*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.

1. Introduction

Задача вычисления триангуляции Делоне (ТД) n сайтов-точек имеет нижнюю оценку вычислительной сложности Ω(n log n). В тех случаях, когда имеется некоторая дополнительная информация для множества сайтов, эта оценка может быть улучшена. В частности, представляет интерес так называемая задача слияния двух триангуляций, когда исходное множество сайтов состоит из двух непересекающихся подмножеств S=S1∪S2, и ТД Del(S1) и Del(S2) уже построены. В [6] описан O(n) алгоритм слияния ТД в случае, когда множества точек S1 и S2 линейно разделимы. Мы рассматриваем более общую постановку задачи слияния ТД: множества сайтов S1 и S2 не пересекаются S1∩S2 =∅, но перекрываются, т.е. пересекаются их выпуклые оболочки Conv(S1)∩Conv(S2) ≠ ∅. Даны ТД Del(S1) и Del(S2), нужно построить ТД Del(S1∪S2) за время O(n) в худшем случае.

В такой постановке задача слияния ТД до сих пор не рассматривалась, возможно, потому, что не представляла практического интереса. Наш интерес к этой постановке связан с построением алгоритма для вычисления расстояния в метрическом пространстве функций двух переменных, заданных на конечных нерегулярных множествах точек. Эта задача возникает, в частности, при анализе 3D поверхностей человеческих лиц, полученных в результате пространственного сканирования [1]. Предложенная модель определяет расстояние между такими функциями на основе построения общей ТД. При этом расстояние между моделями лиц определяется как минимальное по всем движениям множеств S1 и S2 на плоскости. Таким образом, осуществляется подгонка пары поверхностей друг к другу. И на каждой итерации подгонки выполняется слияние триангуляций.

The considered problem of surface comparison is as follows: there exist two surfaces determined by the height functions at two finite sets of points—non-regular grids; it is necessary to calculate some measure of similarity (or difference) between them. An approach proposed by the authors for surface comparison1 consists in the following steps: Delaunay triangulations (DTs) construction on the both grids, interpolation of each of the functions with respect to the other grid, the construction of the common triangulation for the two grids, and construction of *l*2 metrics for two functions on this triangulation. In other words the developed algorithm is based on the idea of supplementing the values of each function at the nodes of the other grid by constructing the DTs and their localization in each other. Note that the numerical complexity of one calculation of the metrics is important because in applications it is required to calculate it several times during iteration process of surface matching. Influence of non-efficient measure calculation will increase on each iteration step.

Поскольку ТД и диаграмма Вороного (ДВ) n сайтов являются двойственными графами и могут быть получены один из другого за время Ɵ(n), для решения задачи в принципе можно использовать известные алгоритмы [8,10]. Алгоритм Киркпатрика [8] строит ДВ Vor(S1∪S2) перекрывающихся множеств сайтов S1 и S2 на основе слияния ДВ Vor(S1) и Vor(S2). Для решения нашей задачи нужно преобразовать исходные ТД Del(S1), Del(S2) в Vor(S1), Vor(S2), построить Vor(S1∪S2) с помощью ТД алгоритма [8], а затем преобразовать Vor(S1∪S2) в Del(S1∪S2). Алгоритм Шазеля [10] строит пересечение двух выпуклых многогранников в 3d-пространстве. Для его использования применительно к нашей задаче требуется построить из исходных ТД Del(S1), Del(S2) сначала ДВ Vor(S1), Vor(S2), затем выпуклые многогранники. Затем нужно построить пересечение этих многогранников с помощью алгоритма [10]. После этого из многогранника получается сначала ДВ Vor(S1∪S2), а после Del(S1∪S2). Таким образом, используя алгоритм [8] и [10], мы получаем решение за 3 и 5 шагов соответственно. И хотя вычислительная сложность этих решений Ɵ(n), их практическая реализация выглядит весьма проблематичной. В отличие от этого, предлагаемый нами алгоритм находит Del(S1∪S2) непосредственно на основе слияния исходных ТД Del(S1) и Del(S2).

