PAS_2S_3, PBS_1S_3, PCS_1S_2, PS_1BC, PS_2AC$ and PS_3AB . Consequently, there is only one *locked* point P in the recovery of facet ABC . Unlike our approach, the

method of George and Du creates elements $S_1S_2S_3B$, S_1S_2BC , S_2S_3CB , S_2S_3CA and S_3ABC as shown in Figure 8c, in which S_2 and S_3 are *locked* points.

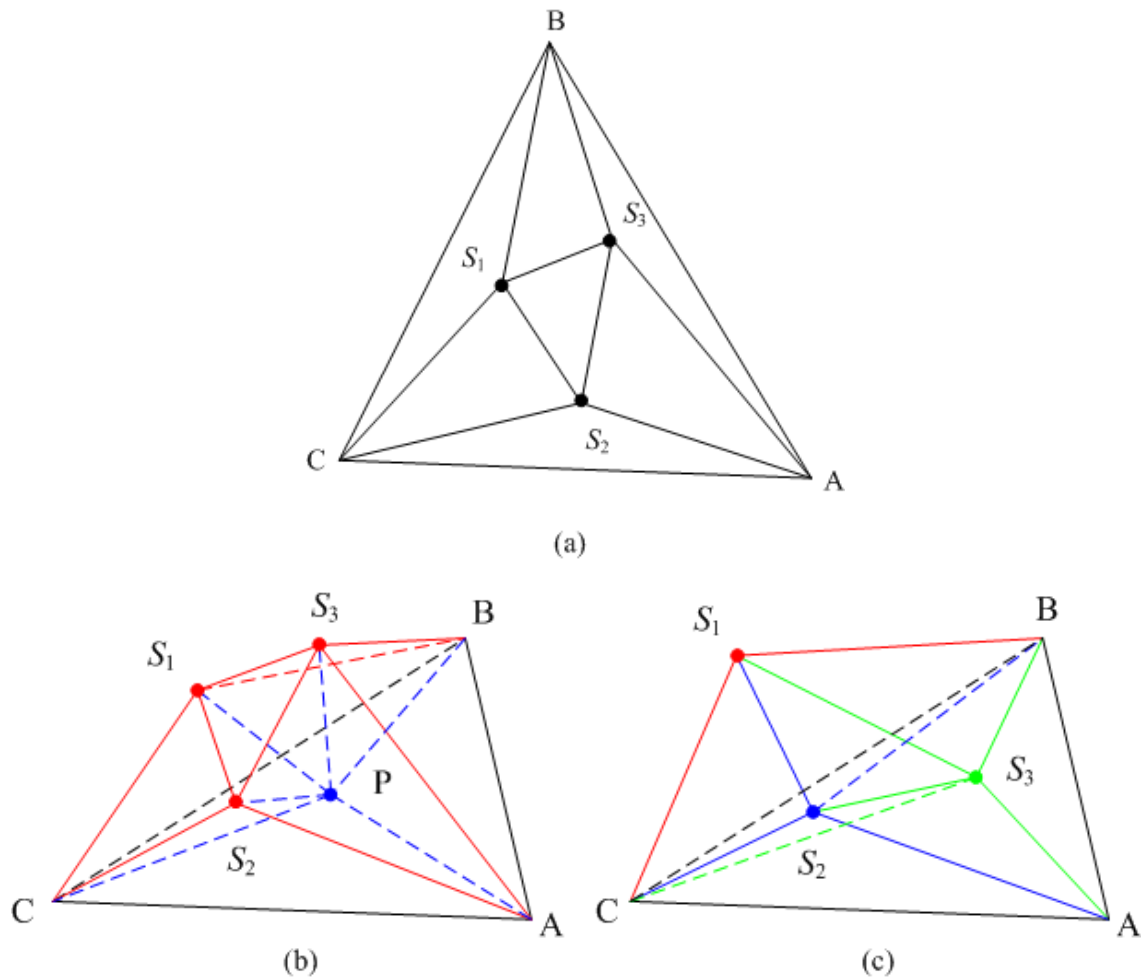


Figure 8. Case I: (a) Boundary facet with 3 Steiner points; (b) Extra point P is introduced; (c) Recovery by the method of George and Du. Colours indicate different steps of node reposition.

Case II: 4 Steiner points

As shown in Figure 9b, an extra Steiner point P is introduced at the barycenter of any three points of S_1 , S_2 , S_3 and S_4 . Elements $PABC$, $PS_1S_2S_3$, $PS_1S_3S_4$, $PS_1S_4S_2$, PAS_2S_3 , PBS_4S_3 , PCS_4S_2 , PS_4BC , PS_2AC and PS_3AB are created to recover the boundary facet ABC , in which P and S_4 are *locked* points. On the other hand, by the method of George and Du, elements $S_1S_2S_3S_4$, $S_2S_3S_4C$, S_2S_3CA , S_3S_4AC , S_3S_4AB and S_4ABC are created, in which three points S_2 , S_3 and S_4 are *locked* as shown in Figure 9c.

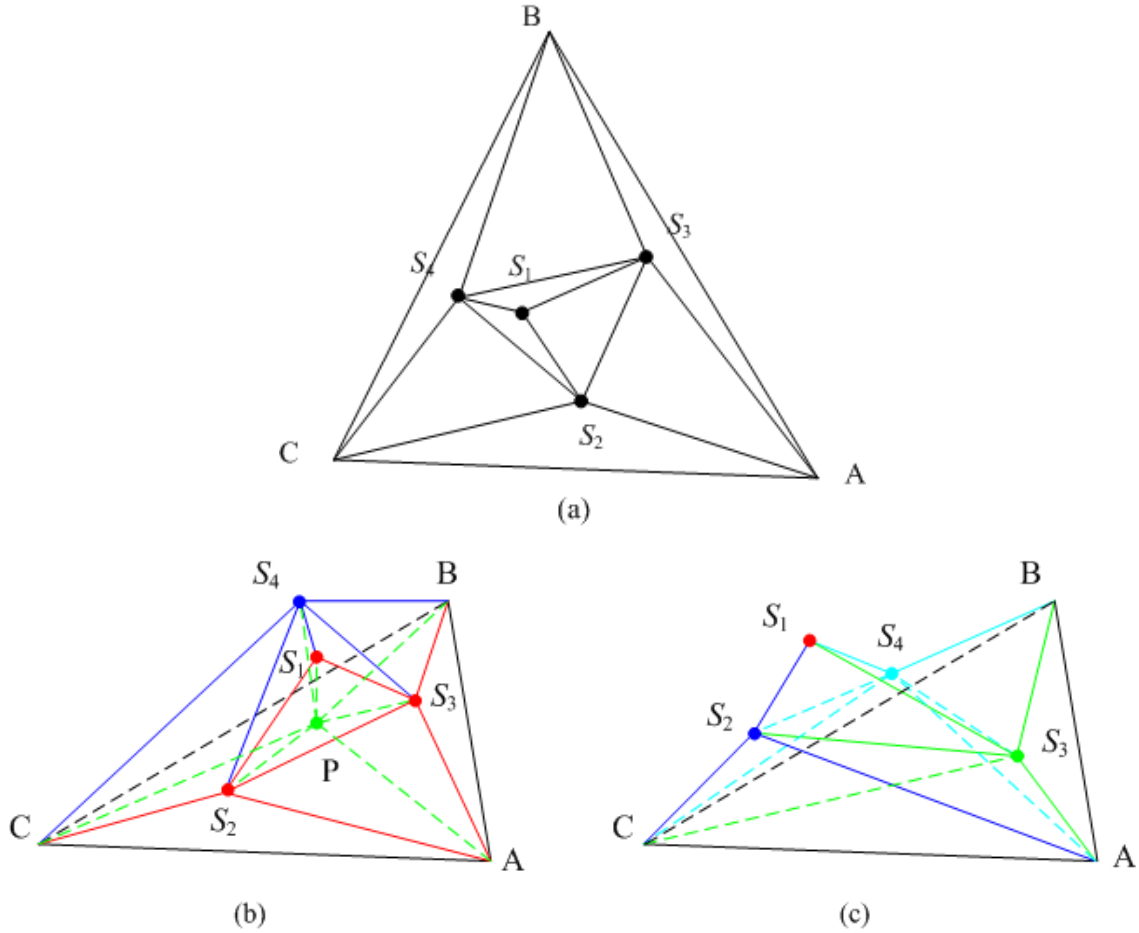


Figure 9: case II: (a) Boundary facet with 4 Steiner points; (b) Extra point P is introduced; (c) Recovery by the method of George and Du. Colours indicate different steps of node reposition.

