International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.8 (2015), pp.107-114
http://dx.doi.org/10.14257/ijmue.2015.10.8.11

Two-Dimensional Boundary Recovery Procedure for Non-Manifold and Intersecting Boundaries

Chaoyan Zhu, Zhoufang Xiao and Lijuang Zeng

Center for Engineering and Scientific Computation, Zhejiang University, China；
Center for Engineering and Scientific Computation, Zhejiang University, China
Ningbo Institute of Technology, Zhejiang University, China
zcy@nit.zju.edu.cn, {xiaozf, zenglijuan}@zju.edu.cn

Abstract

To ensure all boundary edges of each region are retained in the final mesh during mesh generation, a procedure suitable for non-manifold with intersecting boundaries is proposed to recover the non-manifold and intersecting boundaries. In the procedure, Steiner points are inserted directly on the interFigure of missing boundaries and mesh edges from a triangulation. Based on the boundary recovery procedure, Delaunay triangulation algorithms for the domain with interior constraints and for the non-manifold are proposed, and the latter can be used as a new effective polygon Boolean operations algorithm.

Keywords： *Mesh generation; Boundary recovery; Delaunay triangulation; Interior constraints; Boolean operation*

1. Introduction

Mesh generation is the pre-processing step and the major performance bottleneck of finite element analysis [1]. Delaunay triangulation method is a mature and widely used method to generate mesh, but it cannot guarantee each border (edge or face) retained in the final mesh. while the borders are usually used to define the problem domain to distinguish exterior from interior, furthermore, some interior constraint borders are contained in practical engineering models to either represent multi-material interface or characterize the area where physical quantities change drastically (such as cracks, *etc.*)[2-4]. so during the procedure of Delaunay triangulation, boundary recovery should be done after inserting all the points.

Diagonal swapping is one kind of boundary recovery methods of Delaunay triangulation, usually used for manifold, but it is also suitable for non-manifold if auxiliary data structures and well-designed swapping strategies are adopted [5]. However, diagonal swapping can't deal with the case of boundaries intersecting because the interFigure points of the boundaries should be kept while any new points wouldn't be introduced in diagonal swapping. A natural solution to it is to calculate the intersecting point of the boundaries in advance before boundary recovery, and then split the old boundaries at the intersecting point to get new non-intersecting boundaries. But it is not easy to implement if without extra auxiliary data structures, is difficult to calculate precisely, and hard to extend to three dimensions.

In this article, we present a new boundary recovery algorithm in which Steiner points are directly inserted at the intersecting point of the missing boundaries and mesh edges from a triangulation to adapt to the cases of non-manifold with intersecting boundaries. The Steiner points inserted are classified according to the intersecting position, one kind is the intersecting point of two boundaries which should be kept in the final mesh, while the

ISSN: 1975-0080 IJMUE

other is the intersecting point of boundary edge and non-boundary edge which should be removed.

Compared to diagonal swapping, the method in the article apparently has more advantages when dealing with the problems with intersecting boundaries. First, points are inserted during boundary recovery procedure without any extra preprocess operation[6, 7]; Second, the method can be directly extended to three dimension while diagonal swapping can't [8, 9]; Third, the method has a wide range of applications, it can be used for the domain with interior constraints and for the non-manifold, and used to do effective polygon Boolean operations.

The remainders of the paper are organized as follows: after introducing the new boundary recovery algorithm in Figure 2, we discuss the detailed process of Delaunay triangulation with constrained boundaries in Figure 3. In Figure 4 we present the Delaunay triangulation algorithm suitable for non-manifold which can be used as a universal effective Boolean operation algorithm. In Figure 5 we provide some experiments to see the efficiency and correctness of the algorithms.

2. Boundary Recovery Algorithm

Assum $P=\{p_k\}$ is a point set in R^2 , $T_i=\{t_k\}$ are mesh cells from the triangulation defined on P. All cells in T_i Cover a convex polygon area Ω without overlapping each other, and the edge e connecting any two points in P should be contained in Ω. If e is not a triangle edge of any triangle in T_i, then e is called the missing boundary. The boundary recovery procedure starts from the cells in T_i, gets the new triangle T_{i+1} including the missing boundary and T_{i+1} will not overlap with other triangle cells and be contained in Ω.