Another reason for direct DT algorithms (versus VD algorithm) is that the data structures describing the DTs are usually simpler than those describing the VDs. In the DT graphs, all the edges are of the same type – they are segments of finite length. In VDs, there are also rays, which are the sides of infinite Voronoi polygons. This fact complicates the description of algorithms. Apparently, this feature was one of the reasons for the development of the Lee-Schachter DT algorithm6, although the similar VD algorithm described in Ref.12 had already been known. Thus, the development and implementation of an efficient algorithm for merging overlapping DTs is still actual.

Структура статьи следующая: …..

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.

3. Problem Discussion and Related Work

4. The Structure of the Algorithm

Maximal one-color connected components in the combined triangulation *Del*(*B*∪*W*) will be called patches (Figure 2b). These components are directly transferred from Del(B) or Del(W) into *Del*(*B* ∪ *W*) without any changes. The other one-color edges of *Del*(*B*) and *Del*(*W*), which were not included into the patches, must be destroyed during the merge. We shall call these edges as corrupted. Subsets of corrupted edges are called cuts of the initial DTs (Figure 2c,d). The merge process can be considered as the extraction of patches in *Del*(*B*) and *Del*(*W*) and then joining the patches by two-color edges called stitches. A subset of stitches joining two patches is called a seam. These definitions are illustrated by Figure 2. Patches are shown by the solid line and the dotted line shows the seams.

Thus, the merge of two DTs is organized as a consecutive construction of cuts, and creating seams.

The seams can be open, and cyclic (Figure 2b). An open seam has two end stitches in the convex hull *Conv*(*B*∪*W*). All the stitches of a cyclic seam do not belong to the convex hull. The sewing process includes two stages: the search for the first stitch belonging to the seam and then the consecutive creation of a cut and a seam from the first stitch. The first stitch is called a starter. The starter initializes both the cutting and the sewing processes. In our algorithm the cutting and the sewing processes proceed simultaneously.

The algorithm involves the following procedures:

1. find an initial starter,
2. build a cut and a seam from the starter,
3. try to find the next starter,
4. if the next starter is found, then go to step (ii); otherwise, stop.

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.

6. The Starter Search Algorithm

We shall say that an incident circle for the site *V* is semi-empty if this circle is empty with respect to the initial DT containing *V* and is not empty with respect to the other initial DT.

**Lemma 4** (a starter existence sufficiency condition). A site has an incident two-color Delaunay edge if существует a semi-empty circle for this site.

**Proof**. Assume that, for some site *A*, an incident circle centered at *C* does not contain the sites of the same color as *A* but contains one or more sites *D*1*, . . . ,Dk* (*k ≥* 1) of the other color inside itself (Figure 8). Let us consider the set of circles that are incident to the pairs of sites (*A,Di*)*, i* = 1*, . . . , k*, that have a common tangent with *C* at the point *A*. It is easy to see that the circle in this set that has the minimal radius is empty and it is incident to a two-color pair of sites. Hence, this pair of sites forms a Delaunay edge; and it can be used as a starter.

Therefore, if the site *A* has a semi-empty circle centered at *C*, then we can find sites *D*1*,...,Dk* внутри круга and examine the set of circles that are incident to the pairs of sites (*A,Di*), where *i* = 1*, . . . , k*, that have a common tangent with *C* at the point *A*. The circle with the minimum radius is empty and incident to a two-color pair of sites. Consequently this pair of sites forms a Delaunay edge and can be used as a starter.

Фиг. 8. К лемме 4.

*Dmin*

*A*

*С*

*Di*

6.1. Search for the First Starter

The site *V*1 is said to lie to the left of the site *V*2 if *V*1 = (*x*1*, y*1) precedes *V*2 =(*x*2*, y*2) in lexicographically ascending order, i. e. if either *x*1 *< x*2 or *x*1 = *x*2 and *y*1 *< y*2.

