UPDATING ALGORITHMS FOR CONSTRAINT DELAUNAY TIN

WU LiXin [a,b] WANG YanBing [a, c]

[a] Institute for GIS/RS/GPS & Subsidence Engineering Research, China Uni Mining & Technology, Beijing, China, 100083 - awulixin@263.net, awulixin@yahoo.com
[b] Institute for GIS/RS/GPS & Digital Mine Research, Northeastern University, Shenyang, China, 110004
[c] Laboratory of 3D Information Acquisition & Application, Capital Normal University, Beijing, China - cnuwyb@yahoo.com.cn, wybing@cumtb.edu.cn

KEY WORDS: Updating Algorithm, Constraint Delaunay TIN, Integral Ear Elimination, Influence Domain Re-Triangulation, Polymorphism

ABSTRACT:

Present algorithms for D-TIN are far insufficient for the dynamic updating of CD-TIN in different applications. This paper, based on improvement to present insertion and deletion algorithms for D-TIN, a group of algorithms for CD-TIN updating are presented. 1) According to the polymorphism of the constraints in CD-TIN, virtual point is adopted to describe the cross between constraint edges in CD-TIN; 2) Improved from the EE algorithm for D-TIN, an algorithm namely Integral Ear Elimination (IEE) is presented for point deletion in CD-TIN; 3) Considering the polymorphism of constraint edges in CD-TIN, an algorithm namely Influence Domain Re-triangulation for Virtual point (IDRVP) is also presented for deleting constraint lines. An example is shown to test the algorithms.

1. INTRODUCTION

Delaunay triangular irregular networks (D-TIN) and nonstraint delaunay triangular irregular networks (CD-TIN) are two basic concepts in computational geometry, and have many practical applications in several fields including geographic information systems (GIS), finite element methods (FEM), computer graphics and 3D reconstruction (Aurenhammer, 1987; Mostafavi, *et al*. 2003; Zhu, 2000). The method to construct a D-TIN based on a set of given points has been widely studied, and many construction algorithms for D-TIN were proposed, such as divide-and-conquer (Dwyer, 1987; Guibas, 1985), sweepline (Fortune, 1987), incremental (Preparata, 1985; Green, 1977; Guibas, 1992; Ohya, 1984), asymptotically optimal algorithm (Chew, 1987). Furthermore, the updating algorithms for D-TIN insertion and deletion are studied. Devillers (1999) presented ear elimination (EE) algorithm to delete single point in 2D delaunay triangulations (DT). Mostafavi (2003) improved the EE algorithm for deleting point in D-TIN and Voronoi diagram. Marc (2002) presented a method for a set of points to be dynamically updated in regular triangulations (RT). Li (2000) suggested delete some inside points during clipping the boundary of a D-TIN.

Giving a group of endpoints and some non-crossing edges to a D-TIN, a CD-TIN following the criterion of *Visibility* and *Constraint Empty Circle* is obtained (Hao, 2003). The presented updating algorithms for inserting vertices and segments in CD-TIN are constraint graph algorithm (Lee, 1986), divide-and-conquer (Dwyer, 1987; Guibas, 1985), incremental (Preparata, 1985; Mostafavi, 2003) and shell triangulation (Piegl, 1993). However, the updating of CD-TIN not only includes the insertion of contained point and constraint edges, but also includes the deletion of constraint edges. Based on the improvement and extension to present algorithms, this paper proposed integral ear elimination (IEE) algorithm for point deletion, and influence domain re-triangulation for virtual point (IDRVP) algorithm for the insertion and deletion of constraint edges.

2 ALGORITHM FOR CONSTRAINT EDGE INSERTION IN CD-TIN

The insertion of constraint edges is a basic operation for CD-TIN construction. Tsai and Vonderohe presented Convex Hull algorithm that inserting constraint edges before Convex Hull calculating and data dividing for D-TIN (Li, 2000). Marc (2000) improved the asymptotically optimal algorithm for direction constraint CD-TIN construction, which recursive re-triangulated the influence domain of constraint segments. However, the mentioned algorithms can be applied for the intersection of two intersected constraint edges. Virtual point is introduced to solve the problem.

