

Redrawing a Graph within a Geometric Tolerance ^{*}

Manuel Abellanas¹, Ferran Hurtado², Pedro A. Ramos¹

¹ Universidad Politécnica de Madrid

² Universitat Politècnica de Catalunya

Abstract. In this paper we investigate some applications of the concept of *tolerance* to graph drawing. Given a geometric structure, the tolerance is a measure of how much the set of points can be arbitrarily changed while preserving the structure. Then, if we have a layout of a graph and we want to redraw the graph while preserving the mental map (captured by some proximity graph of the set of nodes), the tolerance of this proximity graph can be a useful tool. We present an optimal $O(n \log n)$ algorithm for computing the tolerance of the Delaunay triangulation of a set of points and propose some variations with applications to interactive environments.

1 Introduction

A basic problem in graph drawing is to create a layout of a graph G , given a combinatorial representation of G , that satisfies some requirements (small number of crossings, displaying symmetries, and so on). A summary of the huge amount of work generated in this area can be found in [6].

A variation on this problem is introduced in [8] and named *Anchored Graph Drawing*. Suppose that we have a layout of a graph G whose nodes have geographic information, such as cities in a country. The objective is to redraw the graph with slight changes on the position of the nodes to make the picture clearer. The first difficulty is the meaning of the expression “slight changes”. The aim is that the changes preserve the *mental map* of the graph, that is, that an observer can recognize both graphs as having the same general aspect. Some heuristic strategies are proposed in [9] among them the *Voronoi Diagram Cluster Busting* (VDCB). The meaning of Cluster Busting is that the objective of this strategy is to break clusters of points that can make the graph difficult to read, and more evenly distribute the nodes in the plane. The idea of VDCB is to move each point to the centroid of its Voronoi region.

In [7] it is suggested that a possible way to capture the mental map of a layout of the graph G is by using some *proximity graph* of the nodes. The task is then trying to move the nodes of G while preserving the proximity graph. The Delaunay triangulation is a specially suitable candidate because it contains strong information about proximity. It is nevertheless true that the problem of

^{*} Partially supported by CICYT grant TIC 93-0747-C02

capture the mental map of a set of points has more than one possible solution and that in some cases the EMST or other proximity graph can be more convenient.

It is proved in [9] that even if we apply the VDCB strategy to one point (and the rest remain fixed), we can cause a linear number of changes in the Delaunay triangulation. For preserving the Delaunay triangulation while moving the points, a different approach is required. If we knew a $\varepsilon > 0$ such that we could move freely any node p inside a disk centered at p with radius ε , then a redrawing strategy could incorporate this constraint and so preserve the *mental map*. For example a modified VDCB would consist of moving each point towards the centroid of its Voronoi region, but stopping when the boundary of the disk is encountered. The supreme of all these ε is called the *tolerance* of the Delaunay triangulation. By moving the points less than the tolerance, we can be sure that the Delaunay triangulation does not change. A similar concept of tolerance can be defined for other proximity graphs also useful for describing the mental map, as well as for many other geometric and combinatorial structures. Actually, tolerance was first introduced in [1], where the aim was to preserve a polygonization of a set S when the set S is continuously changed. A similar study to the one of this paper for other geometric graphs can be found in [10].

Let us consider now a related problem. Let $S = \{p_1, \dots, p_n\}$ be a set of points. We shall call *stability region* of a point p_i (denoted by $SR(p_i)$) to the region where p_i can be continuously moved without producing any change in the Delaunay triangulation (now the rest of the points remain fixed). In graph drawing that corresponds to a situation where a user wants interactively change a layout of a graph but preserving the Delaunay triangulation. By clicking in a node, the stability region of this site can be displayed and then the node moved inside that region. By iterating this process, the configuration of the nodes can be changed but again we are guaranteed that we preserve the Delaunay triangulation.

The paper is organized as follows: in Sect. 2 the tolerance is defined in a general setting, in Sect. 3 we give an optimal algorithm for computing the tolerance of the Delaunay triangulation of a set of points, in Sect. 4 we present an algorithm for computing the stability region of a point and we end in Sect. 5 with some remarks.

2 Structural Tolerance

Let S be a set of points in the plane and consider a *discrete* geometric structure associated to the set. If S is in *general position*, we can move the points arbitrarily inside some neighborhood and be certain that the structure remains *topologically* or *combinatorially* the same.