In the proposed boundary recovery algorithm which can be extended to deal boundary interFigure in three dimensions, Steiner points are inserted on the interFigure of the missing boundary and triangle cell edges. Inserting points and diagonal swapping are both applied. Steiner points are classified into two categories, one category refers to the points of the interFigure of two boundaries which should be remained in the mesh, such as the point H in Figure 1(b), the other category refers to the points of the interFigure of the boundary edge and non-boundary edge which should be removed from the mesh, such as the point G in Figure 1(b). The recovery algorithm includes two main steps for each lost boundary edge:

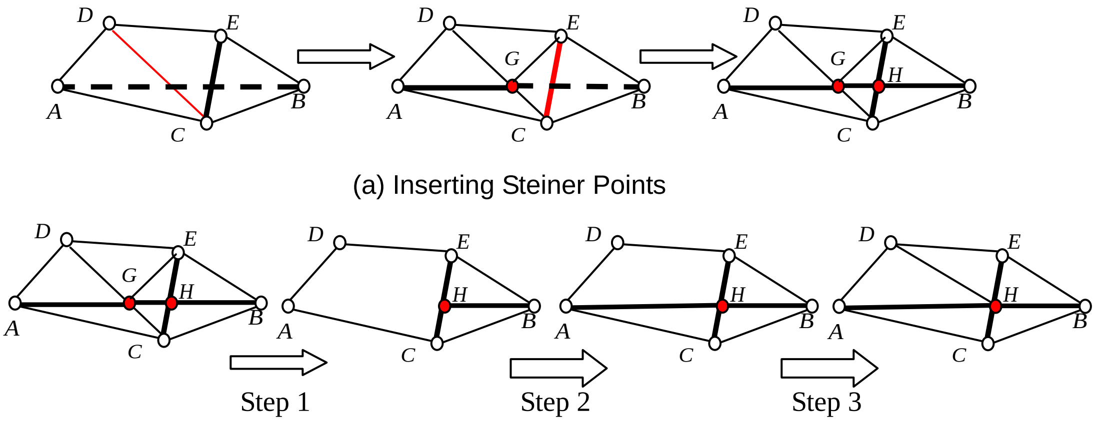
Step1: Insert Steiner points. As shown in Figure 1(a), insert Steiner point at interFigure of the lost boundaries and mesh edges.

Step2: Delete Steiner points. If the Steiner points are located on the interFigure of boundary edge and non-boundary edge, the Steiner points should are deleted, and then diagonal swapping should be carried out to recover the missing part of the border boundary. Figure 1(b) shows the process before and after the deleting of the Steiner point G. There are three detailed steps during the procedure.

1) Delete all the triangle cells which adjacent to the Steiner point, and form a convex polygon cavity.

2) Split the cavity into two polygons with the boundary.

3) Re-mesh the two polygons.



(a) Inserting Steiner Points

(b) Deleting Steiner Points

Figure 1. The Two Major Steps in the Boundary Recovery Algorithm (AB and CE are the Boundary Edges)

3. Delaunay Triangulation Algorithms for Domain with Constraints

Some practical engineering models will contain constraint boundaries to characterize either the multi-material interfaces, or the area where physical quantities change drastically (such as cracks, *etc.*), and sometimes the constraint boundaries may self-intersect or intersect with outer boundaries. These constraint boundaries usually should be kept in the final mesh, but most mesh generation algorithms can't deal the situation directly.