4. REMOVAL OF FLAT ELEMENTS

In the process of boundary recovery, flat elements of zero volume are created. These elements have to be opened up by a combination of face-swap, reposition (Laplacian-smoothing) and the insertion of extra nodes. Reposition method is an essential part of flat element removal. Among existing reposition approaches [27-30], a reposition method by means of linear programming is adopted, in which the optimal position of a Steiner point could be determined such that all tetrahedra connected to it are ensured to be positive. The method is illustrated with a 2D example.

4.1 2D non-positive element removal

As shown in Figure 10a, $P(x_1, x_2)$ is a point in a 2D triangular mesh with adjacent elements bounded by polygon $V_1V_2V_3V_4V_5$. Let \bar{n}_1 be a normal vector to edge V_1V_2

pointing towards the interior of polygon $V_1V_2V_3V_4V_5$. If $V_1P \cdot \vec{n}_1 > 0$, triangle V_1V_2P is positive, hence $P(x_1, x_2)$ is valid if it satisfies the following conditions:

$$V_iP \cdot \vec{n}_i \geq 0 \quad (i = 1, 2, \dots, m) \quad (4.1)$$

where m is the number of the boundary edges, $\vec{n}_i(n_{ix}, n_{iy})$ is the unit normal vector to edge E_i and V_i is the first vertex of the edge E_i . The constraints can be represented by the following inequalities.

$$(P \cdot \vec{n}_i - s_i)\delta_i = c_i\delta_i \quad (4.2)$$

$$c_i = V_i \cdot \vec{n}_i, \quad \delta_i = \begin{cases} 1 & c_i \geq 0 \\ -1 & c_i < 0 \end{cases}, \quad s_i \geq 0 \text{ are slack variables}$$

It may not be easy to solve the formulation 4.2 directly, but we can convert this problem into a linear programming problem, which can be readily solved by the simplex method. Now, equation (4.2) is transformed to a standard format of linear programming:

$$\begin{cases} \min & f = \sum_{j=1}^m s_j \\ \text{s.t.} & AX = b, \quad x_i \geq 0, \quad s_j \geq 0 \end{cases} \quad (4.3)$$

$$A = \begin{bmatrix} \delta_1 \cdot n_{1x} & \delta_1 \cdot n_{1y} & -\delta_1 & & 0 \\ \vdots & \vdots & & \ddots & \\ \delta_m \cdot n_{mx} & \delta_m \cdot n_{my} & 0 & & -\delta_m \end{bmatrix},$$

$$X = \{x_1 \quad x_2 \quad s_1 \quad \dots \quad s_m\}^T, \quad b = \{c_1 \cdot \delta_1 \quad \dots \quad c_m \cdot \delta_m\}^T$$

As we aim at obtaining the feasible region, the objective function could be an arbitrary linear polynomial of the slack variables. The feasible region exists if $f > 0$, and it degenerates into a line or a point if $f = 0$. The feasible region $v_1v_2v_3v_4v_5$ of formulation (4.3) for a concave pentagon is shown in Figure 10b. As the feasible region is always convex, the barycenter of $v_1v_2v_3v_4v_5$ can be the new position for the relocation of P .

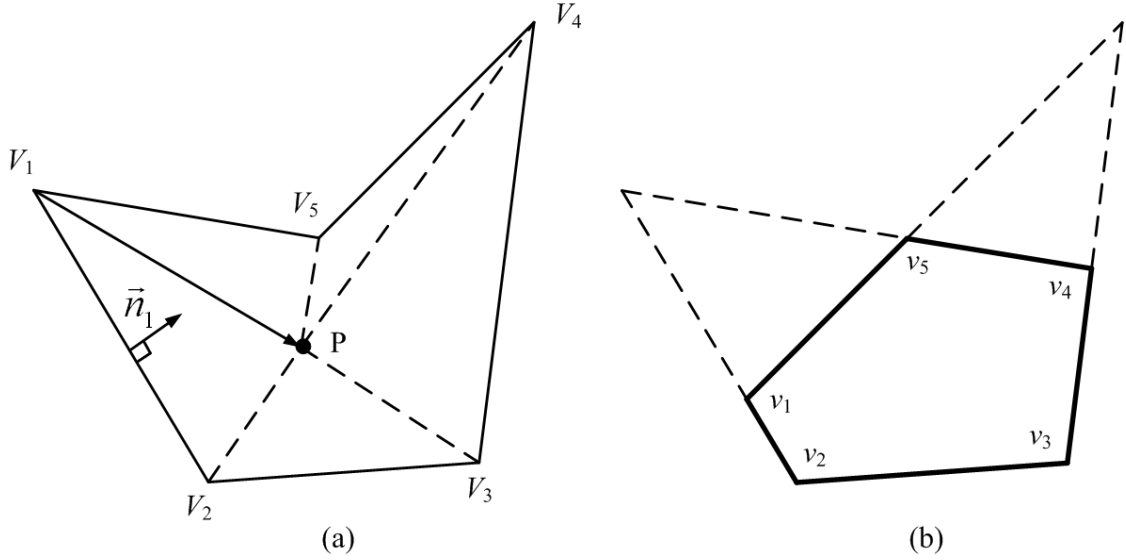


Figure 10. 2D point reposition: (a) a point inside a polygon; (b) the feasible region.

4.2 3D non-positive element removal

In three dimensions, all tetrahedra connected to a point form a polyhedron. As shown in Figure 11a, node P is shared by 8 tetrahedra in the form of a twisted Schönhardt polyhedron. Tetrahedron $ABCP$ is a flat element. The problem of opening up flat element $ABCP$ is equivalent to finding a feasible position for P in the polyhedron $ABCDEF$ as shown in Figure 11b.

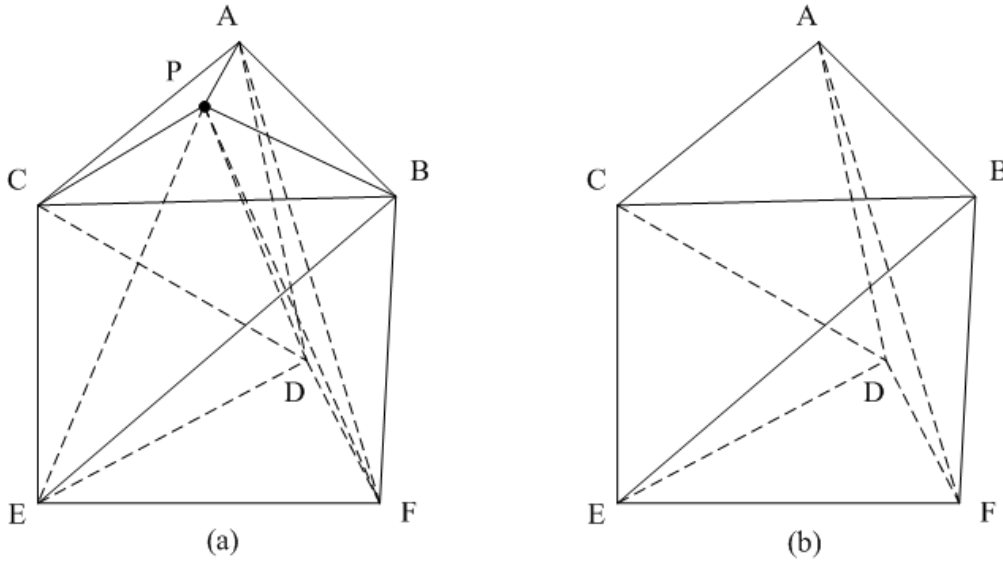


Figure 11. A twisted Schönhardt polyhedron: (a) tetrahedral mesh with flat element $ABCP$; (b) the boundary of the polyhedron.

