



Structural tolerance and Delaunay triangulation [☆]

Manuel Abellanas ^{a,1}, Ferran Hurtado ^{b,2}, Pedro A. Ramos ^{c,*}

^a ESCET, Universidad Rey Juan Carlos, 28931 Móstoles, Madrid, Spain

^b Departamento Matemàtica Aplicada II, Univ. Politècnica de Catalunya, 28871 Alcalá de Henares, Madrid, Spain

^c Departamento de Matemáticas, Universidad de Alcalá, Apartado de Correos 20, 28871 Alcalá de Henares, Madrid, Spain

Received 24 February 1999; received in revised form 2 July 1999

Communicated by T. Lengauer

Abstract

In this paper we consider the *tolerance* of a geometric or combinatorial structure associated to a set of points as a measure of how much the set of points can be perturbed while leaving the (topological or combinatorial) structure essentially unchanged. We concentrate on studying the Delaunay triangulation and show that its tolerance can be computed in $O(n)$ time if the triangulation is given as part of the input, while the problem has complexity $\Theta(n \log n)$ if the triangulation is not known. We also study the problem of computing the tolerance of the edges of the triangulation, and show that the tolerance of all the edges can be computed in $O(n \log n)$ time. Finally, we extend our study to some subgraphs of the Delaunay triangulation. © 1999 Published by Elsevier Science B.V. All rights reserved.

Keywords: Computational geometry; Delaunay triangulation; Perturbation

1. Introduction

Let $S = \{p_1, \dots, p_n\}$ be a set of points and consider:

- the Delaunay triangulation of S , $DT(S)$;
- a property: the points are in convex position;
- a line l that bisects the set S .

These examples correspond respectively to a combinatorial structure, a property and a geometric object *topologically* associated to the set of points: if S is in *general position* (the exact meaning of general position depends on the structure or property under con-

sideration), we can move the points arbitrarily inside some neighborhood (perhaps very small) and be certain that $DT(S)$ does not change, the property is still true and the line l is still a bisector.

The *tolerance* of $DT(S)$ is defined as the supremum of all $\varepsilon > 0$ such that if each point p_i is moved arbitrarily but not more than ε then the combinatorial structure $DT(S)$ does not change. The definitions are analogous in situations (b) and (c) and always have the same *conservative* sense.

The tolerance is naturally related to the accuracy of the input of our data, since for large tolerance, errors comparatively small in the input will be irrelevant. On the contrary, if the tolerance is small, even tiny errors in the data can produce different results. However, the tolerance should not be confused with the concept of *algorithmic robustness*, which studies how small roundoff errors can accumulate during dif-

[☆] Research of the second author is partially supported by CICYT grant TIC 95-0957-E, Projecte UPC-PR-9410 and grant DGES PB96-005-C02-02. Research of the first and third authors is partially supported by DGES grant PB96-0005-C02-01.

* Corresponding author. Email: pedro.ramos@alcala.es.

¹ Email: M.Abellanas@escet.urjc.es.

² Email: hurtado@ma2.upc.es.

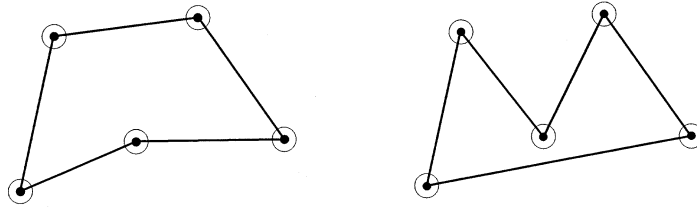


Fig. 1. Two polygonizations of the same set with very different tolerance of their simplicity.

ferent steps of an algorithm and produce a false final result. This is the approach of Guibas et al. [12,13], where the authors define a concept similar to that of tolerance, but from the point of view of algorithmic robustness. The same can be said about the paper [17], by Li and Milenkovic, where the authors propose an algorithm to compute an approximate convex hull taking into account roundoff errors. The main difference is that the tolerance measures possible changes of a combinatorial structure *exactly* associated to a set of points.

The study of tolerance is useful in situations in which we have to deal with perturbations of data, even when we use exact arithmetic in order to avoid roundoff errors, as proposed by Yap in [27], we have to deal with perturbations of data, for instance if we deal with inaccurate data or if the position of the points changes with time (dynamic environment).

