*O*(*N*)-MERGING OF OVERLAPPING DELAUNAY TRIANGULATIONS

LEONID MESTETSKIY

*Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University*

*MSU, GSP-1, 1-52, Leninskiye Gory, Moscow, 119991, Russia*

*l.mest@ru.net*

NATALIA DYSHKANT

*Research Computing Center, Lomonosov Moscow State University*

*MSU, GSP-1, 1-4, Leninskiye Gory, Moscow, 119991, Russia*

*dyshkant@srcc.msu.ru*

ELENA TSARIK

*Tver State University*

*33, Zhelyabova st., Tver, 170100, Russia*

*elena tsarik@mail.ru*

Received (received date)

Revised (revised date)

Communicated by (Name)

In this paper the problem of combining two Delaunay triangulations into a single Delaunay triangulation is considered. It is assumed that the given triangulations can have overlapping convex hulls. An algorithm with linear time complexity for solving this problem is proposed. The algorithm allows simple implementation.

*Keywords*: discrete surface model; Delaunay triangulations; overlapping triangulations; cutting and merging of triangulations; starter edges.

2. Overlapping Delaunay Triangulations

Let *S* be a set of *n*≥3 points in the Euclidean plane. These points are called sites. A triangulation of *S* is a planar straight line graph with vertices from *S* having the largest possible number of edges. A circle is said to be empty if it does not contain sites in its interior. A triangulation of *S* is called a Delaunay triangulation (DT) and is denoted by *Del*(*S*) if the circumcircle of any triangle is empty. A half-plane is called an improper empty circle if it does not contain sites.

Lemma 2. Given a set V = {vz, ..., VN} of points, any edge (vi, vj) is a Delaunay edge of DT(V) if and only if there exists a point x such that the circle centered at x and passing through v~ and v~ does not contain in its interior any other point of V.

Corollary 1. Given a set V = (v1 ..... vN} of points, the edge (vi, vj) on the boundary of the convex hull of V is a Delaunay edge.

Lemma 3. Given a set V = { v1 ..... vN } of points, Δvivjvk is a Delaunay triangle of DT(V) if and only if its circumcircle does not contain any other point of V in its interior.

Ref. [6]

An empty circle is said to be incident to a site if it passes through this site. An edge of a triangulation is called a Delaunay edge if there is an empty circle that is incident to its endpoints. A face of a triangulation is called a Delaunay face if there is an empty circle that is incident to the faces’ vertices. A triangulation of *S* is called a Delaunay triangulation (DT) and is denoted by *Del*(*S*) if all its faces and edges are Delaunay.

The problem of merging two overlapping DTs is presented in Figure 2. Assume that S=B∪W the sites have two different colors: *B* sites are black and *W* sites are white. The combined DT includes one-color and two-color edges and faces depending on the colors of their incident sites. The one-color edges and faces are included in *Del*(*B*∪*W*) directly from *Del*(*B*) or *Del*(*W*), and the two-color edges and faces are newly created.

5. Cutting and Sewing

The cutting and sewing processes are based on the check of the Delaunay condition for the triangulation edges. The Delaunay condition is checked using the angle criterion, which is based on the following statement.

Let’s S is the finite set of points (sites).

**Lemma 1.** Sites A,B ∈ S form a Delaunay edge, iff the condition ACB+ADB ≤ 180◦ is fulfilled for an arbitrary pair of sites C,D ∈ S that lie on different sides of AB.

**Proof**. This follows from the MAX-MIN angle criterion [6]

Lets see how we can construct cuts in original triangulations with use of the angular criterion.

For the site A, the set of incident edges AC*i*, *i* = 1, . . . , k, is called a *bundle*. In accordance to Ref. 6, we will represent such a bundle by an ordered doubly linked circular list of the adjacent sites C1, . . . ,C*k*. The functions A.pred(*Ci*) and A.succ(C*i*) denote the next site after C*i* in the clockwise and counter-clockwise direction, respectively.

Suppose we have found a two-color Delaunay edge AB (Figure 5). In particular, AB may be a starter. Then, we add the edge AB to the bundles of sites A and B.

Ребро AB вставляется в пучок в соответствующее место так, чтобы сохранить правильную последовательность против часовой стрелки в пучке.

This operation can break the Delaunay condition for some edges in these bundles. Such one-color edges will not be included in the combined triangulation and must be destroyed.