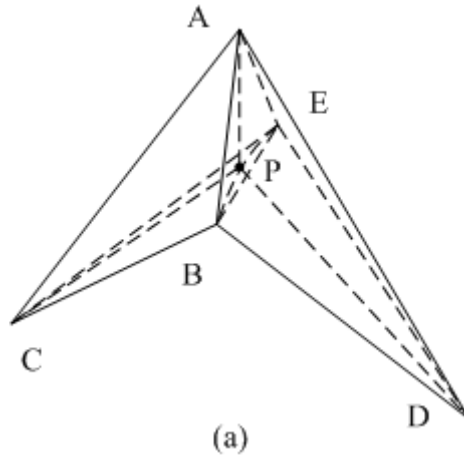
The node reposition problem can be transformed into an optimization problem as in the 2D case. The inequalities in the format of linear programming are given by

$$V_i P \cdot \bar{n}_i \geq 0 \quad (i = 1, 2 \dots m) \quad (4.4)$$

where m is the number of the boundary facets, \vec{n}_i is the unit normal vector to boundary facet F_i and V_i is a vertex on facet F_i . The inequality (4.4) is rewritten in a standard form for linear programming as explained in Section 4.1. In case the feasible region exists, the objective function f is greater than 0, and the barycenter of the feasible region is the new position of P . In fact, the feasible region is given by the bounded region of the extended boundary facets of the given polyhedron.

4.3 Comparison with Laplacian based methods

The drawback of Laplacian based methods is that a valid position for a Steiner point cannot be determined inside a non-convex polygon, whereas the proposed reposition method by means of linear programming could always give the optimal solution should it exist. As shown in Figure 12a, there are six tetrahedra, $PCBA$, $PECA$, $PBDA$, $PDEA$, $CBEP$ and $EBDP$, where $CBEP$ and $EBDP$ are flat elements. To relocate point P , Laplacian based methods move P along PQ outside the polyhedron as shown in Figure 12b. However, the proposed linear programming reposition method finds a valid position Q which is well within the polyhedron as shown in Figure 12c. In the actual implementation, we combine the linear programming reposition method and the smart Laplacian method to relocate Steiner points. Specifically, smart Laplacian method is first applied to determine a new position for a Steiner point, and if this fails, the proposed linear programming reposition method is invoked.



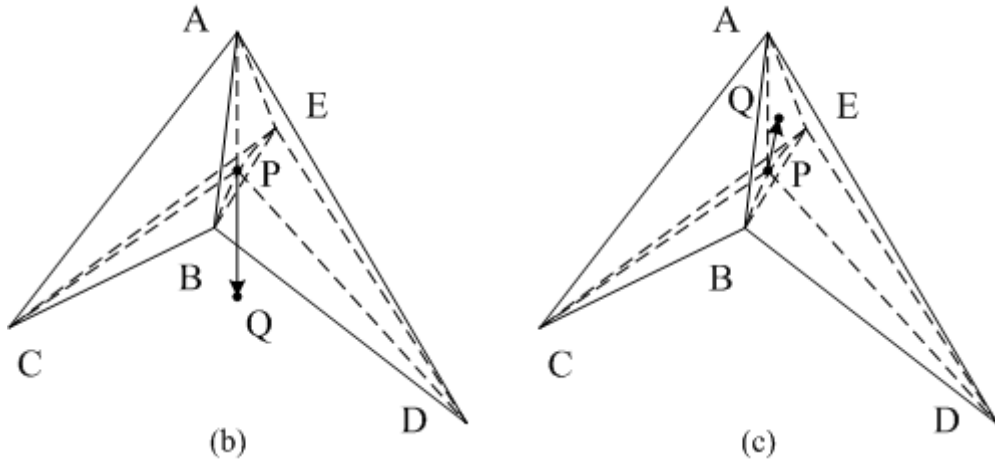
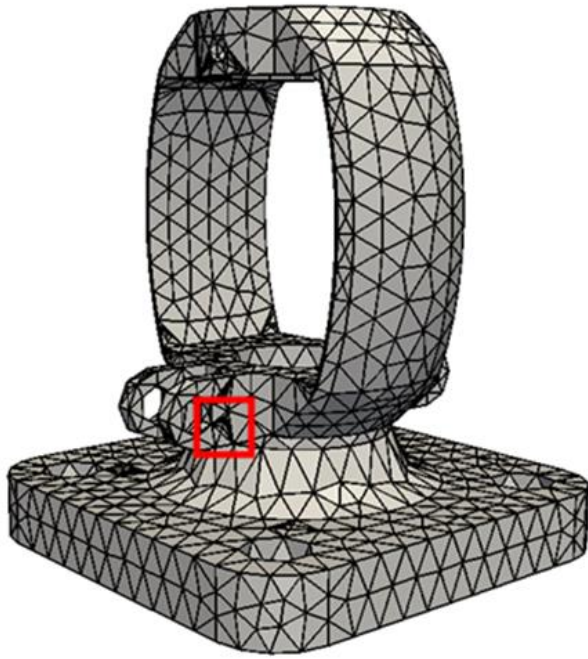


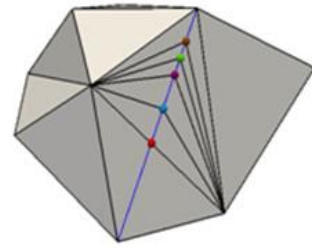
Figure 12: Comparing the proposed reposition method with Laplacian based methods: (a) tetrahedral mesh with flat elements; (b) reposition by the Laplacian based methods; (c) reposition by linear programming.

5. NUMERICAL EXAMPLES

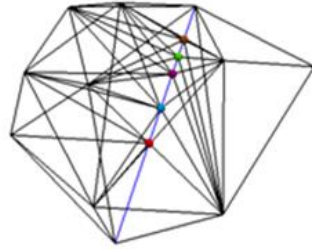
Four application examples are presented to describe some key features of the proposed method in details. The first example is a mechanical support model as shown in Figure 13a. 152 boundary facets were missing in the initial Delaunay triangulation. To recover these missing boundary facets, 50 Steiner points were inserted including 43 points on the original boundary edges. The largest number of Steiner points inserted on a single edge was 5 as shown in Figure 13b. The missing edge is shown in blue color and the 5 Steiner points are marked with different colors. To remove these Steiner points, only two iterations were required following the method proposed in this paper. As shown in Figure 13d, in the first iteration three *unlocked* points are first relocated; then the other two *locked* points are repositioned in the second iteration. On the other hand, random procedure would need 5 iterations to remove all these 5 Steiner points as 4 out of 5 Steiner points were *locked*. As shown in Figure 13e, only one Steiner point could be relocated in each iteration.



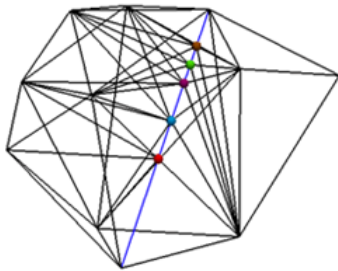
(a)



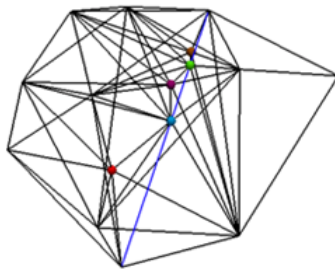
(b)



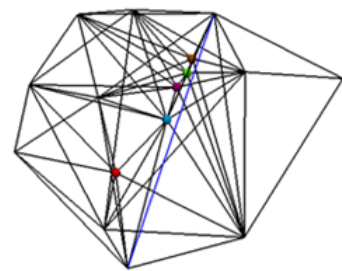
(c)



(Initial)



(1)



(2)

(d)