The article presents a mesh generation algorithm making use of the boundary recovery algorithm introduced in Figure 2, as shown in algorithm 1 which can deal with the models with constraint boundaries automatically. The border of the problem domain is described by CAD curves, and each curve is marked as **boundary** or **constrained**. The **boundary** curves define the entire problem domain to distinguish between interior and exterior, while the **constrained** curves are referred to the interior constraint lines. Just as described in algorithm 1, the first step is constructing an initial triangulated rectangle covering the whole problem domain, as shown in Figure 2(a). Second, the boundary and constrained curves are discretized as segments according to the predefined size field, and those discretized points are inserted using the Bowyer-Watson point insertion algorithm [10], as shown in Figure 2(b). The third step is inserting interior nodes using Bowyer-Watson point insertion algorithm under the special size filed and recovering the missing boundaries with the algorithm described in Figure 2, as demonstrated by Figure 2(c).

The next step is deleting the outer elements which either cover out of the problem domain or in the holes of the domain. The details are as follows:

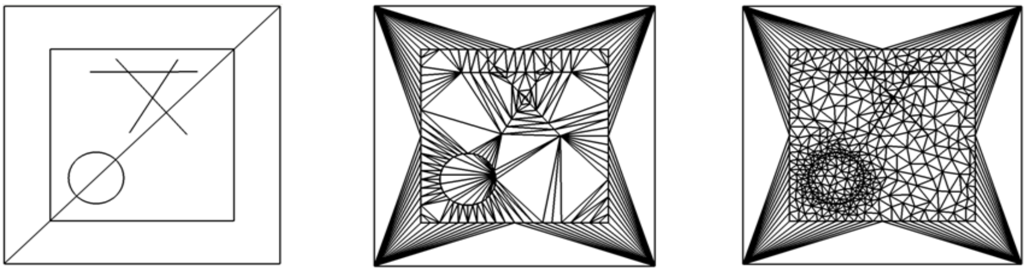
Step 1. The elements adjacent to any node of the initial triangulated rectangle are marked as outer elements.

Step 2. Mark the elements as interior or exterior respectively using flood fill algorithm. Let the marked elements be seeds, the elements adjacent to them will be marked according to the following two rules:

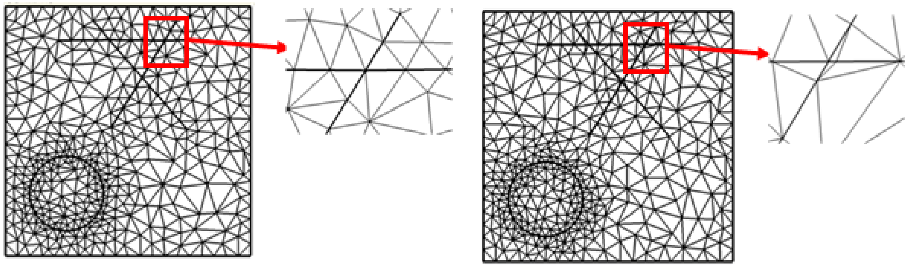
Rule 1: Between two elements sharing any boundary borders, one is outer element, the other is on the contrary.

Rule 2: Between two elements sharing any none-boundary borders, they're both outer elements, or both inner elements.

Boundary borders refer to those mesh edges obtained by boundary curves discretization, and none-boundary borders are other mesh edges, including the edges discretized by the constrained curves. Figure 2(d) shows the final result after deleting the outer elements, and the optimized mesh are displayed in Figure 2(e).



(a) Initial Triangulated Rectangle (b) Triangulation of the Boundaries and Constraints (c) Interior Points Inserting and Boundary Recovery



(d) Deleting Outer Elements (e) Mesh Optimizing

Figure 2. Delaunay Triangulation for the Domain with Interior Constraints

Algorithm 1. Delaunay Triangulation Algorithm for Domain with Constraints

1. Construct the triangulated rectangle which covers the whole problem domain
2. Discretize each **boundary** curve and **constrained** curve, insert all the discretized boundary points using the Bowyer-Watson constrained Delaunay triangulation algorithm
3. Insert all interior nodes according to size field predefined and recover the missing boundaries using the boundary recovery algorithm introduced in Figure 2
4. Delete the elements out of the problem domain
5. Optimize the mesh quality

4. Boolean Operation

A new Delaunay triangulation algorithm suitable for non-manifold appears after improvements on the algorithm 1, and the new algorithm can effectively deal with the polygon Boolean operation under any circumstance as a new polygon Boolean operation algorithm. The new algorithm has no requirements either for the shape of single polygon, or for the position between the polygons to do the Boolean operation, that is, the polygon can be with holes or be concave, and there can be overlapped edges between the polygons. In addition, the number of the polygons to do Boolean operation can be arbitrary. The algorithm can be applied in the imprint step of mesh generation by multiple sources to multiple target sweeping [11], the imprint algorithm will be omitted in the article.