The *tolerance* of the structure is defined as the supreme of $\varepsilon \geq 0$ such that if each point is moved arbitrarily but not more than ε then the structure does not change.

More formally, let $S = \{p_1, \dots, p_n\}$ and $S' = \{p'_1, \dots, p'_n\}$ be two sets of n labeled points and let us define

$$\delta(S, S') = \max_{i=1, \dots, n} d(p_i, p'_i). \quad (1)$$

It is easy to prove that δ is a distance between labeled sets of n points. We shall say that S' is a ε -perturbation of S if $\delta(S, S') = \varepsilon$.

We define the tolerance of $DT(S)$ as follows:

$$\text{tol}(DT(S)) = \sup \{ \varepsilon \geq 0 \mid DT(S) \sim DT(S') \forall S' \text{ such that } \delta(S, S') \leq \varepsilon \}, \quad (2)$$

where $DT(S) \sim DT(S')$ means that p_i and p_j are neighbors in $DT(S)$ if and only if p'_i and p'_j are neighbors in $DT(S')$.

A general picture of the tolerance was given in [2] where the aim was to describe the concept, its variations and its applications. In this paper, as stated in the introduction, we focus on the tolerance of the Delaunay triangulation because of its applications to graph drawing.

3 Tolerance of the Delaunay Triangulation

We denote by $CH(S)$ the convex hull of the set S . Hereafter, when we say convex hull we mean the boundary of the convex polygon that can be described by the list of its vertices. We shall call extreme points to the points of S that are in $CH(S)$ and interior points to the points of S that are not in $CH(S)$.

We recall the definition of $\text{tol}(DT(S))$ from (2). If some face of $DT(S)$ is not a triangle, the tolerance is clearly zero. Hereafter we assume that this is not the case for S .

We denote by e_{ij} the edge of the Delaunay triangulation between the points p_i and p_j . In order to check possible changes in $DT(S)$, we have to deal with three different types of changes:

- a) If e_{ij} is an interior edge of $DT(S)$, consider the vertices of the two adjacent triangular faces (p_k and p_l in Fig. 1). We denote by α_{ij} the size of the smallest perturbation of the points p_i, p_j, p_k, p_l for which the perturbed points become cocircular.
- b) If the edge e_{rs} belongs to the convex hull, then consider the face that contains the edge e_{rs} (with vertices p_r, p_s and p_t in Fig. 1). In this case, β_{rs} will denote half the distance between p_t and the line determined by p_r and p_s .
- c) Finally, if p_u, p_v and p_w are three consecutive vertices of the convex hull (like in Fig. 1) we denote by γ_v half the distance from the point p_v to the line determined by p_u and p_w .

If we denote by α , β and γ the minimum of α_{ij} , β_{rs} and γ_v over all interior edges, convex hull edges and convex hull vertices respectively, then we have:

Lemma 1. $\text{tol}(DT(S)) = \min \{ \alpha, \beta, \gamma \}$.

Proof. It is important to take notice of α_{ij} is not the tolerance of the edge e_{ij} that we can define in the obvious way: the size of the smallest perturbation of the set that destroys the edge. As it is shown in Fig. 2, the point that destroys the edge e_{ij} can be different from p_k and p_l . Analogous remarks can be made about β_{rs} and δ_v . Nevertheless, we know that changes in $DT(S)$ are always a

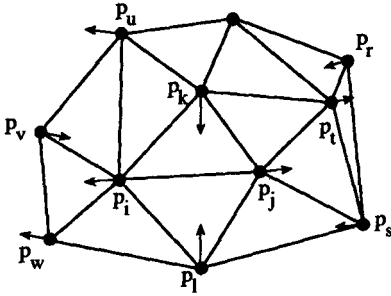


Fig. 1. Different types of changes in the Delaunay triangulation.