2.1 The Introduction of Virtual Point

If a new constraint segment A_1 intersects with A_2 in CD-TIN, the point *v* of intersection would be added in A_1 and A_2 as a new point for updating CD-TIN. The intersection point *v* is called *virtual point*, and it will be deleted while delete A_1 or A_2, i.e., *virtual point v* is only an interpolated inner point of the constraint segments.

The broken lines are composed of several segments, and constraint polygons are composed of constraint segments end to end. The insertion and deletion of broken lines and polygon can be conducted as segments insertion and deletion in order.

2.2 Influence Domain and CD-TIN Re-Triangulation

Without the consideration for polymorphism, the influence domain re-triangulation presented by Marc (2000) for inserting constraint segments cannot be applied directly for the intersection of constraint. Based on the introduction of virtual point, it is extended for constraint edge with polymorphism being considered.

Let p_1, p_2 be the two endpoints of constraint segment p_1p_2 (p_1, $p_2 \in V$), the procedures for p_1p_2 insertion into CD-TIN can be summarized as the following:

1) to insert constraint segment p_1p_2 in original CD-TIN: all edges e_i (i=1, 2, …n) which are crossed by p_1p_2 are saved in an edge stack. If an edge e_i is a constraint segment, then e_i is marked e'_i and the virtual point v_i intersected between e'_i and p_1p_2 is inserted.

2) to delete all the non-constraint edges e_i crossed by p_1p_2 so that a polygon that we called Influence Domain of constraint edge p_1p_2 without triangulation is left. Clearly, the influence domain contains some constraint edges intersected with p_1p_2.

3) to recursively re-triangulate the upper and lower influence domain (V_{Ui}, V_{Di}, i=1, 2, … n) that is cut by constraint segment p_1p_2 and e'_i following *constraint empty circle* criterion.

Figure 1 shows the insertion process of a constraint segment p_1p_2 in original CD-TIN. For the intersection of constraint segment p_1p_2, p_3p_4 and p_5p_6, the intersecting point v_1 and v_2 are interpolated into p_1p_2 as virtual points so that p_1p_2 is composed of three constraint segments p_1v_1, v_1v_2 and v_2p_2. To search all the edges crossed by p_1p_2 and to remove all the non-constraint edges to form the influence domain polygon, and then to re-triangulate the upper and lower polygons (V_{Ui}, V_{Di}, i=1, 2, … 6) cut by p_1p_2, p_3p_4 and p_5p_6. In each recursive re-triangulation, the empty circle criterion with respect to the bounding points of the polygon is tested, and the constraint segments p_1v_1, v_1v_2, v_2p_2, p_3v_1, v_1p_4, p_5v_2, v_2p_6 is visible. Hence, anyone of the polygons is re-triangulated according to the *constraint empty circle* criterion.

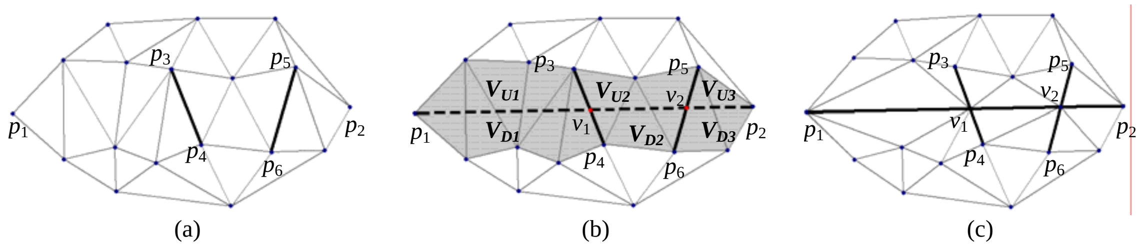


Figure 1. Virtual point insertion and re-triangulation.

(a) original CD-TIN before inserting constrained edge p_1p_2. (b) Influence domain of p_1p_2 and the intersect point between p_1p_2 and other constrained edges. (c) updated CD-TIN after inserting constrained edge p_1p_2.