In the first case, the tolerance can be interpreted as a measure of how far a given configuration is from being degenerate, because degenerate configurations will have tolerance zero (arbitrarily small perturbations can give different structures).

The problem of the dynamic maintenance of structures associated to a set of moving points is an area that has received some attention in the past. Assuming that the movement of the points is parameterized by algebraic functions of constant degree, it is possible to give bounds on the maximum number of changes in the structure (see for instance [5,9,15,23]). If nothing is known in advance about the movement of the points, the problem is hopeless; however, if an upper bound of the velocity of the points is known, the tolerance gives an interval of time in which the structure does not change and then the passage between the initial and final configurations can be discretized.

Finally, we can observe that in examples (a) and (b) the geometric object under consideration is uniquely defined, but when we have other possibilities as in (c), where we consider a line that bisects a set of points, it would be reasonable to try to select the object with maximum tolerance. In some sense, this would be the best solution to the problem, because it is the most stable if we have errors in the data. This approach leads to *optimization* problems; for example, in [1], the tolerance of the simplicity of a polygon is computed and the following problem is proposed: how should a set of points be polygonized such that the resulting polygon has maximum tolerance (Fig. 1)?

The concept of tolerance, because it is so natural, has already appeared implicitly in different settings, although a systematic study as the one proposed in this paper was missing. For example, when studying Voronoi diagrams of moving points in the plane, Guibas et al. [11] observe that non-degenerate changes in the Voronoi diagram of a set of points are *diagonal flips* produced when four points are cocircular, and they derive some bounds for the number of such *topological changes* in the diagram under suitable assumptions for the movement of the points, although no attempt to compute the size of the perturbation that produces such a change is made. In [4] Arkin et al. consider polygons subjected to an assumption that we can describe now as a fixed lower bound for the tolerance of the simplicity of the polygon. In graph layout, it is sometimes interesting to redraw a graph with slight changes to make the picture clearer while preserving the *mental map* of the diagram; a possible way to do this, suggested by Eades et al. [6] and Lyons, Meijer and Rappaport [18,19], is to preserve some geometric graph, such as the Delaunay triangulation or the minimum spanning tree of the points that correspond to the nodes. The tolerance would then

give bounds for safe perturbations of the nodes. These ideas are explored in a previous version of this paper [3]. In [20] Monma and Suri define the *sensitivity* of a set of points in a way that we can refer to now as the tolerance of the Euclidean minimum spanning tree of the set.

More recently, Weller [25] studies the stability of Voronoi neighborhood under perturbation of the sites focusing on a single edge of the Delaunay triangulation (local tolerance), and Tarasov [24] relates the problem of hardness of roundness computation in higher dimensions to the stability of Voronoi diagrams.

We have mentioned several directions of research that are suggested by the concept of tolerance, but certainly a basic question is how to compute the tolerance of a given structure of a set. In this paper we deal with an example that is important in two ways—it illustrates the techniques involved and the structure considered could not be less important: the Delaunay triangulation. Specifically, we show that the tolerance of $DT(S)$ can be computed in linear time if the triangulation is given as part of the input, while the problem has complexity $\Theta(n \log n)$ if $DT(S)$ is not known. Furthermore, we explore some variations of the problem, such as the problem of computing the tolerance of a single edge of the triangulation, and show that we can compute in $O(n \log n)$ time the tolerance of all the edges of the triangulation. Finally, we study the tolerance of some subgraphs of $DT(S)$. In [20], Monma and Suri show that the *sensitivity* of a set of points (which is exactly the tolerance of its Euclidean minimum spanning tree) can be computed in $O(n \log n)$ time. We show that the tolerance of the all-nearest-neighbors graph can be computed within the same asymptotic complexity. Many other structures, as well as the other aspects we have mentioned before are largely studied in [21].

2. Tolerance of the Delaunay triangulation

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

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

It is easy to prove that δ is a distance between labeled sets of n points. In the special case where the closest

point to each point of S is the corresponding point of S' , it reduces to the Hausdorff distance between the sets, but this is not true in general. We shall say that S' is an ε -perturbation of S if $\delta(S, S') = \varepsilon$.