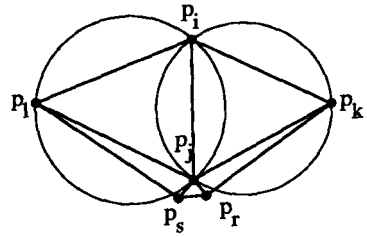


Fig. 2. It is enough to check Delaunay faces.

flip of an interior diagonal or a change in the convex hull. If the first change that can occur is produced in an interior diagonal, then it is computed in Case a (in Fig. 2, before p_r can destroy edge e_{ij} it has to destroy edge e_{jk} and then it is checked). On the contrary, if the first change occurs in the convex hull, we have two possibilities: either a new point appears in the convex hull (and then it is checked in Case b) or a point disappears from the convex hull (and then it is checked in Case c). \square

The algorithm that computes $tol(DT(S))$ can be sketched as follows:

ALGORITHM toLDT(S)

1. Compute $DT(S)$.
2. For each vertex p_i of the convex hull, compute δ_i .
3. For each convex hull edge e_{ij} of $DT(S)$, compute β_{ij} .
4. For each interior edge e_{ij} of $DT(S)$, compute α_{ij} .
5. $tol(DT(S))$ is the minimum over steps 2, 3 and 4.

Step 1 is carried out in $O(n \log n)$ time and clearly steps 2 and 3 take linear time. For completing the analysis, we next show that each α_{ij} can be computed in constant time. As a consequence, the overall running time will be $O(n \log n)$.

Lemma 2. *Let e_{ij} be an interior edge of $DT(S)$ and let p_i, p_j, p_k and p_l the vertices of the two faces that share the edge e_{ij} . Then α_{ij} can be computed in $O(1)$ time.*

Proof. Let us denote by C the annulus of minimum width that contains the points p_i, p_j, p_k and p_l . It is clear that α_{ij} is half the width of C . Then, we have to prove that C can be computed in $O(1)$ time. It is not hard to see that in the solution, the four points have to be on the boundary of C and then we only have seven different options (four with three points on one circle and three with two points on each circle). \square

We can prove that the former algorithm is optimal by providing a lower bound for the computation of $tol(DT(S))$. Details are omitted due to space limitations. We summarize preceding paragraphs with the following result:

Theorem 3. *The algorithm $tol_DT(S)$ computes the tolerance of the Delaunay triangulation of S in $O(n \log n)$ time and this is asymptotically optimal.*

4 Stability Region of a Point

Let us consider a set of points $S = \{p, p_1, \dots, p_n\}$. We define the *global stability region* of p , denoted by $GSR(p)$, as the set of points where p can be moved to without producing any change in the Delaunay triangulation of S . More formally,

$$GSR(p) = \{q \in \mathbb{R}^2 \mid DT(S) \sim DT(S') \text{ for } S' = \{q, p_1, \dots, p_n\}\}. \quad (3)$$

We know that triangular faces of $DT(S)$ correspond to empty circles passing through three points of S and then in order to prevent changes in the Delaunay triangulation of S we are going to check changes in these circles. Three possibilities have to be considered, as two different types of changes are possible in the inner faces and moreover we have to take into account possible changes in the convex hull:

- i) p enters a previously empty circle and then destroys a face of $DT(S)$;
- ii) p leaves a circle determined by three points that after that becomes empty and then creates a face of $DT(S)$;
- iii) p is an extreme point and becomes interior or vice versa.

Of course, changes of type i) and ii) are diagonal flips and when a face is destroyed another one is created and vice versa. The first important observation is that circles involved in changes i) and ii) are always of these two types:

- i) α -circles, that are empty circles corresponding to the triangular face associated to two consecutive Delaunay neighbors of p , plus a third site different from p (like the circle passing through p_1, p_2, p_7 in Fig. 3);
- ii) β -circles, that are circles passing through three consecutive Delaunay neighbors of p in convex position, containing p and no other site (like the circle passing through p_2, p_3, p_4 in Fig. 3).

In order to deal with changes in the convex hull, we introduce a point at ∞ and we consider that all extreme points have as Delaunay neighbor this extra point. Then, when p is a vertex of a face having an edge on $CH(S)$, convex hull edges are generalized α -circles and if p is a vertex of the convex hull the line defined by the two adjacent vertices of the convex hull is a generalized β -circle. Hereafter, when we say α -circles and β -circles we include the generalized case.

Lemma 4. $GSR(p) = (\cap \alpha_i^e) \cap (\cap \beta_j)$ where subscripts range over all α -circles and β -circles of p .