The interior of the influence domain V_{Ui} and V_{Di} are re-triangulated by using of recursive algorithm (as in figure 2). As for the re-triangulation of V_{U1}, the triangulation of V_{U1} must contain a triangle with constraint segment p_1v_1 being its edges. The recursive algorithm searches a point p'_0 in the bounding of V_{U1} that forms a triangle T (p_1, v_1, p'_0) with vertices p_1, v_1. The triangle meets the constraint empty circle criterion, that is to say, the circumcircle of T (p_1, v_1, p'_0) cannot contain any point of V_{U1} except for its three vertices. After inserting T (p_1, v_1, p'_0), the V_{U1} is divided into two sub-domains, F_E(p_1, p'_1, … p'_0) and F_D(p'_0, p'_i, …, v_1). To apply the algorithm recursively to re-triangulate the sub-domains of F_E and F_D until the entire domain be triangulated.

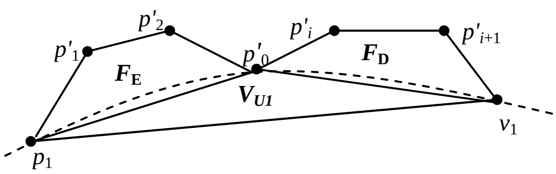


Figure 2. The reconstruct of influence domain

3. ALGORITHM FOR CONSTRAINT POINT DELETION IN CD-TIN

As compared to the point deletion in D-TIN, the deletion of points in CD-TIN must take the constraint edges into consideration. In other words, the point to be deleted refers to both non-constraint point and constraint point. To delete a non-constraint point, there may be constraint edges, which should be kept invariable during flipping edges, in the influence domain boundary of this point. While to delete a constraint point, the constraint segment that contains the point would be deleted while the other endpoint of this segment should kept be invariable.

3.1 Integral Ear Elimination (IEE) Algorithms

The first process of non-constraint point deletion can be shown in figure 3, where p is a non-constraint point and p_1p_2 is a constraint segment. When deleting p, p_1p_2 is one edge of the influence domain boundary of p, and the constraint property of p_1p_2 must be kept in re-triangulating the influence domain. However, the deletion of constraint point p is different as in figure 4, where the constraint segment pp_4 must be deleted and keep point p_4 invariable. Furthermore, a constraint condition is reduced because of the deletion of pp_4, and the diagonal flipping test has to be done according to constraint Delaunay criterion for pp.

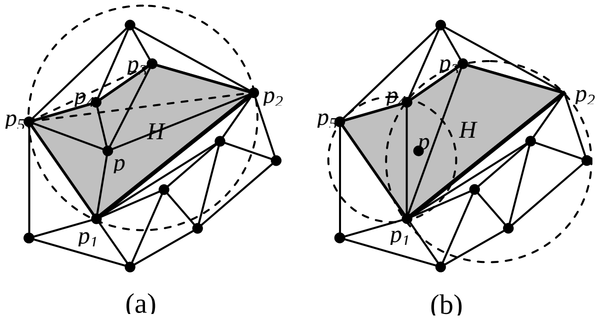


Figure 3. Deletion of non-constraint point p

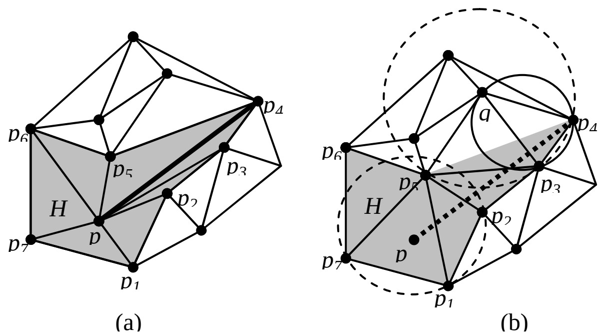


Figure 4. Deletion of constraint point p

3.2 Integral Ear Elimination (IEE) Algorithm

Integral ear elimination (IEE) algorithm must not only meet Delaunay criterion, but also keep topological completeness. The procedures are: 1) to search the influence domain H of point p to be deleted, and to identify whether p is constraint or not; 2) to select three points in the boundary of H as a potential triangle T so as to test if it is an ear or not. If T is an ear, to flip the diagonals of quadrilateral formed by T and p and to delete T; if not, to return to the second step. Note that in the step of flipping diagonal, it is needed to test the ear and its adjoining triangles referring to *constraint empty criterion* if p is constraint.