Пусть *A* – левый, а *B* – правый сайты ребра Делоне *AB* и ребро *AB* вставляется в пучок сайта *A* (фиг.4 *а*). Пусть *AC1* и *AC2* одноцветные ребра пучка *A* такие, что *C1=A.succ(B)*, *C2=A.succ(C1)* и сайты *C1* и *C2* лежат слева от *AB*. Тогда нарушение условия Делоне для ребра *AC1*выражается в том, что ∠*AC2C1*+∠*ABC1* >180°. В этом случае ребро *AC1* разрушается и далее рекурсивно аналогичной проверке подвергается ребро *AC2*. Если же ∠*AC2C1*+∠*ABC1* ≤180°, то ребро *AC1* сохраняется и ребро *AC2* не тестируется. Заметим, что если слева от *AB* отсутствует сайт *C2=A.succ(C1)*, то ребро *AC1* не тестируется и не удаляется.

Фиг. 4. Коррекция пучков левого и правого сайтов разноцветного ребра.

*C2*

*B*

*C1*

*A*

*A*

*B*

*D1*

*D2*

(*а*)

(*б*)

*D1*

*D2*

Аналогичная проверка выполняется для ребер, лежащих в пучке сайта A перед ребром AB. Это ребра *AD1* и *AD2* одноцветные ребра пучка *A* такие, что *D1=A.pred(B)*, *D2=A. pred (D1)*

Подобным же образом осуществляется тестирование и коррекция пучка правого сайта *B* (фиг.4*б*).

Thus, every new formed two-color Delaunay edge includes into two bundles. Consequently all corrupted edges in bundles will be destroyed. We call a bundle that has been tested and corrected as a *proper bundle*.

Now, we consider the construction of new two-color edges (Figure 5).

**Lemma 2**. Suppose that AB is a two-color Delaunay edge and the bundles of sites A and B are proper bundles. In addition, suppose that C = A.succ(B), D = B.pred(A) and the sites C and D lie to the left from AB. If ACB ≥ ADB then CB is a Delaunay edge; if ACB ≤ ADB, then AD is a Delaunay edge.

**Proof**. Since AB is a Delaunay edge, there is an empty circle incident to A and B (Figure 5, dashed line). Therefore, C and D lie outside this circle. Let us consider the circumcircles of the triangles △ACB and △ADB. Пучки A и B являются правильными после присоединения к ним ребра AB и соответствующей коррекции. Следовательно, описанные круги △ACB and △ADB являются пустыми. It is obvious that the arcs of these circles that are to the right of AB are contained within the empty circle of the Delaunay edge AB; therefore, the segments of these circles lying to the right of AB don’t contain sites. To the left of AB, one of these circles is also empty. This circle corresponds to the triangle that has a larger angle opposite to AB. This follows from the assumption that the bundles corresponding to A and B are proper bundles.

This lemma provides a basis for the construction of the next adjacent stitch for the current stitch. The seam construction is similar to the merging of overlapping triangulations in the Lee-Schachter algorithm.6 Заметим, что если ACB = ADB, то в качестве следующего стежка стежка можно выбрать любое из ребер AC или DB.

Fig. 5. Construction of a new multi-color Delaunay edge.

*A*

*B*

*D*

*С*

If a one-color edge was destroyed when constructing a new stitch (i.e., if the cut was continued), then a new adjacent stitch appears. This follows from the next statement.

**Lemma 3.** If a site is incident to a destroyed one-color edge, then it is incident to a two-color edge in the combined triangulation.

**Proof**. The one-color destroyed edge AB (Figure 6) had an incident empty circle C1 in the initial triangulation. In the process of merging, this circle ceased to be empty; therefore, it now contains sites of the different color. Then, it is easy to see that this circle contains a site D of the other color that has the common incident empty circle C2 with the site B. This circle lies inside the circle C1 and has a common tangent with it at the point B. Hence, the pair of the multi-colored sites B, D forms a Delaunay edge, which will be included in the combined triangulation.

*A*

*B*

*D*

*C1*

*C2*

Фиг. 6. К лемме 4.2.

Thus, when the starter is given, the process of the simultaneous cutting and sewing is as follows.

(1) Create an initial stitch from the starter (a two-color edge). Include this edge into the bundles associated with its sites. Set the new stitch as current.

(2) Correct the bundles associated with the sites of as illustrated in Figure 5. As a result, some of the one-color edges can be destroyed. After the correction, the bundles of both sites become proper.

(3) Build a new adjacent stitch for the current stitch; i. e., build a new two-color Delaunay edge as illustrated in Figure 5. If no new stitch can be built, then the current stitch is the end stitch of the seam. If this happened for the first time with the seam being created, then we conclude that this is the end of the seam and the seam should be continued to the other side. For this purpose, one must return to the first stitch and “reverse” it, i.e., redefine the left and the right sites in it. Then, we declare the first stitch to be current again and go to step 2. If no new stitch can be built for the second time, then the current stitch is the second end stitch of the seam and the algorithm terminates.