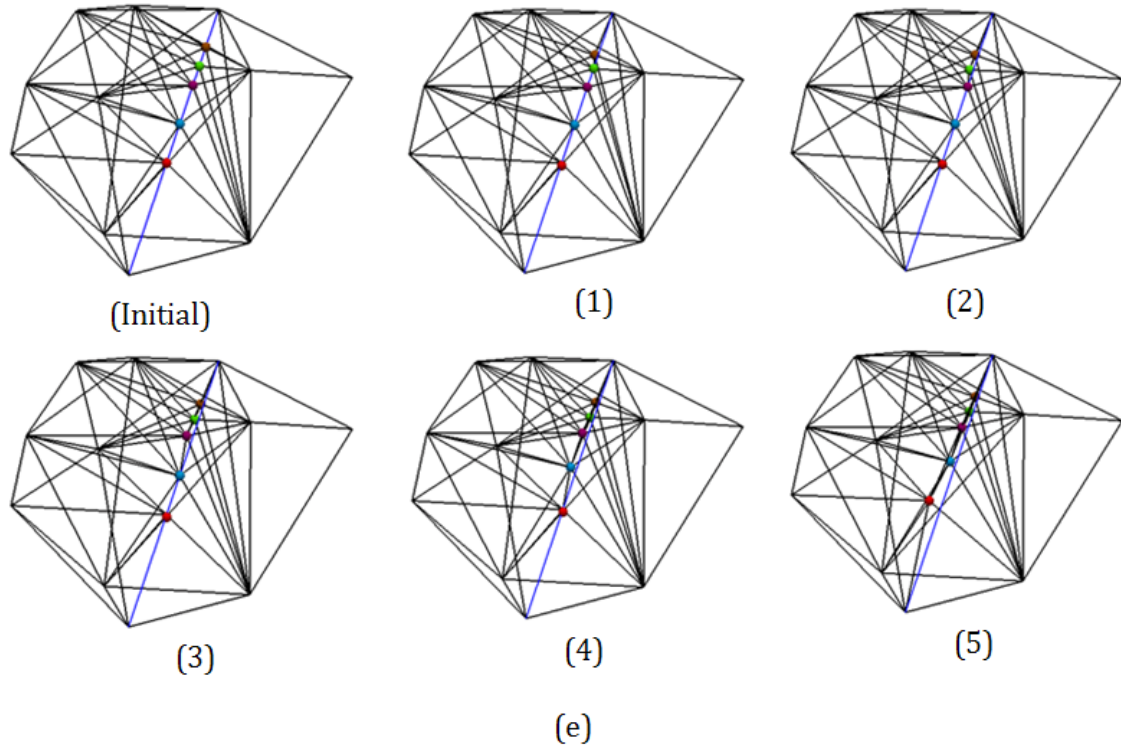
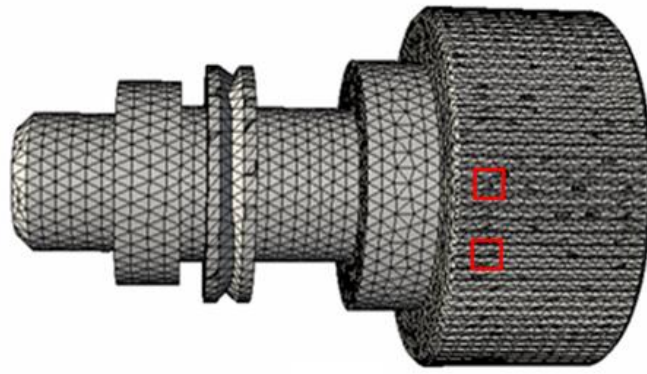


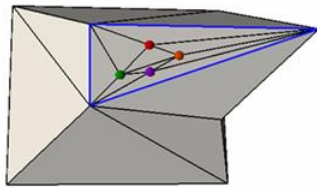
Figure 13. Boundary edge recovery: (a) the support model; (b) and (c) missing edge with 5 Steiner points; (d) Steiner point removal by the proposed method; (e) Steiner point removal by random order

The second example is a screw model as shown in Figure 14a. In this model, 433 Steiner points are introduced including 40 points on the boundary facets, in which the largest number of Steiner points inserted on a single facet is 4. One facet with 4 Steiner points is shown in Figure 14b and 14c, where Figure 14c is the mesh of Figure 14b in a different view. To remove these Steiner points, Figure 14d depicts the process following the proposed suppression order optimization (i.e. higher priority to points with small number of Steiner point neighbors). Compared with the proposed method, the random procedure required more iterations to suppress Steiner points on the missing boundary facet. As shown in Figure 14e, although it is not the worst case (4 iterations), it still needed one more iteration compared to the weighted method.

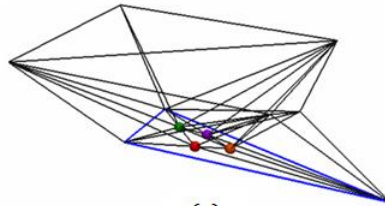
Another case of boundary facet recovery is shown in Figure 14f and 14g. All the 3 Steiner points on the missing facet have the same number of Steiner point neighbors. Figure 14h and 14i depict respectively the procedures of the proposed approach and that of George and Du. Although the proposed approach introduces one more Steiner points, it took only 2 iterations as shown in Figure 14h, which is less than the method of George and Du as shown in Figure 14i.



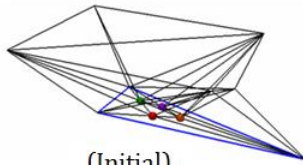
(a)



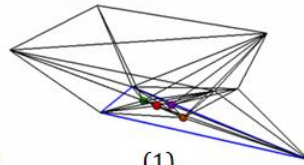
(b)



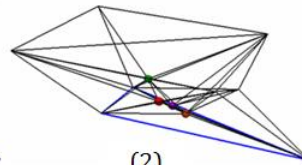
(c)



(Initial)

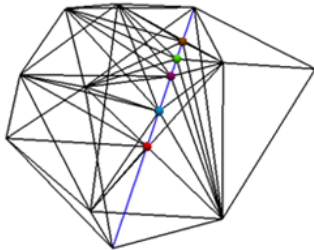


(1)

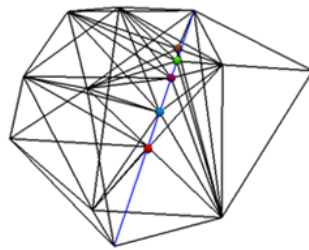


(2)

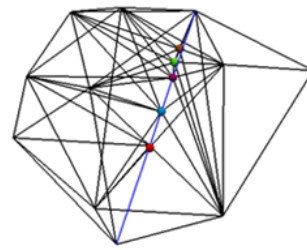
(d)



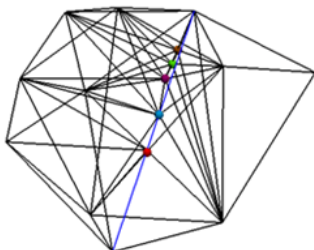
(Initial)



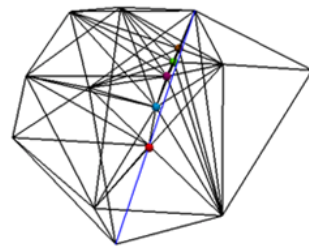
(1)



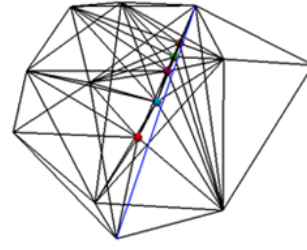
(2)



(3)



(4)



(5)

(e)