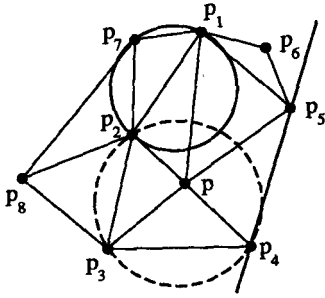


Fig. 3. Examples of α -circle and β -circle.

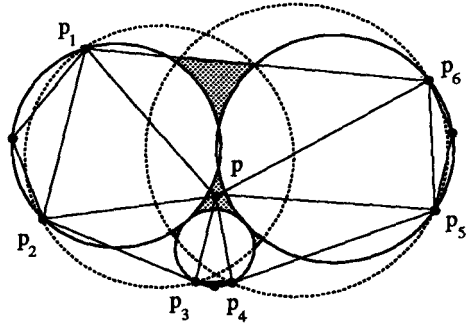


Fig. 4. The global stability region of p can be not connected.

Proof. It is clear that $GSR(p)$ lies in the exterior of all α -circles and in the interior of all β -circles. We are going to prove that the boundary of $GSR(p)$ cannot contain any other element. Suppose that, on the contrary, the boundary of $GSR(p)$ has an element γ that does not belong neither to a α -circle nor to a β -circle. If p cut across γ , a change has to be produced in the Delaunay triangulation of S . Because the only possible changes are diagonal flips, we conclude that either a Delaunay edge starting from p is created (and then γ is part of a α -circle) or a Delaunay edge starting from p is destroyed (and then γ is part of a β -circle). \square

Next observation is that $GSR(p)$ can be not connected as is shown in Fig. 4. In the context of graph layout and recalling the motivation for this problem, it is reasonable to restrict ourselves to movements of p that can be performed continuously without changing the Delaunay triangulation of S . Of course, this can be done by restricting the study to the connected component of $GSR(p)$ containing p that we shall call stability region of p and denote by $SR(p)$. In the rest of this section, we are going to give an upper bound for the combinatorial complexity of $SR(p)$ and an efficient algorithm for its computation.

Let us denote by $Dn(p) = (p_1, \dots, p_k)$ the ordered list of Delaunay neighbors of p . We have k α -circles and at most $\lfloor k/2 \rfloor$ β -circles and then the stability region of p is a single cell in an arrangement of $O(k)$ circles. It follows that the combinatorial complexity of $SR(p)$ is $O(k)$.

Next we are going to give an algorithm for the computation of the stability region. The first step is to determine the α -circles and β -circles for p . We suppose that $DT(S)$ is given (if this is not the case, the Delaunay triangulation would be computed in a first step) and then we have $Dn(p)$. Therefore we are given the set of α -circles. The set of β -circles is a bit more complicated to obtain because we have to obtain the circles determined by three consecutive neighbors of p

that contain just the point p . We can compute the β -circles by exploiting the fact that, because they only contain the point p , they become Delaunay circles if we remove p . Then, if we consider the set $Dn(p)$ and compute $DT(Dn(p))$, the points p_i, p_{i+1}, p_{i+2} (subscripts must be understood modulus k) determine a β -circle if and only if they are consecutive, determine a face in $DT(Dn(p))$ and are in convex position.

Once we have determined the k α -circles and the at most $\lfloor k/2 \rfloor$ β -circles, we only have to compute the cell corresponding to p in the arrangement, that is, the connected component of $(\cap \alpha_i^c) \cap (\cap \beta_j)$ containing p (i ranges over the α -circles and j over the β -circles). We compute $\cap \alpha_i^c$ and $\cap \beta_j$ using one of the algorithms that compute the union and the intersection of a set of disks (see [4],[5]). This can be done in $O(k \log k)$ time. We denote by Δ the boundary of $\cap \beta_j$ and by Γ the boundary of the connected component of $\cap \alpha_i^c$ containing p . If we show that the total number of intersections between Γ and Δ is $O(k)$ then we will be able to compute $SR(p)$ in $O(k \log k)$ time with a standard sweep step.