Let *W*min and *B*min be the leftmost sites in *W* and *B* respectively. Without loss of generality, we may assume that *W*min lies to the left of *B*min. Let us examine the circle incident to the sites *W*min and *B*min, centered on the horizontal ray that goes from *B*min to the left. Этот круг может оказаться пустым и тогда пара *W*min and *B*min образует стартер. Если же он не пустой, то внутри у него лежат сайты из W и он является полупустым. It is obvious that *B*min and this circle satisfy the conditions of Lemma 4 and, consequently, we can find a starter using the method illustrated in Figure 8. The time taken by this procedure is *O*(|*B*|+ |*W*|).

6.2. Minimum Spanning Trees of DTs

A solution of the starter search problem is based on the use of the minimum spanning trees (MST) of the initial DTs. A DT minimum spanning tree is defined as a connected subgraph of DT having the least total length of the edges. MSTs were used by Kirkpatrick8 for merging overlapping Voronoi diagrams8. The same idea can be used for merging overlapping DTs. We obtain MSTs for the initial DTs by Cheriton-Tarjan algorithm9 which has time complexity *O*(|*B*|+ |*W*|).

The use of MSTs for finding a starter is based on the obvious fact that if a DT loses the connectivity because of a cut, then at least one edge of the MST is destroyed. Убрать?

A circle whose diameter is a DT edge is called the circle of influence of this edge. Consider some important properties of MSTs edges.

**Lemma 5.** The circle of influence of MST edge is a DT empty circle.

**Proof.** Assume the opposite: let *AB* be an edge of a MST, and let the circle with the diameter *AB* contain a point *C*. In Δ *ABC*, the angle ∠*C* is obtuse; therefore, *AC < AB* and *BC < AB*. Then, if we remove *AB* from the MST, the tree breaks up into two connected components one of which contains the vertex *C*. If *C* belongs to the same component as *A*, it is incident to a tree edge *CB*, which is shorter than *AB*. Similarly, if *C* belongs to the same component as *B*, we can replace the edge *AB* by *AC*. In both cases, we obtain a spanning tree that is shorter than the initial tree, which contradicts the assumption that *AB* belongs to the MST.

**Lemma 6.** The distance between the centers of two MST edges is not less than half of the length of each of these edges.

**Proof.** Assume the opposite. Let AA1 and BB1 be edges of an MST with the midpoints A0 and B0 such that A0B0 ≤ AA0 (Figure 9). Then their circles intersect at some points C and D. Without loss of generality, suppose that AB ≥ A1B1. Let us draw the diameters CE and CF of the circles A0 and B0 from the point C. In the ΔCDE, the angle ∠CDE is right; and in the ΔCDF, the angle ∠CDF is right; hence, EF passes through the point D.

It follows from Lemma 5 that A lies on the arc ED and the point B lies on the arc DF. The angels EAD and DBF are obtuse because the arcs ED and DF are less than semicircles. Hence, AB < AD + DB < ED + DF = EF = 2 A0B0; i. e. AB < AA1. Since A1B1 ≤ AB, we have A1B1 < AA1.

Фиг.9. К лемме 6.

*A1*

*F*

*B*

*A*

*C*

*B1*

*B0*

*D*

*E*

*A0*

Let us show that in this case AA1 cannot be an edge of the MST. Indeed, if we remove the edge AA1 from the spanning tree, the one of the points A or A1 appears in the same component as the edge BB1. If this is *A*, then AA1 may be replaced by A1B1 and we obtain a shorter spanning tree. If this is A1, then AA1 may be replaced by ABwith the same result. This contradiction proves the lemma.

If the connectivity of a DT is violated by a cut, then at least one edge of the EMST is destroyed. The idea of searching a starter using an EMST is based on this obvious fact.

6.3. Search for the Next Starter

A destroyed MST edge is called *a bridge* (following Ref. 8). Since an edge (bridge) is destroyed, one of its sites is already included into the combined triangulation and the other site is not. We will call these sites fixed and free, respectively.