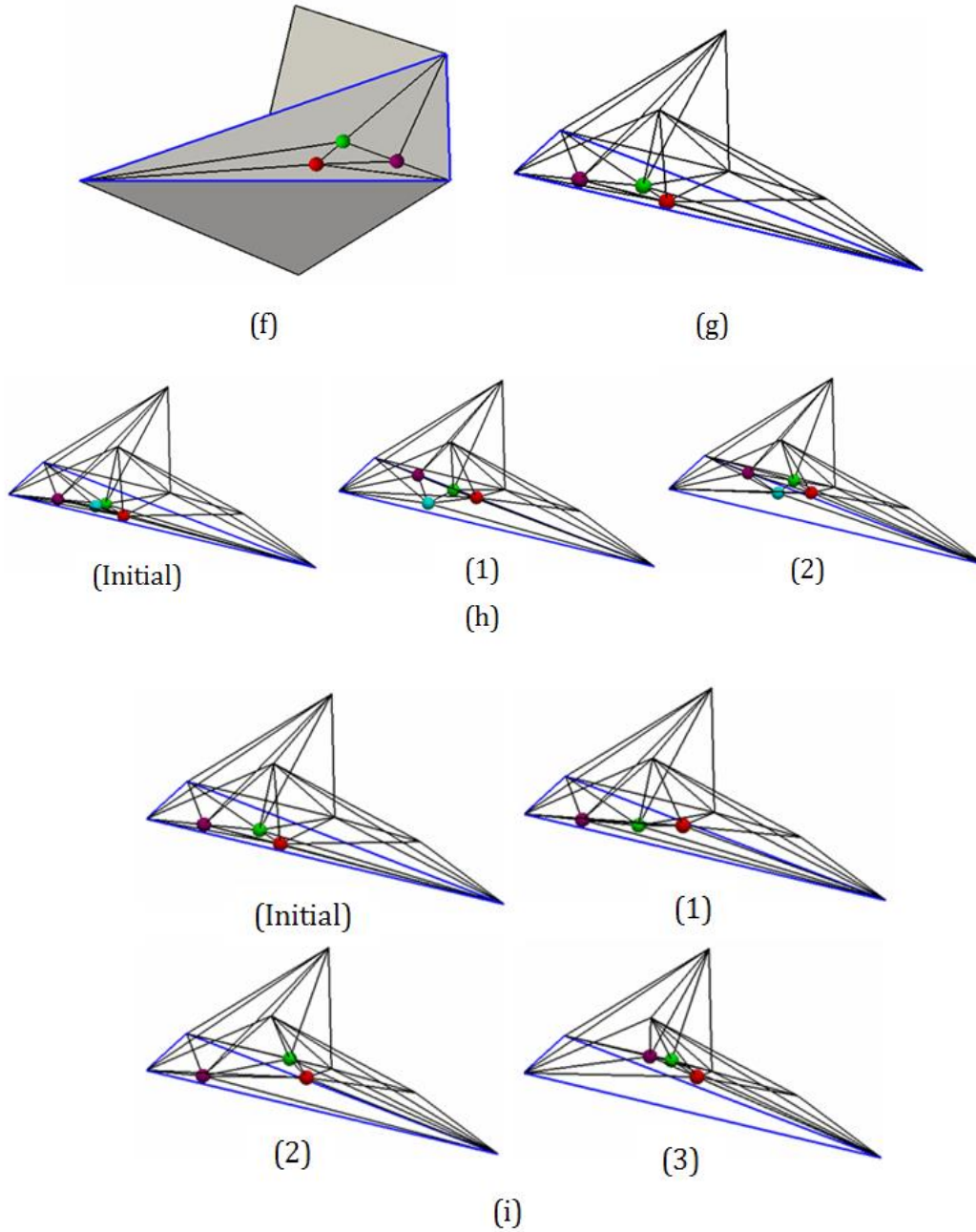


Figure 14. Boundary facet recovery: (a) the screw model; (b) and (c) missing facet with 4 Steiner points; (d) relocating 4 Steiner points by the proposed method; (e) relocating 4 Steiner points by random order; (f) and (g) missing facet with 3 Steiner points; (h) relocating 3 Steiner points by the proposed method; (i) relocating 3 Steiner points by the method of George and Du

The third example is an impeller model with 172 Steiner points inserted as shown in Figure 15a. Point relocation is an essential part in opening up flat elements. Magnified views of a critical recovery zone in the form of a concave polyhedron are shown in Figure 15b and 15c. The node reposition aims at relocating the Steiner point shown in red color in Figure 15b and 15c to remove the flat elements connected to it. The feasible region of the Steiner point is presented as the green polyhedron as shown in Figure 15d. The new position for the Steiner point is at the barycenter of the region, which is

marked red in Figure 15e. Consequently, all flat elements connected to the point are opened up as shown in Figure 15f.

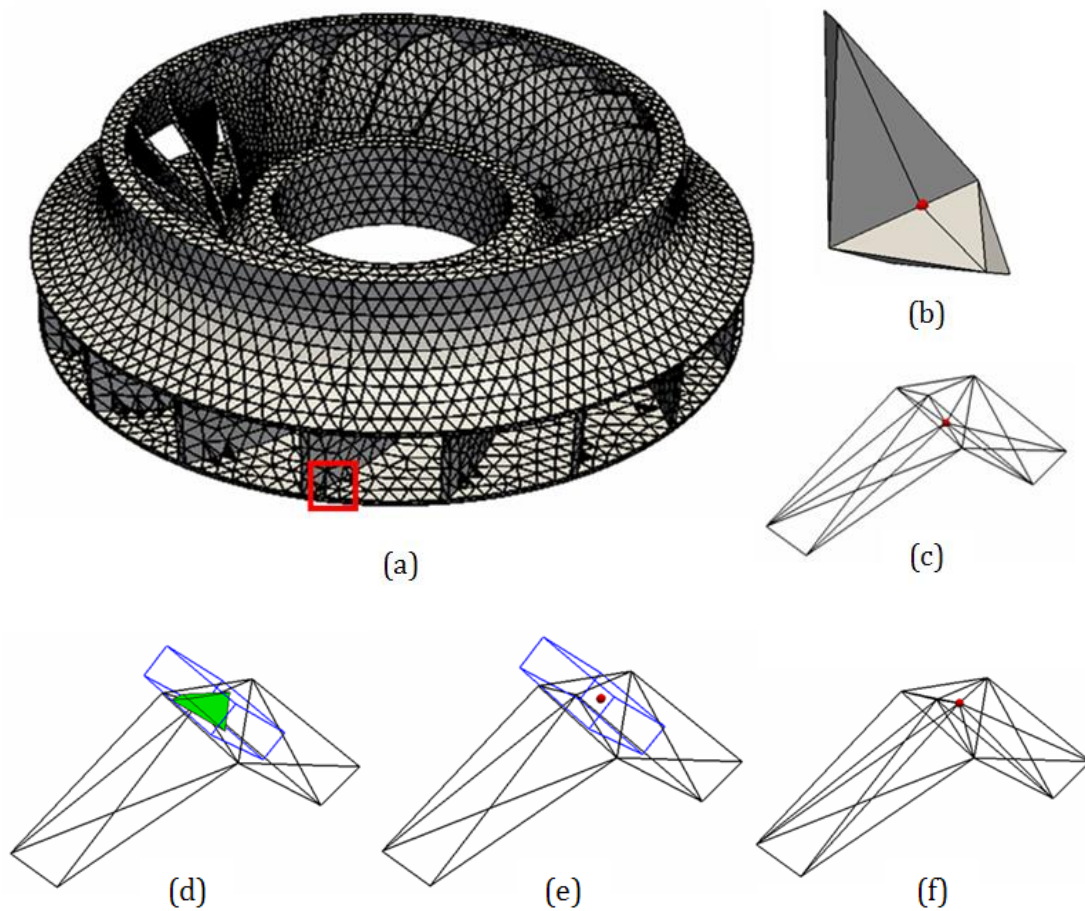


Figure 15. Boundary facet recovery: (a) An impeller model; (b) and (c) concave polyhedron; (d) feasible region and its bounding box; (e) new position of the Steiner point; (f) Steiner point relocated

The boundary recovery approach proposed in this paper has been integrated into our own in-house developed mesh generator. Surface mesh models from the repository <http://www-roc.inria.fr/gamma/gamma.php> are utilized to demonstrate the effectiveness of the proposed approach. The falcon model as shown in Figure 16a is reported here to discuss the main features of our tetrahedral mesh generator. There are 141470 triangles on the boundary, including 19738 (14%) sharp triangles with an angle less than 5° . The result of mesh generation is shown in Figure 16b. To compare with existing tetrahedral mesh generators, ANSYS13.0, HyperMesh10.0 and HyperMesh11.0 were applied to mesh the falcon model. If Steiner points were permitted to stay on the boundary, all three mesh generators succeeded in producing a valid tetrahedral mesh; otherwise, only HyperMesh11.0 was able to produce a valid mesh, and ANSYS13.0 and HyperMesh10.0 crashed in the process of boundary recovery. Comparing the tetrahedral mesh generated by the proposed method with the one by HyperMesh11.0, the smallest dihedral angle in the mesh by the new approach is 2.09° , whereas in the mesh of HyperMesh11.0 it is 0.513° . Considering the distribution of dihedral angles from 0° to

3° with an increment interval of 0.3°, there are much fewer sharp angles in the mesh generated by the proposed recovery procedure compared to the mesh of HyperMesh11.0 as shown in Table 1 and Figure 16c.