In the Figure 3, L_t and L_s are respectively the loop of two polygons, and R_t and R_s are polygon regions bounded by Loop L_t and L_s. The Boolean operation defined on R_t and R_s are as follows:

$$\begin{cases} \Theta_o = R_t \cap R_s \\ \Theta_t = R_t - R_s \\ \Theta_s = R_s - R_t \end{cases}$$

The algorithm 2 is an algorithm of Delaunay triangulation for non-manifold which can deal with the Boolean operation of polygons in Figure3. The inputs of the algorithm are loops of polygons, and the output is the result of the Boolean operation. The detailed steps are as follows: first, construct the initial triangulated rectangle as Figure4(a); second, discretize the boundaries and insert all boundary points to get the triangulation as shown in Figure4(b); third, recover the missing boundaries using the algorithm described in Figure 2, and the result of the step can be seen in Figure 4(c). Finally, we use the following rules to judge the result of the Boolean operation:

Rule 1. The two triangles sharing the border edges of the polygons are dyed with different colors.

Rule 2. The two triangles sharing the non-border edges of the polygons are dyed with the same color.

The final result of the Boolean operations corresponding to Θ_o、Θ_t、Θ_s respectively can be seen in Figure 4(d).

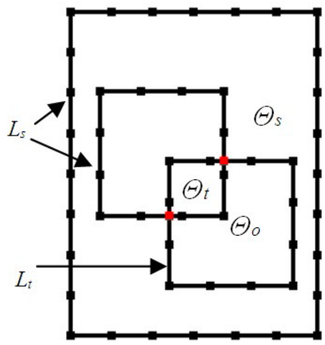
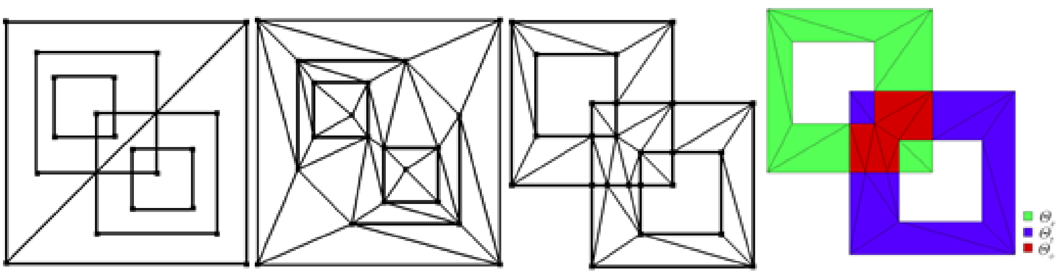


Figure 3. The Schematic Diagram of Boolean Operation



(a) Initial Triangulted Rectangle (b) Triangulation of the Boundary Points (c) Boundary Recovery (d) Boolean Operation Result

Figure 4. The Major Steps of Boolean Operation with Delaunay Triangulation

Algrithm 2. The Delaunay Triangulation Algorithm Suitable for Non-Manifold Used for Boolean Operation

Inputs:

$T = \{e_t\}$:represent all edges of Loop L_t

$S = \{e_s\}$:represents all edges of Loop L_s

Variables:

R_t : The region bounded by loop L_t

R_s : the region bounded by loop L_s

Outputs:

$\Theta_o = \{t_o\}$: all the triangle cells included in the region of $R_t \cap R_s$

$\Theta_t = \{t_t\}$: all the triangle cells included in the region of $R_t - R_s$

$\Theta_s = \{t_s\}$: all the triangle cells included in the region of $R_s - R_t$