**Lemma 7.** If a site is free, then an incident starter does exist for it.

**Proof.** Consider a bridge that is incident to a free site. By definition, it is an edge of the MST of one of the initial triangulations; therefore, by Lemma 5, its circle of influence contains no other sites of the same color. However, since this edge was destroyed while constructing a cut, the circle of influence encloses sites of the other color. Hence, the free site under consideration satisfies the conditions of Lemma 4, which proves the lemma.

Thus, when a bridge is destroyed, a free site appears. Then, there exists a starter incident to this site. The following statement describes a property that facilitates the search of the second site for the starter.

**Lemma 8.** Let the sites B and C form a starter. Suppose that B is free and the initial DT T contains the site C. Then, B is an internal point of at least one maximum empty circle in T that is incident to C.

**Proof.** Any empty circle incident to C belongs to the union of maximal empty circles incident to C. Therefore, if the sites C and B form a Delaunay edge, then any empty circle incident to C also belongs to this union, which proves the lemma.

Let AB be a bridge, the site A be fixed, and the site B be free (Figure 10). By Lemma 5, B can be used as a new starters first site. The sites A and B have the same color (black in Figure 10), and the circle of influence of AB contains sites of the other color (white in Figure 10) because the bridge AB is formed by a destroyed edge. Together with B, one of these sites forms a new starter because the circle of influence is semi-empty and the conditions of Lemma 4 are satisfied.

This lemma indicates that the second site for the new starter is to be found among the vertices of the faces of *T* whose circumcircles contain *B*. In the example in Figure 10, the circumcircles of the faces Δ*C*3*C*4*C*5, Δ*C*3*C*5*C*6, and Δ*C*6*C*5*C*7 of the white initial DT *Del*(*W*) contain the site *B*. Therefore, we restrict our search to their vertices *C*3, *C*4, *C*5, *C*6, and *C*7. Moreover, only *C*3, *C*4, *C*6 are within the circle of influence of the bridge *AB*. Hence, we apply the procedure illustrated in Figure 8 only to the sites *C*3, *C*4, and *C*6.

Фиг. 10. Трассировка ребер и граней ТД вдоль моста *AB*. Свободный сайт *B* попадает в описанные окружности граней ΔC3C4C5, Δ*C3C5C6*, Δ*C6C5C7*.

*A*

B

*C1*

*C2*

*C3*

*C4*

*C5*

*C6*

*C7*

To implement this process, we have to solve the bridge localization problem: find a face in the initial DT whose circumcircle contains the free site. Let us examine the edges of the DT that intersect the bridge (Figure 10). These edges are naturally ordered in the direction from the fixed site to the free one. It is easy to see that the last edge intersected by the bridge is incident either to the triangle face containing *B* or to the half-plane containing *B*. This face or half-plane provide a solution of the bridge localization problem. Thus, the bridge localization problem is solved by tracing the edges and the faces of the triangulation along the bridge.

The circumcircle of resulting face contains the free site *B*. The other triangular faces whose circumcircles contain *B* are found by examining the adjacent faces. Combining these faces, we obtain a polygon that contains the site B (the polygon *C*3*C*4*C*5*C*7*C*6 in Figure 10).

Thus, the search for a starter on the basis of a free site incident bridge can be implemented as follows:

(i) Examine the ordered edges of the DT intersected by the bridge. Find a face whose circumcircle contains the free site (Figure 10).

(ii) Find all the adjacent faces whose circumcircles contain the free site (Figure 10).

(iii) Find the vertices of these faces that are contained in the bridge’s circle of influence.

(iv) Among these vertices, select the site corresponding to the minimum radius circle centered at the bridge and incident to this site and to the free site (Figure 8).