Let CD-TIN graph be $G = (V \cup \{p\}, A)$, to delete p. The algorithm goes as in the following:

1) to identify p and the other constraint point p' connected with p along the constraint segment (as in Figure 4) if point p to be deleted is a constraint point, else go to next step.

2) to search out all the points $\{p_1, p_2, \dots p_{n-1}, p_n\}$ connected with p to form the influence domain $H = \{p_1, p_2, \dots p_{n-1}, p_n = p_1\}$ of p in anticlockwise order.

3) to check if $T(p_i, p_{i+1}, p_{i+2})$ is an ear. Each triple of points of H is considered as a "potential ear" $T(p_i, p_{i+1}, p_{i+2})$. If T meets any one of the following conditions: a) $D(p_i, p_{i+1}, p_{i+2})$ is negative (means that T forms a re-entrant, not an ear, of the polygon formed by the adjoining of p); b) $D(p_i, p_{i+2}, p)$ is negative (means that T encloses p) and c) $H(p_i, p_{i+1}, p_{i+2}, p_j)$ is positive (means that one or more remaining points p_j of H fall inside the circumcircle of T), T is not an ear and is rejected immediately. If T meets all of these conditions, T is an ear, and may be removed by switching the diagonals pp_{i+1} and p_ip_{i+2} of quadrilateral formed by T and p.

4) if the point p_{i+1} of $T(p_i, p_{i+1}, p_{i+2})$ equal to p', e.g., $p_{i+1}=p'$ (means that $p_{i+1}p'$ is a constraint segment), it is needed to test whether T meets the *constraint empty circle* criterion. That is to say, if another point q of the triangles adjoining to segment p_ip_{i+1} or $p_{i+1}p_{i+2}$ is in the circumcircle of T, it is needed to switch the diagonals of quadrilateral formed by T and q (as in figure 4).

5) to repeat step 3) and step 4) until only three adjoining points in H remained, in which case the three points are merged into one triangle and p is deleted.

4. ALGORITHM FOR CONSTRAINT EDGE DELETION IN CD-TIN

The deletion of constraint edge in CD-TIN is an inverse operation of constraint edge insertion, i.e. to remove all the segments of the constraint edge one by one. If delete a constraint edge with a virtual point, the deletion must meet the *constraint empty circle* criterion, and another constraint edge crossing the virtual point cannot be deleted in considering the topological completeness of CD-TIN. Therefore, the deletion of constraint edge falls into two kinds of situations that with virtual point or without virtual point.

4.1 Constraint Edge Deletion without Virtual Point

If there is no virtual point in constraint edge, the constraint point of this edge can be deleted one by one by using of IEE algorithm (as in figure 5).

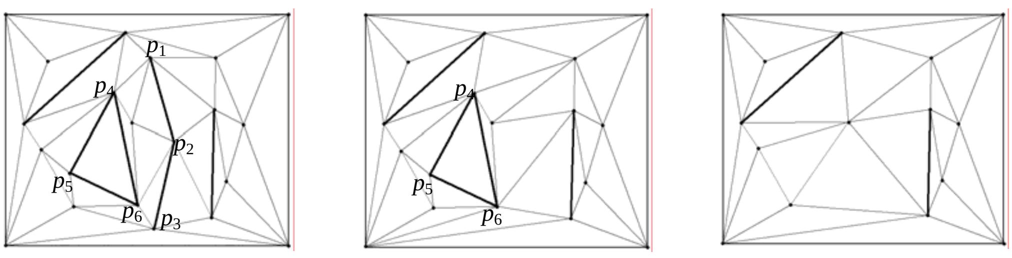


Figure 5. To delete constraint edges without virtual point

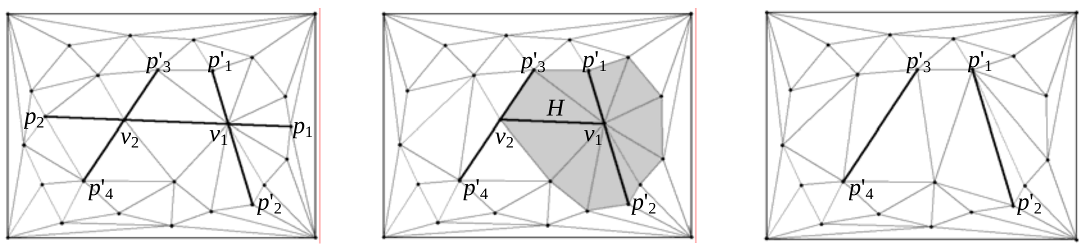
(a) delete constraint broken line $p_1p_2p_3$; (b) delete constraint triangular $T(p_4p_5p_6)$; (c) the reconstructed CD-TIN