$\Theta_e = \{t_e\}$: all the triangle cells included in the region outside R_t and R_s ($\overline{R_s} \cap \overline{R_t}$)

Steps:

1. Construct the initial triangulated rectangle covering the region of R_t and R_s

2. Insert all discretized points of T and S using Bowyer-Watson point insertion algorithm

3. Recover the missing boundaries in T and S using the boundary recover algorithm

4. Judge each triangle t is inside the region of R_t ($t \subset R_t$) or not

5. Judge each triangle t is inside the region of R_s ($t \subset R_s$) or not

6. Visit each triangle in the triangulation result recursively

   if ($t \subset R_t$ and $t \subset R_s$) then t is added to Θ_o ;

   if ($t \subset R_t$ and $t \not\subset R_s$) then t is added to Θ_t ;

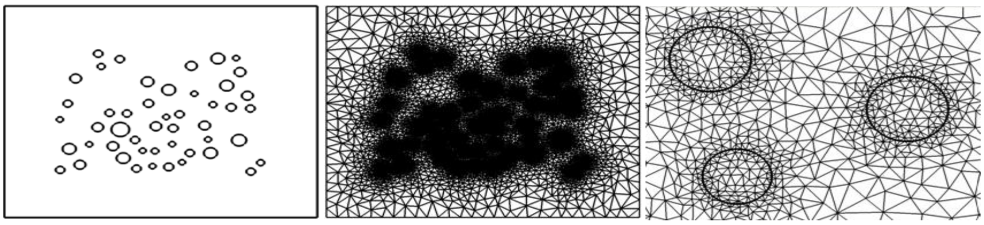
   if ($t \subset R_s$ and $t \not\subset R_t$) then t is added to Θ_s ;

   if ($t \not\subset R_s$ $t \not\subset R_t$) then t is added to Θ_e

7. end the loop

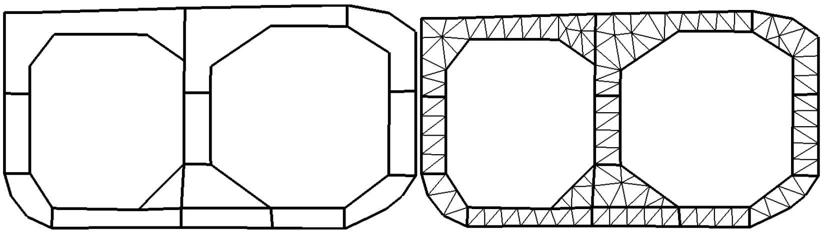
5. Experiments and Analysis

A model with interior constraints is presented in Figure 5(a) which is concrete aggregates containing a number of randomly distributed gravels [12], and Figure 5(b) shows its final mesh. From the enlarged picture in Figure5(c), we can see that constraints on behalf of gravels are retained in the final mesh whose quality is high and size transition is well-graded. Figure 6 demonstrates a cross-Figureal model of a ship [13]. In the conventional algorithms, the model should be divided into a number of small regions without constraint boundaries and then be triangulated with the traditional Delaunay triangulation method. But applying the new triangulation algorithm, the triangulation can be obtained directly without the initial dividing step. Figure7 shows the Boolean operation between a self-intersecting polygon and a square. The Boolean operation with the triangulation algorithm suitable for non-manifold can effectively deal the Boolean operation of polygons under any circumstance, including the cases with holds, concave polygons and self-intersecting, and can be applied in the imprint step of multiple source to multiple target Sweeping.



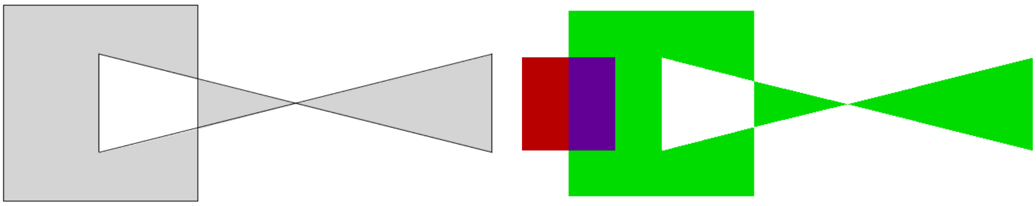
(a) Model of Concrete Aggregates (b) The Final Mesh (c) The Enlarged Mesh