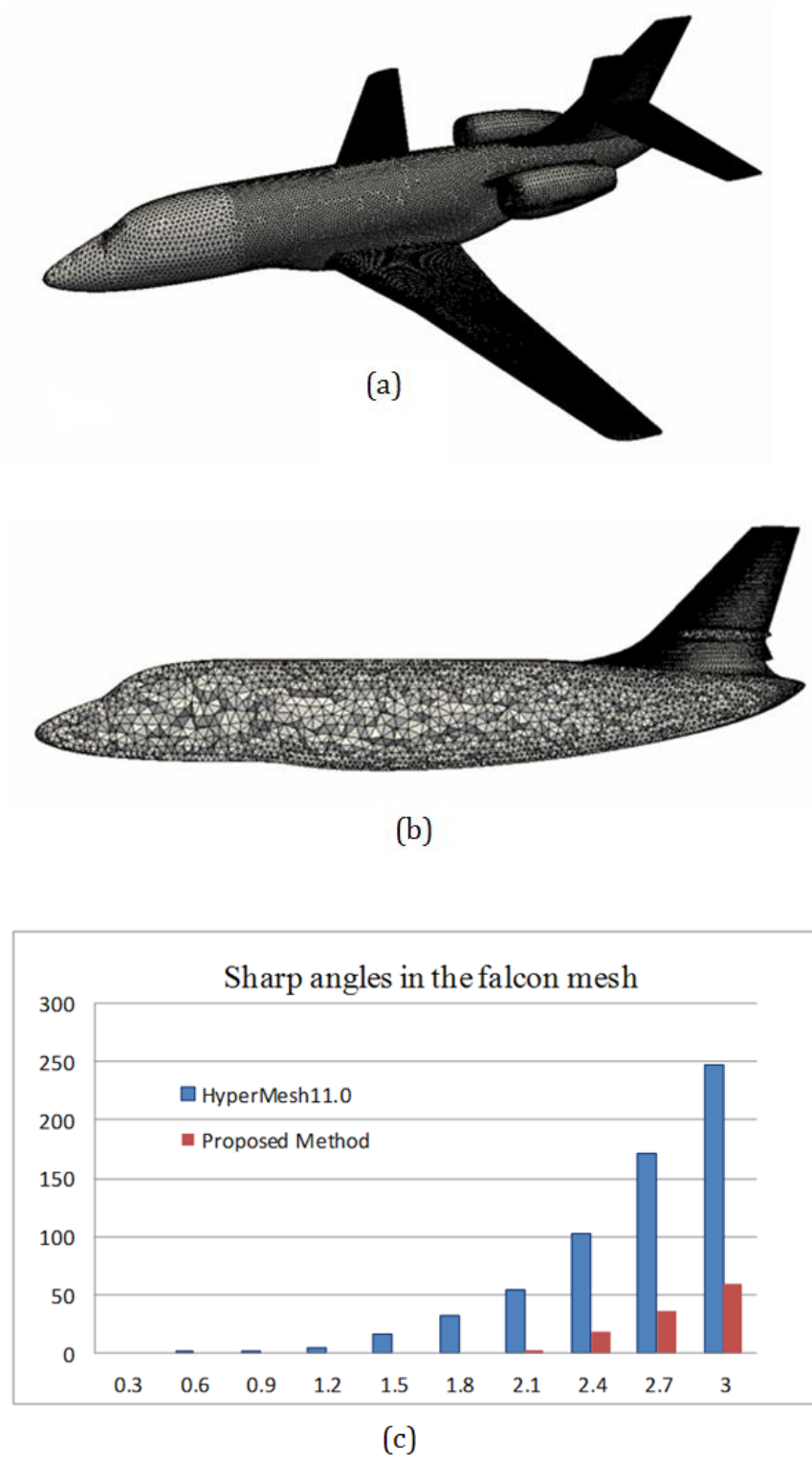


Figure 16. (a) A falcon model; (b) tetrahedral mesh generated by our mesh generator; (c) sharp angles in the falcon mesh

Table1. Sharp angles in falcon

Angle (degree)	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3.0
Hyper Mesh 11.0	0	1	2	5	16	32	54	102	171	248
Proposed Method	0	0	0	0	0	0	2	19	36	59

Four more industrial examples are included to demonstrate the capability and the characteristics of the proposed boundary recovery procedure in dealing with complex practical applications. The general description and the statistics in the boundary recovery process of the model are listed in Table 2. For the quoted examples, in each case, there are about one thousand missing edges and a couple of thousands of missing faces; however, the number of Steiner points needed to assist the edge and face recovery are relative few, showing that most of the missing quantities can already be recovered by some topological operations involving the swap of element edges and faces. The seahorse model is the largest example with 173160 boundary faces meshed in 833309 tetrahedral elements. However, it is not the most difficult example as the shape qualities of the boundary triangles are not that bad, even though 11 Steiner points are needed in the recovery of one boundary face. The tyre and engine models can be considered as more difficult as more Steiner points are required for the edge and face recovery. The minimum γ -quality of the tetrahedral elements are quite low, probably due to the poor boundary facets and more Steiner points are used in recovering the boundary faces.

Table 2. Statistics of the boundary recovery process for industrial applications

Model	Seahorse	Fish	Tyre	Engine
Figure	17	18	19	20
<i>Model description</i>				
Number of boundary faces	173160	93054	99198	25502
Number of nodal points	86576	46519	49581	12703
Edge aspect ratio (worst/average)	0.192/0.697	0.270/0.721	0.012/0.733	0.022/0.580
Radius aspect ratio (worst/average)	0.069/0.857	0.176/0.878	2.08e-4/0.892	1.39e-4/0.712
Minimum angle	8.24	15.18	0.28	0.36
<i>Statistics of the boundary recovery process</i>				
Number of missing edges	2161	897	1963	2611
Number of missing faces	4226	1798	3835	4168
Number of Steiner points on edges	26	11	40	306
Max. Steiner points on one edge	5	4	4	4
Number of Steiner points on faces	62	45	1975	875
Max. Steiner points on one face	11	8	5	5
<i>Characteristics of output tetrahedral mesh</i>				

Number of tetrahedral elements	833309	517766	463147	51766
Number of nodal points	184611	111161	98599	14670
γ -quality (worst/average)	0.064/0.776	0.064/0.809	3.85e-5/0.738	3.4e-4/0.477
Edge aspect ratio (worst/average)	0.139/0.629	0.118/0.657	0.008/0.620	0.002/0.445
Radius aspect ratio (worst/average)	0.038/0.805	0.037/0.836	2.54e-4/0.770	2.04e-4/0.508
Dihedral angle (minimum/average)	10.03/47.33	5.42/49.23	0.27/44.71	0.24/32.43

Key: $Radius\ aspect\ ratio = Inradius/circumradius$

$$\gamma - quality\ of\ tetrahedron = \frac{72\sqrt{3}\ volume\ of\ tetrahedron}{(sum\ of\ squares\ of\ edges)^{3/2}}$$