4.2 Constraint Edge Deletion with Virtual point

If there is virtual point in the edge, another algorithm called influence domain re-triangulation for virtual point (IDRVP) is presented.

Given a CD-TIN graph $G=\{V, A\cup A'\}$, to delete a constraint edge C with virtual point v. Firstly, the influence domain of v is searched out, which is a polygon H of the boundary of all the triangles connected with v. Secondly, H is divided into two parts, H_L and H_R, by the other constraint edge C_i crossing v. Thirdly, H_L and H_R are re-triangulated individually referring to the constraint empty circle criterion. Finally, the virtual point v is deleted.

As shown in figure 6, p_1p_2 is a constraint edge to be deleted, v_1 and v_2 are two points where $p'_1p'_2$ and $p'_3p'_4$ intersected with p_1p_2. Edge p_1p_2 is composed of three constraint segments p_1v_1, v_1v_2 and v_2p_2. When delete p_1p_2, v_1 and v_2 will be deleted too, but $p'_1p'_2$ and $p'_3p'_4$ are left in the CD-TIN. The detailed operations are: 1) to delete the constraint point p_1 and p_2 by using of IEE algorithm, and the constraint segments p_1v_1 and v_2p_2 will be deleted simultaneously (as in figure 6a); 2) to search the influence domain H of v_1 and to divide it into two parts H_L and H_R by $p'_1p'_2$, and then H_L and H_R are re-triangulated individually, and to delete v_1 (as in figure 6b); 3) to repeat step 2) to delete v_2. Finally, a new CD-TIN is constructed where the constraint edge p_1p_2 is deleted (Figure 6c).



(a) (b) (c)

Figure 6. To delete constraint edges with virtual point

(a) the original CD-TIN and constraint edge p_1p_2 with virtual points v_1 and v_2; (b) the reconstructed CD-TIN after deleting constraint points p_1, p_2; (c) the final CD-TIN after deleting virtual points v_1 and v_2.

The algorithm procedures for the deletion of constraint edges are as in the following:

1) to search all the segments e_i (i=1,2,…n-1) of constraint edge $C = \{p_1p_2 \ldots p_n\}$;

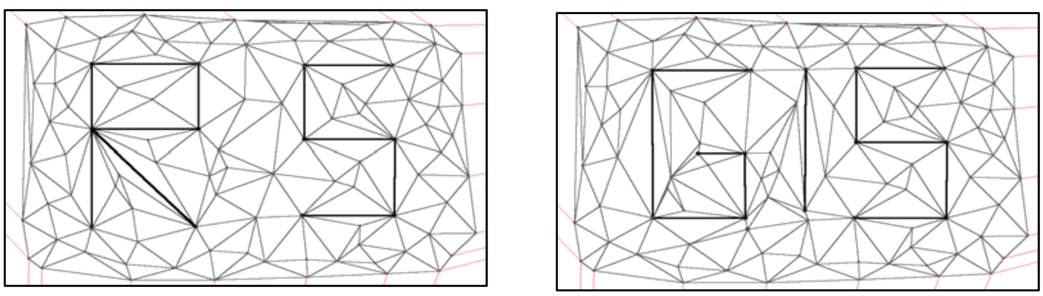
2) to identify the constraint points p_i (i=1,2,…n) and virtual points p'_j (j=1,2,…m) and to store it into two point lists L_C and L_V;

3) to delete all the points p_i(i=0,1,…n) in L_C by of using IEE algorithms;

4) to delete all the points p'_j(j=0,1,…m) in L_V by using of IDRVP algorithms.

5. A CASE FOR ALGORITHMS TESTING

A prototype system with VC++ is developed to test properties of the algorithms. Figure 7 shows the test results for shape reconstruction, where the dynamic deletion and insertion of constraint edges in CD-TIN is illustrated with the letters modification for “RS” to “GIS”.