Let $\mathcal{V}(S)$ denote the *Voronoi diagram* of the set S . The k th order *Voronoi diagram*, denoted $\mathcal{V}_k(S)$, is a partition of the plane into regions such that the k nearest neighbors of the points in a region are the same. We denote by $CH(S)$ the *convex hull* of the set S and by $DT(S)$ its *Delaunay triangulation*. Hereafter, when we say *convex hull*, we mean the boundary of the convex polygon that can be described by the ordered list of its vertices. The edge between the points p_i and p_j is denoted by e_{ij} , and we say that the points p_i and p_j are *Delaunay neighbors* if $e_{ij} \in DT(S)$.

The tolerance of the Delaunay triangulation can be formally defined as follows:

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

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

It is well known that a triangle $p_i p_j p_k$ is a face of $DT(S)$ if and only if the circle through these three points does not contain any other point of S . Topological changes in the Delaunay triangulation are *diagonal flips*: when the point p_l (see Fig. 2) enters the circle passing through the points p_i, p_j and p_k , the edge e_{ij} disappears and is substituted by the edge e'_{kl} . A topological change is also produced when three consecutive vertices of the convex hull become collinear; in order to avoid this special case, we use the *one-point-compactification* approach proposed in [11]: an artificial point p_0 is added at ∞ and

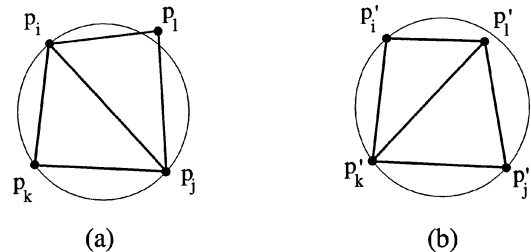


Fig. 2. Example of a diagonal flip.

the extended Delaunay triangulation is obtained by connecting p_0 with the vertices of the convex hull. If we interpret a line as a circle passing through ∞ , then topological changes in the convex hull are *generalized diagonal flips*. Hereafter, except if otherwise indicated, we shall focus on this extended graph, which we also denote by $DT(S)$.

Now, in order to compute the tolerance of $DT(S)$ we only have to detect the smallest perturbation that produces a diagonal flip. Therefore, for two adjacent triangular faces $p_i p_j p_k$ and $p_i p_j p_l$, we want to compute the size of the smallest perturbation that makes the four points cocircular. This is equivalent to compute the circle that minimizes the maximum distance to the four points, or to compute the *annulus of minimum width* containing the points. It has been shown [10,22] that the annulus of minimum width containing a set of points has two points on the inner circle and two points on the outer circle, interlacing angle-wise as seen from the center of the annulus. Therefore, the center will be the intersection of the bisectors of $p_i p_j$ and $p_k p_l$. Now we can conclude that the size of the smallest perturbation making the points $p_i p_j p_k p_l$ cocircular can be computed in $O(1)$ time and, repeating this procedure for every pair of adjacent faces, we can compute the tolerance of the Delaunay triangulation in $O(n)$ time and space if $DT(S)$ is known in advance. If $DT(S)$ is not known, we are going to see that the problem of computing the tolerance has complexity $\Omega(n \log n)$.

To this end, let $N = \{x_1, \dots, x_n\}$ be a set of natural numbers. We are going to reduce the problem of element uniqueness in N , which is known to be $\Omega(n \log n)$ in the algebraic decision tree model (see [26]), to the problem of the tolerance of the Delaunay triangulation of a certain set S , constructed in linear time from the set N . We shall assume that x_1 and x_n are, respectively, the minimum and the maximum of N . The idea is to construct a set having a Delaunay triangulation with tolerance bounded from below by a certain constant if there are no repeated elements, while if these occur, it has tolerance smaller than that constant.

We denote $r = x_n - x_1$ and define

$$a_i = \frac{x_i - x_1}{r}.$$

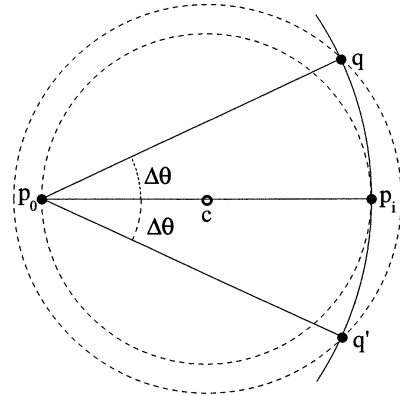


Fig. 3. Worst-case situation in the lower bound.