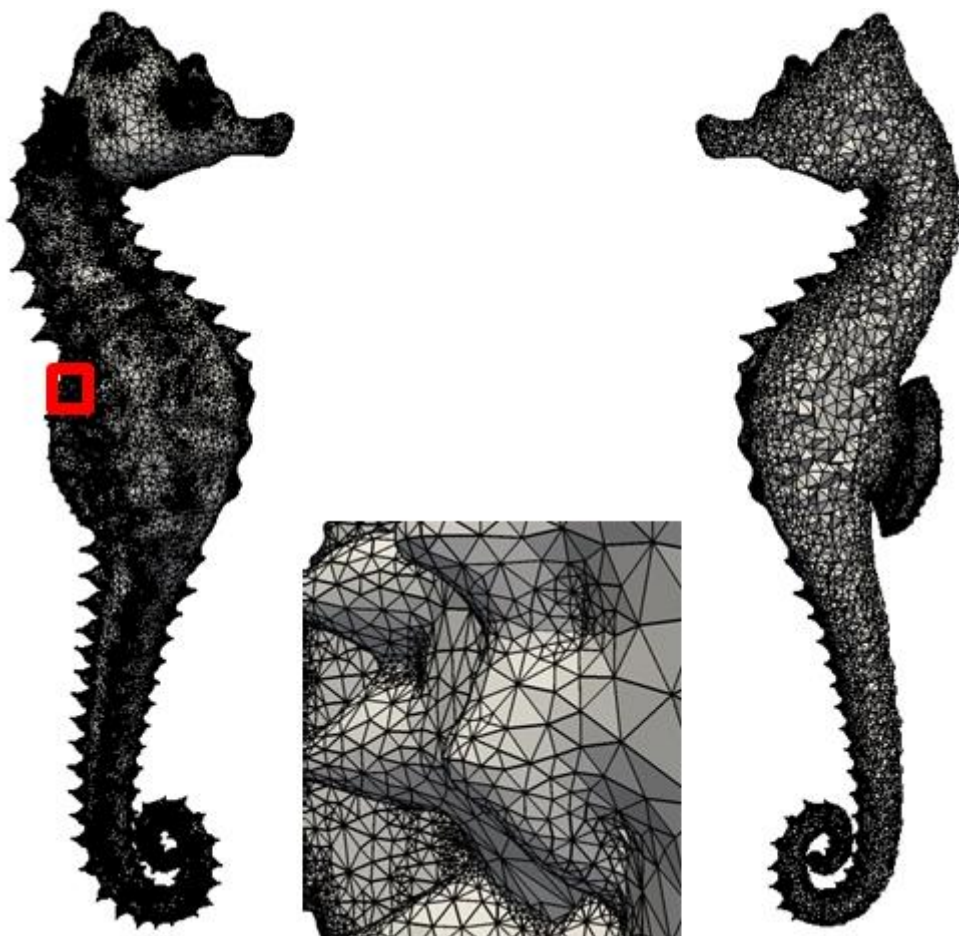


Figure 17. A seahorse of 173160 boundary triangles meshed in 833309 tetrahedra

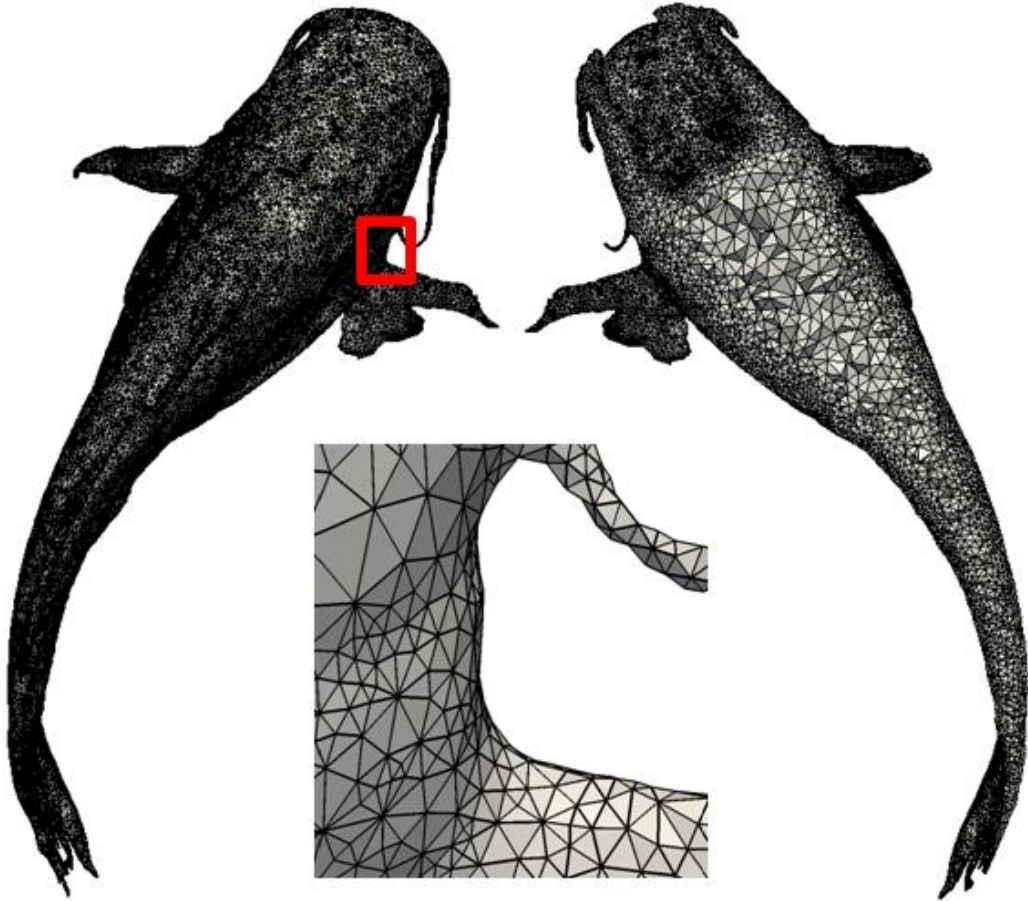


Figure 18. Fish of 93054 boundary triangles meshed in 517766 tetrahedra

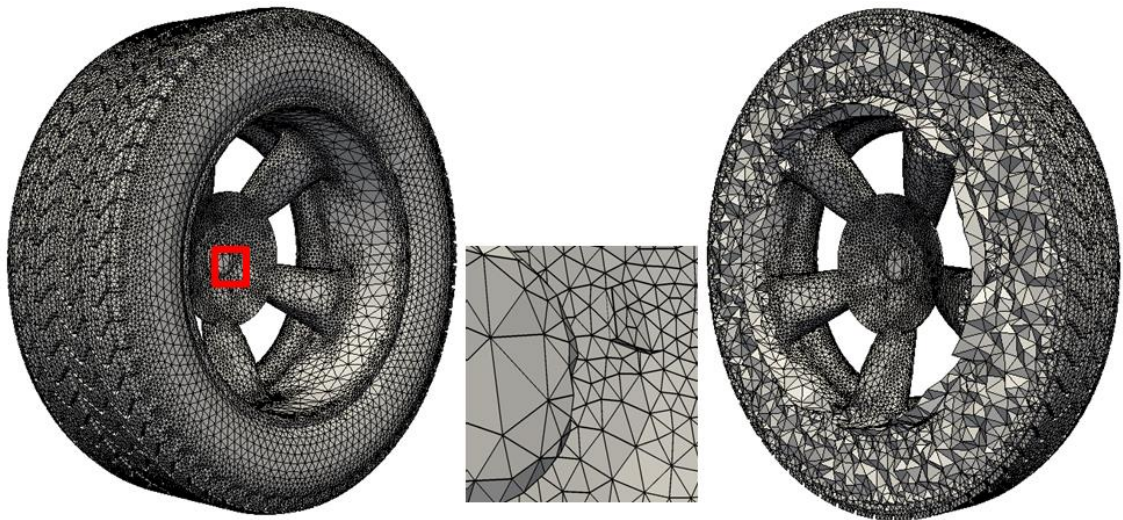


Figure 19. A tyre of 99198 boundary triangles meshed in 463147 tetrahedra

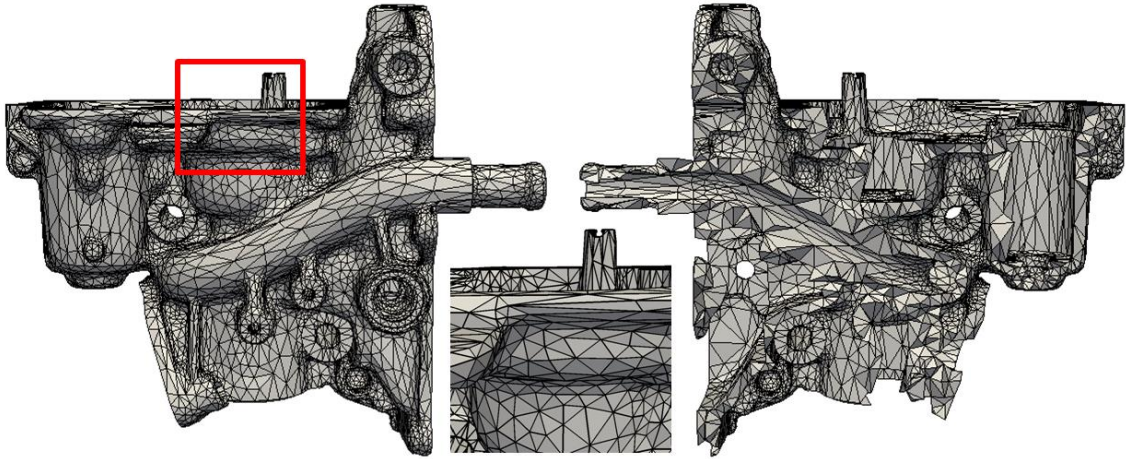


Figure 20. An engine of 25502 boundary triangles meshed in 51766 tetrahedra

6. CONCLUSIONS AND DISCUSSIONS

New ideas for the constrained boundary recovery of 3D Delaunay triangulation are presented to address some of the critical issues in finite element mesh generation. The notion of *locked* points has been introduced to elucidate why the repositioning and the removal of flat elements are sometimes very difficult. The sequence in removing the Steiner points on an edge has been carefully reviewed so as to minimize the number of *locked* points in the recovery process. As for the removal of Steiner points on a triangular facet, the order of removal is determined by a weight related to the number of neighboring Steiner points, which in general is superior to a random selection process commonly adopted. A linear programming technique is also employed to determine the feasible region, at the center of which a Steiner point can be relocated to ensure that all tetrahedral elements connected to it are positive. Work examples are included to show the details of the boundary recovery procedure, along with several practical applications adopted from the industry with difficult boundary conditions. It is founded that the method proposed in this paper can handle complicated industrial finite element models and could produce better results than popular commercial software available in the market in terms of mesh quality.

ACKNOWLEDGEMENT

The financial support from HKSAR GRF Grant to the research project HKU715110E on “Drifted based seismic fragility analysis of high-rise RC building with transfer structures” is greatly acknowledged. The authors would also like to appreciate the joint supports to this project by the National Natural Science Foundation of China (Grant No. 10872040, 10572032 and 10721062).

REFERENCES

- [1] N. Chentanez, B.E. Feldman, F. Labelle, J.F. O'Brien, J.R. Shewchuk, Liquid simulation on lattice-based tetrahedral meshes, in: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, Eurographics Association, 2007, pp. 219-228.
- [2] H.W. Zhang, H. Wang, J.B. Wang, Parametric variational principle based elastic-plastic analysis of materials with polygonal and Voronoi cell finite element methods, *Finite Elem. Anal. Des.* 43 (2007) 206-217.
- [3] T. Grosge, H. Borouchaki, D. Barchiesi, New adaptive mesh development for accurate near - field enhancement computation, *J. Microsc.* 229 (2008) 293-301.
- [4] H.W. Zhang, H. Wang, B.S. Chen, Z.Q. Xie, Analysis of Cosserat materials with Voronoi cell finite element method and parametric variational principle, *Comput. Methods Appl. Mech. Eng.* 197 (2008) 741-755.
- [5] R. Löhner, P. Parikh, Generation of three-dimensional unstructured grids by the advancing-front method, *Int. J. Numer. Meth. Fluids* 8 (1988) 1135-1149.
- [6] A. Rassineux, Generation and optimization of tetrahedral meshes by advancing front technique, *Int. J. Numer. Methods Eng.* 41 (1998) 651-674.
- [7] M. Shephard, M. Georges, Automatic three-dimensional mesh generation by the finite octree technique, *Int. J. Numer. Methods Eng.* 32 (1991) 709-749.