(4) Check whether the new stitch coincides with the starter. If it does, then the seam is cyclic and all its stitches are found. Than algorithm terminates. Otherwise, we set the new stitch as the current one and go to step 2.

Sec 2:

\* What exactly is the input to your algorithm? Do you allow S\_1 and S\_2 to share points? Do you make specific assumptions? E.g., do you assume general position of the input?

Что точно является входом вашего алгоритма? Вы позволяете S\_1 и S\_2 иметь общие точки? Делаете ли вы конкретные предположения? Например, предполагается ли общее положение точек во входных данных?

\* An algorithm might have an O(...) worst-case complexity; a problem does not have an O(...) but an \Omega(...) or possibly \Theta(...) worst-case complexity.

Алгоритм может иметь O (...) сложность в худшем случае; Задача не имеет O(...), но Ω(...) или, возможно, Ɵ(...) сложность в худшем случае.

\* Lem 2: What shall we do if the two angles are identical? GPA assumed? (grade point average)

Лем 2: Что мы будем делать, если два угла одинаковы?

 \* Proof of Lem 2: "Therefore, C and D lie outside this circle."  
   Or on the circle!  
   "To the left of AB, one of these circles is also empty."  
   Why?

Доказательство Лем 2: "Таким образом, С и D лежат вне этого круга."  
Или на круге!  
"Слева от AB один из этих кругов также является пустым".  
Почему?

\* Proof of Lem 6: "Then, their circles of influence meet at certain points C  
   and D."  
   Why are they guaranteed to intersect? You seem to miss some argument.  
   (Granted, it's not difficult.)

Доказательство Лемма 6: "Тогда, их круги влияния пересекаются в определенных точках C и D."  
    Почему они гарантированно пересекаются? Вы, кажется, пропустили некоторые аргументы.  
    (Конечно, это не сложно)

   "...the diameters CE and CF of the circles A\_0 and B\_0"  
   A\_0 and B\_0 are points and no circles!

"... диаметры СЕ и CF кругов A\_0 и B \_0"  
    A\_0 и B\_0 являются точками а не кругами!

"It follows from Lemma 5 that A lies on the arc ED..."  
   Why?  
   BTW, the claim AB < 2A\_0B\_0 and A\_1B\_1 < 2A\_0B\_0 simply follows from  
   the Intercept Theorem.

(The **intercept theorem**, also known as Thales' **theorem** (not to be confused with another **theorem** with that name), is an important **theorem** in elementary geometry about the ratios of various line segments that are created if two intersecting lines are **intercepted** by a pair of parallels.)

"Из леммы 5 следует, что A лежит на дуге ED ..."  
Почему?  
Кстати, требование АВ <2A\_0B\_0 и A\_1B\_1 <2A\_0B\_0 просто вытекает из Теоремы Фалеса.

References

1. N. Dyshkant“Comparison of Point Clouds Acquired by 3D scanner” // Lecture Notes in Computer Science: Discrete Geometry for Computer Imagery. – Vol. 7749. – Pp. 47–58 (2013)

2. Mestetskiy, L., Tsarik, E.: Delaunay triangulation: recursion without spatial sepa- ration (in Russian). In: GrafiCon-2004: Proc. 14th Int. Conf, CMC MSU, Moscow, Pp. 267–270 (2004)

3. N. Dyshkant “Localization Problems in Delaunay Triangulations” // Abstracts of the 10th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2013). – Pp. 29–33 (2013)

4. L. M. Mestetskiy, Continuous Morphology of Binary Images: Figures, Skeletons, Circulars (Fizmatlit, Moscow, 2009) [in Russian].

5. Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. Computational Geometry: Algorithms and Applications (3rd ed.). TELOS, Santa Clara, CA, USA (2008)

6. D.T. Lee and B.J. Schachter, Two algorithms for constructing a Delaunay triangulation, International Journal of Computer and Information Science 9 (1980) 3, 219–242.

7. F.P. Preparata and M.I. Shamos, Computational geometry An Introduction, Springer- Verlag (1985)

8. D.G.Kirkpatrick, Efficient computation of continuous skeletons, *in: proceedings of the 20th Annual IEEE Symposium on FOCS* (1979) 18–27.

9. D. Cheriton and R.E. Tarjan, Finding minimum spanning trees, *SIAM J.Comput.* **5** (1976) 724–742.

10. B. Chazelle, An optimal algorithm for intersecting three-dimensional convex polyhedrons, *SIAM J.Comput* **21** (1992) 671–696.

11. R.A. Dwyer, A faster divide-and-conquer algorithm for constructing Delaunay triangulations, Algorithmica 2 (1987) 137–151.

12. M.I. Shamos and D. Hoey, Closest-point Problems, *in: proc. 16th IEEE Annu. Symp. Found. Comput. Sci.* (1975) 151–162.