Consider now the set $S = \{p_0, p_1, \dots, p_n\}$ where p_0 is the origin of the coordinate system and for $i = 1, \dots, n$, p_i has coordinates

$$(x_i, y_i) = r \left(\frac{1 - a_i^2}{1 + a_i^2}, \frac{2a_i}{1 + a_i^2} \right).$$

Assume for the moment that there are no repeated elements in N . Then $DT(S)$ consists of triangles with vertices at p_0 and two consecutive points on the circle. It is not hard to see that the angle between two consecutive points is at least $1/(2r)$ and that the tolerance is bounded from below by three consecutive points on the circle with minimum angular separation as shown in Fig. 3. Then, a straightforward computation shows that, for r big enough,

$$\begin{aligned} \text{tol}(DT(S)) &\geq \frac{1}{2} (d(c, q) - d(c, p_i)) \\ &= \frac{r}{2} \left(\sqrt{5 - 4 \cos \frac{1}{2r}} - 1 \right) \geq \frac{1}{10r}. \end{aligned}$$

If we perturb the abscissas of the points p_1, \dots, p_n by $1/(20r + i)$ (depending on the indices) and denote the resulting set by S' , then in the case of no repeated elements in N we have

$$\text{tol}(DT(S')) \geq \frac{1}{10r} - \frac{1}{20r} = \frac{1}{20r}$$

while if there are repeated elements in N , S' has two points at distance smaller than $1/20r$, and therefore,

$$\text{tol}(DT(S')) \leq \frac{1}{40r},$$

concluding the proof of:

Theorem 1. *The problem of computing the tolerance of the Delaunay triangulation has complexity $\Theta(n \log n)$. Furthermore, if $DT(S)$ is known as part of the input, the tolerance can be computed in linear time.*

If we define the *tolerance of a triangulation* \mathcal{T} of S in the obvious way, namely, as the supremum of all $\varepsilon > 0$ such that \mathcal{T} is a triangulation for every ε -perturbation of S , a natural question would be try to compute the *most tolerant* triangulation of a given set of points. Because it is easily seen that the tolerance of a triangulation is half the smallest height among all the triangles in the triangulation, we conclude that the solution to this problem is the well known *max-min height triangulation*, which can be computed in $O(n^2 \log n)$ time [7].

2.1. Local tolerance

Given an edge $e_{ij} \in DT(S)$, the *tolerance of e_{ij}* is defined as the supremum of all the $\varepsilon > 0$ such that for all S' with $\delta(S, S') \leq \varepsilon$ we have $e_{ij} \in DT(S')$. If d_i denotes the number of Delaunay neighbors of p_i , Weller has shown [25] that the decision problem of whether $tol(e_{ij}) \geq \varepsilon_0$ can be solved in $O(d_i + d_j)$ time and, by using an extension of Megiddo's technique for linear programming, the optimization version of the problem can be solved within the same asymptotic bound. This result is based on the fact that points that destroy the edge e_{ij} are Delaunay neighbors of either p_i or p_j .

The drawback of this algorithm is that if we want to compute the tolerance of all the edges of $DT(S)$, the complexity can grow to quadratic because we sum up the number of Delaunay neighbors on the set of edges. For an example, consider a set $S = \{p_0, p_1, \dots, p_n\}$ where p_0 is the origin of the coordinate system and the rest of the points are arranged along the unit circle (we can slightly perturb the points if we want to assume general position). Then, the Delaunay triangulation contains all the edges between p_0 and the rest of the points. Therefore, we compute the tolerance of *each* edge $p_0 p_i$ in linear time (because the degree of p_0 is n), for a total of quadratic time if we compute the tolerance of all the edges. An alternative approach can overcome this situation:

Theorem 2. *The tolerance of all the edges of $DT(S)$ can be computed in $O(n \log n)$ overall time.*

Proof. We recall again the known fact that if points do not coincide, changes in the Delaunay triangulation are always diagonal flips. We observe that the edge e_{ij} is also destroyed if a point p_k ($k \neq i, j$) is coincident with p_i or p_j , but this can easily be checked in linear time for all the edges of the triangulation if $DT(S)$ is known in advance.