Let us consider a circular arc γ in Γ (see Fig. 5). It is part of a α -circle passing through three points of S , two consecutive Delaunay neighbors of p and a point q that is not neighbor of p . By construction of β -circles, q is outside all of them. Therefore, when γ cross Δ , it is in the interior of all β -circles and when it intersects Δ again, it goes out a β -circle where it cannot enter again. Then, we have proved that γ intersects Δ in at most two points and then $\Gamma \cap \Delta$ has linear size. We have proved that the stability region of a point p having k Delaunay neighbors can be computed in $O(k \log k)$ time. Moreover, we can repeat this calculation for all the points of S and the total amortized cost will be $O(n \log n)$. We summarize the paragraphs above with the last theorem:

Theorem 5. *Let S be a set of n points. Once we know the Delaunay triangulation of S , the stability region of a point p having k Delaunay neighbors can be computed in $O(k \log k)$ time. The stability regions for all the points of S can be obtained in $O(n \log n)$ overall time.*

5 Concluding Remarks

In this paper we have presented some applications of tolerance to a special class of problems that appears in graph layout, problems where we want to redraw a

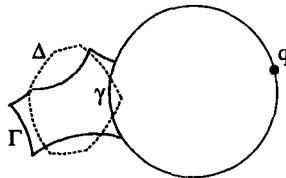


Fig. 5. γ cuts the intersection of the β -circles at most twice.

graph while preserving some structure that captures the mental map of the set of nodes. The relevant case of the Delaunay triangulation has been considered, and we have shown how to compute its tolerance. An step-by-step process for layout modification within the same constraints has also been described.

It is worth noticing that if $DT(S)$ is given, then $tol(DT(S))$ can be computed in linear time. As a particular case, if the points are moved away from their position less than the tolerance, the new tolerance can be computed in linear time.

Half the distance between the closest pair of points of S is obviously a lower bound for $tol(DT(S))$. So the presence of a small cluster of points highly concentrated will result in a very small value for $tol(DT(S))$. It is then reasonable to introduce a concept of *local tolerance* relative not to the full structure but to some subset of edges or faces. These ideas, as well as the variants mentioned in the introduction are developed in [10] for $DT(S)$ and many other structures.

References

1. M. Abellanas, J. García, G. Hernández, F. Hurtado, O. Serra, J. Urrutia: *Updating polygonizations*, Computer Graphics Forum, vol.12, n.3 (1993), pp.143-152.
2. M. Abellanas, F. Hurtado, P. Ramos: *Tolerance of geometric structures*, Proc. 6th Canadian Conference on Computational Geometry (1994).
3. P.J. Agarwal, M. Sharir, S. Toledo: *Applications of parametric searching in geometric optimization*, Proc. 3rd ACM-SIAM Symposium on Discrete Algorithms (1992), pp.72-82.
4. F. Aurenhammer: *Improved algorithms for discs and balls using power diagrams*, Journal of Algorithms 9 (1988), pp.151-161.
5. K.Q. Brown: *Geometric transforms for fast geometric algorithms*, Ph.D. thesis, Rep. CMU-CS-80-101, Dept. of Computer Science, Carnegie-Mellon Univ., Pittsburgh (1980).
6. P. Eades, R. Tamassia: *Algorithms for drawing graphs: An annotated bibliography*, Technical report, Dept. of Computer Science, Brown Univ., Providence, Rhode Island (1989).
7. P. Eades, W. Lai, K. Misue, K. Sugiyama: *Preserving the Mental Map of a Diagram* Research Report IIAS-RR-91-16E, International Institute for Advanced Study of Social Information Science, Fujitsu Laboratories Ltd., 17-25, Shinkamata 1-chome, Ohta-ku, Tokyo 144, Japan, August 1991.
8. K.A. Lyons: *Cluster Busting in Anchored Graph Drawing* Ph.D. thesis, Queen's University (1994).
9. K.A. Lyons, H. Meijer, D. Rappaport: *Properties of the Voronoi Diagram Cluster Buster* Proc. 1993 CAS Conference (CASCON'93) Vol. II, A. Gawman, W.M. Getleman, E. Kidd, P. Larson, J. Slonim, Eds, IBM Canada Ltd. Laboratory Center for Advanced Studies and NRC Canada, Toronto (Canada), October 24-28 1993, pp. 1148-1163.
10. P. Ramos: *Tolerancia de Estructuras Geométricas y Combinatorias* Ph.D. Thesis (in spanish), in preparation.