7. Time Complexity of the Algorithm

7.1. Complexity of the Starter Search

Let *n* = |W| + |B|be the total number of sites in the two initial DTs. The search for the first starter takes a time *O*(*n*) in the worst case. The search for the next starters includes two scan processes. The first one is the bridge selection among the destroyed MST edges. The second process is the edge tracing performed while solving the bridge localization problem. It is easy to find the worst cases in which the number of bridges is *O*(*n*) and the number of edges intersected by a bridge is also *O*(*n*). Nevertheless, it will be shown that the total number of intersections of the edges of the initial DTs with the bridges is *O*(*n*) in any case.

Let us consider the selection of the next bridge. The total number of edges in the MST is *O*(*n*). The edges of the MST that were destroyed in the process of a seam construction are moved to a special list. If both sites incident to such an edge become fixed, the edge is deleted from this list. The first edge in this list is used as the next bridge for the starter; this requires a time *O*(1). The DT merging process completes when this list become empty.

Assume that one of the initial triangulations, for example *Del*(*W*), is cut and the cut divides it into two components. It is obvious that at least one of the MST edges is destroyed, i.e. a bridge appears. It follows from the properties of MSTs that the circle of influence of the bridge does not contain the endpoints and the midpoints of other bridges. This fact allows us to estimate a measure of intersection of a bridge with the circle of influence of another bridge.

**Lemma 9.** Any bridge can cut from the circle of influence of another bridge an arc not greater than 60◦.

**Proof**. Let AA1 and BB1 be the bridges with the midpoints A0 and B0, respectively (Figure 11a), and D1 and D2 be the points where AA1 meets the circle of influence of BB1. It follows from Lemma 5 and 6 that A0, A, and A1 lie outside the circle B0. Hence, the segment D1D2 entirely lies on AA0 or A0A1. Let, for definiteness, D1D2 ⊆ AA0. Then, D1D2AA0. In the triangle △AA0B0, the side AA0 is no longer than the sides AB0 and A0B0. Hence, ∠AB0A0 ≤60◦. Therefore, ∠D1B0D2 ≤ 60◦, which completes the proof of the lemma.

Фиг.11a. К лемме 9.

*A*

*B*

*B1*

*A1*

*B0*

*D2*

*A0*

*D1*

**Lemma 10**. Suppose the bridges AA1 and BB1 of the same color intersect the DT edge PQ of the other color and the circles of influence of both bridges contain the site P. Then, the difference between the angles ∠APA1 and ∠BPB1 is not greater than 60◦.

**Proof**. The circle of influence of the bridge BB1 has two intersection points with AA1; we denote them by D1 and D2 (Figure 11b). Let us designate an arc of a circle by the symbol ⎝. Since ∠BPB1 lies inside ∠APA1, we have ∠APA1 = ∠BPB1 +∠APB +∠A1PB1. It is obvious, that ∠APB ≥∠D1PB ≥ 0.5 ·⎝D1B and ∠A1PB1 ≥∠D2PB1 ≥ 0.5 ·⎝D2B1.

Then, ∠APB +∠A1PB1 ≥ 0.5 · ( ⎝D1B+ ⎝D2B1 ) = 0.5 · (180◦−⎝D1D2). According to Lemma 9, we have ⎝ D1D2 ≤ 60◦. Therefore, ∠APB + ∠A1PB1 ≥ 0.5 · (180◦ − 60◦) = 60◦.

Hence, ∠APA1 ≥∠BPB1 + 60◦, which proves the lemma.

Фиг.11b. К лемме 10.

*B*

*A1*

*B1*

*D1*

*P*

*D2*

*A*

Фиг.12. К лемме 7.2.

*A*

*B*

*B1*

*A1*

*D1*

*Q*

*P*

*D2*

Фиг.12. К лемме 7.2.

*A*

*B*

*B1*

*A1*

*D1*

*Q*

*P*

*D2*

**Lemma 11**. If a DT edge intersects several bridges of the other color, then each endpoint of this edge can fall inside the circles of influence of no more than two bridges.