Assume now that the edge e_{ij} is destroyed by a diagonal flip with a perturbation of size δ_{ij} , and let p_k and p_l be the points that produce the change. The flip is produced when the points p'_i, p'_j, p'_k and p'_l are on a common circle \mathcal{C} with center c and radius r and, furthermore, the points p'_i and p'_j interlace angle-wise with the points p'_k and p'_l as seen from the center of the circle (see Fig. 4). Because the change is the diagonal flip that destroys the edge e_{ij} with a perturbation of minimum size, the circle centered at c with radius $r + \delta_{ij}$ cannot contain any other point and thus the region of $\mathcal{V}_4(S)$ corresponding to the points p_i, p_j, p_k and p_l is nonempty. Because $\mathcal{V}_4(S)$ has linear complexity and can be computed in time $O(n \log n)$ [16], we conclude observing that in order to compute the tolerance of all the edges of $DT(S)$, we can explore all the regions of $\mathcal{V}_4(S)$ and, for each region, compute in $O(1)$ the corresponding annulus. \square

It is worth noticing that, if $DT(S)$ is known and its degree is bounded by a constant, the tolerance of all the edges can be computed in linear time with a brute force approach, using the fact that points that

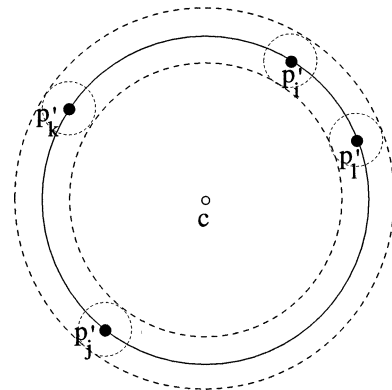


Fig. 4. Illustration of the proof of Theorem 2.

destroy the edge e_{ij} are Delaunay neighbors of either p_i or p_j . It is an open problem to decide whether the tolerance of all the edges can be computed in linear time in the worst case, or if the problem has $\Omega(n \log n)$ complexity even if $DT(S)$ is given as part of the input.

Another natural variation of the concept of local tolerance would be to consider as fixed all but one point, and compute the region where the moving point can be perturbed without producing any changes in the triangulation. It can easily be seen that this problem can be solved in $O(k \log k)$ time, where k is the number of Delaunay neighbors of the moving point, by constructing the arrangement of empty circles defined by consecutive Delaunay neighbors of the moving point.

2.2. Subgraphs of the Delaunay triangulation

The Euclidean minimum spanning tree of S , denoted by $EMST(S)$, is defined as the tree of minimum total length whose vertices are the points of S . Monma and Suri [20] define the *sensitivity* of a set of points in a way that we can refer to now as the tolerance of the Euclidean minimum spanning tree of the set. They show that the tolerance of this graph can be computed in $O(n \log n)$ time. By reduction to the problem of the maximum gap in a set of integers, it is easy to see that this algorithm is asymptotically optimal if $EMST(S)$ is not known, but it is an open problem to decide whether this problem can be solved in linear time if $EMST(S)$ is given as part of the input. In the rest of this section we focus on another subgraph of $DT(S)$, the *all nearest neighbors* graph, which has been used as a tool for collecting proximity information of the set S [14].

Let $nn(p_i)$ denote the nearest neighbor of p_i in the set S . The (directed) graph $ANN(S)$ is defined as follows: p_i is connected to p_j if and only if $p_j = nn(p_i)$. When dealing with inexact data or moving points, the tolerance of this structure is again a useful tool. More generally, we say that $tnn(p_i) = p_k$ if $p_k \neq nn(p_i)$ and there exists $\varepsilon > 0$ such that for all ε -perturbations of S , the nearest neighbor of p'_i is either $nn(p_i)'$ or p'_k . Of course, it is not true that $tnn(p_i)$ is the second nearest neighbor of p_i . Actually, it is not hard to construct examples where $tnn(p_i)$ is the *furthest* neighbor of p_i in S .

Following [20], we observe that given p_i, p_j and p_k with $d(p_i, p_j) < d(p_i, p_k)$, computing the

smallest perturbation such that $d(p'_i, p'_j) = d(p'_i, p'_k)$ is equivalent to computing the smallest $\varepsilon > 0$ such that the branch of the hyperbola $d(x, p_k) - d(x, p_j) = 2\varepsilon$ is tangent to the circle $d(x, p_i) = \varepsilon$. Hereafter, we assume that this can be done in $O(1)$ time.