- [8] J.R. Shewchuk, Delaunay refinement mesh generation. Carnegie Mellon University, 1997.
- [9] P.J. Frey, H. Borouchaki, P.L. George, 3D Delaunay mesh generation coupled with an advancing-front approach, *Comput. Methods Appl. Mech. Eng.* 157 (1998) 115-131.
- [10] P.L. George, H. Borouchaki, Delaunay triangulation and meshing: application to finite elements, Kogan Page, 1998.
- [11] Q. Du, D. Wang, Recent progress in robust and quality Delaunay mesh generation, *J. Comput. Appl. Math.* 195 (2006) 8-23.
- [12] H. Si, Constrained Delaunay tetrahedral mesh generation and refinement, *Finite Elem. Anal. Des.* 46 (2010) 33-46.
- [13] P.L. George, H. Borouchaki, E. Saltel, 'Ultimate' robustness in meshing an arbitrary polyhedron, *Int. J. Numer. Methods Eng.* 58 (2003) 1061-1089.
- [14] Q. Du, D. Wang, Constrained boundary recovery for three dimensional Delaunay triangulations, *Int. J. Numer. Methods Eng.* 61 (2004) 1471-1500.
- [15] H. Si, K. Gaertner, Meshing piecewise linear complexes by constrained Delaunay tetrahedralizations, in: Proceedings of the 14th International Meshing Roundtable, Springer-Verlag New York Inc, 2005, pp. 147-163.
- [16] J. Liu, B. Chen, Y. Chen, Boundary recovery after 3D Delaunay tetrahedralization without adding extra nodes, *Int. J. Numer. Methods Eng.* 72 (2007) 744-756.
- [17] H. Ghadyani, J. Sullivan, Z. Wu, Boundary recovery for Delaunay tetrahedral meshes using local topological transformations, *Finite Elem. Anal. Des.* 46 (2010) 74-83.

- [18] N.P. Weatherill, O. Hassan, Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints, *Int. J. Numer. Methods Eng.* 37 (1994) 2005-2040.
- [19] P.L. George, F. Hecht, E. Saltel, Automatic mesh generator with specified boundary, *Comput. Methods Appl. Mech. Eng.* 92 (1991) 269-288.
- [20] Q. Du, D. Wang, Boundary recovery for three dimensional conforming Delaunay triangulation, *Comput. Methods Appl. Mech. Eng.* 193 (2004) 2547-2563.
- [21] J. Chen, D. Zhao, Z. Huang, Y. Zheng, S. Gao, Three-dimensional constrained boundary recovery with an enhanced Steiner point suppression procedure, *Comput. Struct.* 89 (2011) 455-466.
- [22] Z. Guan, C. Song, Y. Gu, The boundary recovery and sliver elimination algorithms of three-dimensional constrained Delaunay triangulation, *Int. J. Numer. Methods Eng.* 68 (2006) 192-209.
- [23] B. Delaunay, Sur la sphere vide, *Bulletin of Academy of Sciences of the USSR* 6 (1934) 793-800.
- [24] A. Bowyer, Computing Dirichlet tessellations, *Comput. J.* 24 (1981) 162-166.
- [25] D.F. Watson, Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes, *Comput. J.* 24 (1981) 167-172.
- [26] S.H. Lo, Optimization of tetrahedral meshes based on element shape measures, *Comput. Struct.* 63 (1997) 951-961.
- [27] N. Amenta, M. Bern, D. Eppstein, Optimal point placement for mesh smoothing, in: *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, New Orleans, Louisiana, United States, 1997, pp. 528-537.
- [28] L.A. Freitag, C. Ollivier-Gooch, Tetrahedral mesh improvement using swapping and smoothing, *Int. J. Numer. Methods Eng.* 40 (1997) 3979-4002.
- [29] B. Klingner, J. Shewchuk, Aggressive Tetrahedral Mesh Improvement, in: *Proceedings of the 16th International Meshing Roundtable*, 2008, pp. 3-23.
- [30] P. Alliez, D. Cohen-Steiner, M. Yvinec, M. Desbrun, Variational tetrahedral meshing, *ACM Trans. Graph.* 24 (2005) 617-625.