(a) Shape of letter “RS” as constraint edge (b) Delete “R” and insert “GI” as constraint edge

Figure 7. The shape reconstruction with dynamic deletion and insertion of letters in a CD-TIN

6. CONCLUSION

The CD-TIN is a widely used structure in GIS and computer geometry, and the insertion and deletion operation of points and constraint edges are the basis of CD-TIN dynamic updating. By introducing virtual point, this paper has made improvements and extensions to present updating algorithms for D-TIN and had reached a group of new algorithms in consideration of polymorphism. The new algorithms include the integral ear elimination (IEE) algorithms for constraint point deletion in CD-TIN and the influence domain re-triangulation for virtual point (IDRVP) algorithms for constraint edge deletion in CD-TIN. The test shows that the algorithms are efficient and could be widely applied for the updating of DEM, DTM and other integral vector models of digital city and digital mine.

ACKNOWLEDGMENTS

The author is grateful for the jointly support of the National Outstanding Youth Funds of China (No. 50525414), the National Natural Science Funds of China (No.40571137), National High-Tech Project (No.2006AA12Z216) and the Science Development Plan Awarded by Beijing Municipal Commission of Education Science (No. KM200710028014).

REFERENCES

Aurenhammer, F., 1987, Power diagrams: criterion algorithms and applications. *SIAM Journal of Computing*, 16: 78–96.

Chew, L. P., 1987, Constraint Delaunay Triangulations. In: *Proceedings of the Annual Symposium on Computational Geometry ACM*: 215-222.

Devillers, O., 1999, On deletion in Delaunay triangulations. In: *15th Annual ACM Symposium on Computational Geometry*: 181–188.

Dwyer, R. A., 1987, A fast divide-and-conquer algorithm for constructing delaunay triangulations. *Algorithmica*, 2: 137--151.

Fortune, S., 1987, A Sweepline Algorithm for Voronoi Diagrams. *Algorithmica*, 2:153-174.

Green, E., R. Sibson, 1977, Computing Dirichlet tessellations in the plane, *Comput. J*. 21: 168-173.

Guibas, L., D. Knuth, M. Sharir, 1992, Randomized incremental construction of Delaunay and Voronoi diagrams, *Algorithmica*. 7: 381--413.

Guibas, L., J. Stolfi, 1985, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, *ACM Trans. Graphics*. 4 (2): 75-123.
.

Hao, H. S., Wu LiXin, 2003, 3D-Geosciences Modeling and Visualization Based on Strongly Constraint Delaunay TIN. *Geography and Geo-Information Science*(in Chinese), 19(2): 15-18.

Lee, D. T., Lin H. K., 1986, Genaeralized Delaunay Triangulation for planar graphs. *Discrete Computational Geometry*. 1: 201-217.

Li, Z. L., Zhu Qing, 2000, Digital Elevation Model (in Chinese). Wuhan, Wuhan Surveying And Mapping University Press.

Marc, V., NuHria Pla., 2000, Computing directional constraint Delaunay triangulations. *Computers & Graphics*, 24: 181-190.

Marc, V., Núria Pla, Josep Cotrina., 2002, Regular triangulations of dynamic sets of points. *Computer Aided Geometric Design*, 19: 127-149.

Mostafavi, M. A., Christopher Gold, Maciej Dakowicz, 2003, Deletion and insert operations in Voronoi/Delaunay methods and applications. *Computers & Geosciences*, 23: 523-530.

Ohya, T., M. Iri and K. Murota, 1984, Improvements of the incremental method for the Voronoi diagram with computational comparison of various algorithms, *J. Oper. Res. Soc. Japan* 27: 306-337.

Piegl, L. A., Arnaud M. Richard, 1993, Algorithm and Data Structure for Triangulation Multiply Connected Polygonal Domains [J]. *Computer and Graphics*, 17(5): 563-574.

Preparata, F. P., Shamos M. I., (editors), 1985, *Computational Geometry, An Introduction*. (New Yerk: Springer-Verlag).

Zhu, X. X., 2000, Free curved lines and face modeling technology(in Chinese). Beijing, Science Press, 215-235