In order to collect more proximity information about the set S , we generalize $ANN(S)$ to $ANNT(S)$ by connecting the point p_i to $nn(p_i)$ and to $tnn(p_i)$. From the observation in the previous paragraph, it is obvious that the tolerance of $ANN(S)$ can be computed in linear time if we have $ANNT(S)$. Now we concentrate on how to compute this graph.

Theorem 3. *If $DT(S)$ is known, $ANNT(S)$ can be computed in $O(n)$ time.*

Proof. We claim that $tnn(p_i)$ is connected in $DT(S)$ either to p_i or to $nn(p_i)$. In order to prove the claim, let ε be the size of the smallest perturbation such that $d(p'_i, nn(p_i)') = d(p'_i, tnn(p_i)')$. We observe that the only points of S inside the circle centered at p'_i with radius $d(p_i, nn(p_i)) + \varepsilon$ are $p_i, nn(p_i)$ and $tnn(p_i)$ (see Fig. 5). Therefore, there exists a circle passing through $tnn(p_i)$ and containing only the points $nn(p_i)$ and p_i which proves that either $nn(p_i)$ or p_i are Delaunay neighbors of $tnn(p_i)$.

Now, in order to conclude the proof, we only have to observe that $ANNT(S)$ can be constructed from $DT(S)$ in linear time. This holds because each point of S is the nearest neighbor of at most six points of S , and therefore the extra work of computing $tnn(p_i)$ for each $p_i \in S$ is proportional to the complexity of $DT(S)$. \square

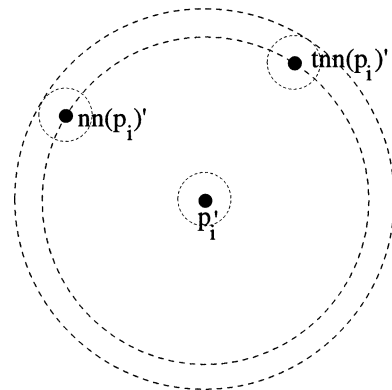


Fig. 5. Illustration of Lemma 3.

3. Concluding remarks

In this paper we have studied the concept of *tolerance* of a geometric or combinatorial structure. Because it is so natural, it had already appeared implicitly in different settings, but a systematic study as the one proposed in this paper was never carried out. We have shown that the tolerance of the Delaunay triangulation can be computed in linear time if the triangulation is given as part of the input, while the problem has $\Theta(n \log n)$ complexity if the triangulation is not known. We have also shown that the tolerance of all the edges of the triangulation can be computed within the same asymptotic bound, but it is an open problem to decide if this can be improved to linear time if the triangulation is given. Finally, we have studied the tolerance of the all nearest neighbors graph of a set of points.

From the point of view of combinatorial optimization, the concept of tolerance also suggests a number of problems. If we define the tolerance of a convex polygon in the obvious way and interpret it as a *measure of convexity*, then the following problem, proposed in [2], comes to mind naturally: what is the *most convex* polygon that can be enclosed in the unit circle? The natural conjecture is the regular n -gon, and this has been shown to be true by Evans in [8]. In a similar way, the following problem can be proposed: what is the n point configuration that can be enclosed in the unit circle that has a Delaunay triangulation of maximum tolerance?