Figure 5. The Mesh of Concrete Aggregates



(a) The Cross-Figure of a Ship (b) The Final Mesh

Figure 6. The Mesh of the Cross-Figureal of the Ship



(a) Polygons with Self-Inter Figure (b) The Result of the Boolean Operation

Figure 7. Boolean Operations between Self-Intersecting Polygons and a Square

6. Conclusions

The article presents a boundary recover algorithm by inserting points on the interFigure of missing boundaries and the triangulation edges, proposes a Delaunay triangulation algorithm to deal with the practical models with interior constraints and applies the algorithm to triangulate the non-manifold models and effectively do Boolean operation of polygons with a majority of special cases. Finally, some practical examples are demonstrated to show the effectiveness and correctness of the algorithm. In addition, the algorithm of the boundary recovery can be extended to three dimensions.

Acknowledgements

This work is based upon wok funded by Natural Science Foundation of P.R. China under Grant No. 11172267, No. 10872182, No. 61100160 and Ningbo Natural Science Foundation of China under Grant No.Y1110038.

References

[1] Z. Q. Guan, C. Song and Y. X. Gu, "Recent Advances of Research on Finite Element Mesh Generation Methods," Journal of Computer Aided Design and Computer Graphics, vol.15 no.1, **(2003)**.
[2] W. Lin, Y. H. Tang and C. B. Zhao, "An algorithm for automatic 2D finite element mesh generation with line constraints," Computer-Aided Design, vol. 43 no. 12, **(2011)**.
[3] K. Y. Lee, I. I. Kim and D. Y. Cho, "An algorithm for automatic 2D quadrilateral mesh generation with line constraints," Computer-Aided Design, vol. 35 no. 12, **(2003)**.
[4] C. Park, J. S. Noh and I. S. Jang, "A new automated scheme of quadrilateral mesh generation for randomly distributed line constraints," Computer-Aided Design, vol. 39 no. 4, **(2007)**.
[5] Q. Du and D. S. Wang, "Recent progress in robust and quality Delaunay mesh generation," Journal of Computational and Applied Mathematics, vol. 195 no. 1, **(2006)**.
[6] J. J. Chen and Y. Zheng, "Redesign of a conformal boundary recovery algorithm for 3D Delaunay triangulation," Journal of Zhejiang University, Science A, vol.7 no. 12, **(2006)**.
[7] J. J. Chen, D. W. Zhao and Z. G. Huang, "Three-dimensional constrained boundary recovery with an enhanced Steiner point suppression procedure," Computers and Structures, vol. 89 no. 5/6, **(2011)**.
[8] H. Si and K. Gärtner, "3D boundary recovery by constrained Delaunay tetrahedralization," International Journal for Numerical Methods in Engineering, vol. 85 no. 11, **(2011)**.
[9] Q. Du and D. S. Wang, "Recent progress in robust and quality Delaunay mesh generation," Journal of Computational and Applied Mathematics, vol. 195 no.1, **(2006)**.
[10] P. L. George, "Improvements on Delaunay-based three-dimensional automatic mesh generator," Finite Elements in Analysis and Design, vol. 25 no.3/4, **(1997)**.
[11] M. W. Lai, S. Benzley and D. White, "Automated hexahedral mesh generation by generalized multiple source to multiple target Sweeping," International Journal for Numerical Methods in Engineering, vol.49 no.1/2, **(2000)**.
[12] Q. H. Gao, Z. Q. Guan and Y. X. Gu. "Automatic Generation of Finite Element Model for Concrete Aggregate," Journal of Dalian University of Technology, vol. 46 no.5, **(2006)**.
[13] B. S. Jang, Y. S. Suh and E. K. Kim, "Automatic FE modeler using stiffener-based mesh generation algorithm for ship structural analysis," Marine Structures, 2008, vol. 21 no.2/3, **(2008)**.