**Proof**. Let the edge PQ intersect several bridges and the point P is inside the circles of influence of three bridges AA1, BB1, and CC1 (Figure 11c). Then, by Lemma 10, we have: ∠APA1 ≥∠BPB1+60◦ ≥∠CPC1+120◦. However, ∠APA1 ≤ 180◦; consequently, ∠CPC1 ≤ 60◦. Since the angle ∠CPC1 is subtended by the diameter CC1 of the circle of influence, the point P lies inside this circle. Hence, it cannot be less than 90◦. We have arrived at a contradiction, which shows that P cannot be inside the circle of influence of the third bridge CC1, which completes the proof.

Using these statements, we can estimate the number of bridges in one DT that can be destroyed by an edge of the other DT when they are combined.

Фиг.11c. К лемме 11.

*A*

*B*

*A1*

*Q*

*P*

*C1*

*C*

**Theorem 1.** When two DTs are merged, any edge destroys not more than four bridges.

**Proof**. It follows from Lemma 11 that a bridge is destroyed by an edge only when an endpoint of the edge is inside the bridge’s circle of influence. By the same lemma, an endpoint can belong to the circles of influence of not more than two bridges. Hence, the number of bridges destroyed by an edge cannot exceed four.

This theorem provides a means for estimating the time complexity of searching all the starters. If two segments intersect and each of them is a Delaunay edge in its own triangulation, then one of them must be destroyed. This follows from Lemma 1. Consequently, if a bridge and an edge intersect, then either the bridge destroys the edge or the edge destroys the bridge. The edges destroyed by a bridge meet only a single bridge because by the moment when the next bridge is examined these edges are already removed. If an edge destroys a bridge, then it can destroy not more than four bridges (Theorem 1). Consequently, the total number of intersections of the bridges with the edges cannot exceed the total number of edges in both initial DTs more than by a factor of 5; therefore, it is O(n). Thus, we have proved the following theorem.

**Theorem 2.** The time complexity of the search for all the starters is O(n).

7.2. Complexity of Triangulation Merging

In the merging of triangulations, one-color edges of the initial DTs are destroyed, and two-color edges of the combined DT are created.

The creation of a two-color edge requires one angle test for a pair of competing edges; i.e. it takes a fixed time *O*(1). For each new two-color edge, the adjacent one-color edges are checked for correctness. This requires two angle tests (the criterion of Lemma 1). If an edge is destroyed in accordance with the test result, then another one-color edge becomes adjacent to the two-color edge, and it is also tested. Thus, the time required for the elimination of a one-color edge is estimated as a constant *O*(1). The number of the constructed edges does not exceed the total number of edges in the combined DT *O*(*n*), and the total number of destroyed edges does not exceed the number of edges in both initial DTs *O*(*n*).

We must also take into account the time required to insert new edges into the bundles of their end sites. While inserting the first stitch for a starter, the scan of all the edges belonging to the bundles of its both end sites might be required. Therefore, the total time required to insert the initial edges into the bundles is estimated as *O*(*n*). However, each next stitch doesn’t require the complete bundle scan because it is adjacent to the previous stitch in the bundle. Therefore, the total time required for the insertion of all the edges is *O*(*n*).

Thus, taking into account Theorem 2, we have proved the main result of this paper.

**Theorem 4**. The proposed algorithm combines two overlapping DTs in a time O(n).

It is obvious that the size of memory used by the algorithm is also *O*(*n*).

9. Conclusions

The computationally efficient algorithm for combining of two overlapping DTs was proposed. The estimations for time complexity of the algorithm was proved, the proposed approach was theoretically justified. The proposed approach allows very efficient calculation of metrics for surface comparison: putting construction of DTs and MSTs for initial sets at the preprocessing stage, we can calculate it in linear time. During surface matching process the algorithm will do linear amount of operations on each iteration. So the total estimation of the algorithm for surface comparison is *O*(*n* log *n*)+*O*(*mn*), where *m* is the number of iterations, *n* is the total number of points in two initial sets.

10. Acknowledgements

This work was supported by the Russian Foundation for Basic Research, project no. 14-01-00716-a.

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.