References

- [1] M. Abellanas, J. García, G. Hernández, F. Hurtado, O. Serra, J. Urrutia, Updating polygonizations, *Computer Graphics Forum* 12 (3) (1993) 143–152.
- [2] M. Abellanas, F. Hurtado, P. Ramos, Tolerance of geometric structures, in: Proc. 6th Canadian Conference on Computational Geometry, 1994, pp. 250–255.
- [3] M. Abellanas, F. Hurtado, P. Ramos, Redrawing a graph within a geometric tolerance, in: Proc. Graph Drawing'94, Lecture Notes in Computer Sci., Vol. 894, Springer, Berlin, 1995, pp. 246–253.
- [4] E.M. Arkin, P. Belleville, J.S.B. Mitchell, D. Mount, K. Romanik, S. Salzberg, D. Souvaine, Testing simple polygons, *Comput. Geom. Theory Appl.* 8 (1997) 97–114.
- [5] M.J. Atallah, Some dynamic Computational Geometry problems, *Comput. Math. Appl.* 11 (1985) 1171–1181.
- [6] P. Eades, W. Lai, K. Misue, K. Sugiyama, Preserving the mental map of a diagram, in: Proc. of Compugraphics'91, 1991, pp. 24–33.
- [7] H. Edelsbrunner, T.S. Tan, R. Waupotitsch, $O(N^2 \log N)$ time algorithm for the minmax angle triangulation, *SIAM J. Sci. Statist. Comput.* 13 (1992) 994–1008.
- [8] W. Evans, Regular polygons are most tolerant, in: Proc. 7th Canadian Conference on Computational Geometry, 1995, pp. 109–113.
- [9] J.J. Fu, R.C.T. Lee, Voronoi diagrams of moving points in the plane, *Internat. J. Comput. Geom. Appl.* 1 (1991) 23–32.
- [10] J. García, P.A. Ramos, Fitting a set of points by a circle, in: Proc. 13th Annual ACM Symposium on Computational Geometry, 1997, pp. 139–146.
- [11] L.J. Guibas, J.S.B. Mitchell, T. Roos, Voronoi diagrams of moving points in the plane, in: Proc. 17th Internat. Workshop Graph-Theoretical Concepts of Computer Science, Lecture Notes in Computer Sci., Vol. 570, Springer, Berlin, 1991, pp. 113–125.
- [12] L. Guibas, D. Salesin, J. Stolfi, Epsilon-geometry: Building robust algorithms for imprecise computation, in: Proc. 5th Annual ACM Symposium on Computational Geometry, 1989, pp. 208–217.
- [13] L. Guibas, D. Salesin, J. Stolfi, Constructing strongly convex approximate hulls with inaccurate primitives, *Algorithmica* 9 (1993) 534–560.
- [14] J.W. Jaromczyk, G.T. Toussaint, Relative neighborhood graphs and their relatives, *Proc. IEEE* 80 (1992) 1502–1517.
- [15] N. Katoh, T. Tokuyama, K. Iwano, On minimum and maximum spanning trees of linearly moving points, *Discrete Comput. Geom.* 13 (1995) 161–176.
- [16] D.T. Lee, On k -nearest neighbor Voronoi diagram in the plane, *IEEE Trans. Comput.* 31 (1977) 478–487.
- [17] Z. Li, V. Milenkovic, Constructing strongly convex hulls using exact or rounded arithmetic, *Algorithmica* 8 (1992) 345–364.
- [18] K.A. Lyons, Cluster busting in anchored graph drawing, Ph.D. thesis, Queen's University, 1994.
- [19] K.A. Lyons, H. Meijer, D. Rappaport, Properties of the Voronoi diagram cluster buster, in: Proc. 1993 CAS Conference (CAS-CON'93), Vol. 2, 1993, pp. 1148–1163.
- [20] C. Monma, S. Suri, Transitions in geometric minimum spanning trees, *Discrete Comput. Geom.* 8 (1992) 265–293.
- [21] P.A. Ramos, Tolerancia de estructuras geométricas y combinatorias, Ph.D. thesis, Universidad Politécnica de Madrid, 1995.
- [22] T.J. Rivlin, Approximation by circles, *Computing* 21 (1979) 93–104.
- [23] T. Roos, Tighter bounds for Voronoi diagrams of moving points, in: Proc. 5th Canadian Conference on Computational Geometry, 1993, pp. 358–363.
- [24] S. Tarasov, On hardness of roundness calculation, in: Proc. 9th Canadian Conference on Computational Geometry, 1997, pp. 175–180.
- [25] F. Weller, Stability of Voronoi neighborhood under perturbation of the sites, in: Proc. 9th Canadian Conference on Computational Geometry, 1997, pp. 251–256.
- [26] A.C. Yao, Lower bounds for algebraic computation trees with integer inputs, *SIAM J. Comput.* 20 (1991) 655–668.
- [27] C.K. Yap, Low level issues in Computational Geometry, in: Proc. 5th Canadian Conference on Computational Geometry, 1993, pp. 405–419.