A Note on Checking Proximity Graphs*

(Abstract)

Giuseppe Liotta[†]

The problem of efficiently computing subgraphs of the Delaunay diagram of a set of points is the subject of a rich body of literature in computational geometry. One of the first papers is due to Cheriton and Tarjan [CT76], who show a $O(n)$-time algorithm to compute an Euclidean minimum spanning tree from a Delaunay diagram with n vertices. The Gabriel graph can be computed from the Delaunay diagram in $O(n)$-time by using the algorithm of Matula and Sokal [MS80]. The first $O(n \log n)$-time algorithm to compute a relative neighborhood graph from a Delaunay diagram is due to Supowit [Sup83], who improves the previous $O(n^2)$-time bound showed by Toussaint [Tou80]. Jaromczyk and Kowaluk [JK87] show that the relative neighborhood graph can be computed from the Delaunay diagram in $O(n\alpha(n,n))$-time, where $\alpha(\cdot)$ is the inverse of the Ackermann's function. The bound is reduced to $O(n)$ by Jaromczyk, Kowaluk, and Yao [JKY]. A different proof of this last result is given by Lingas [Lin94]. For a survey on algorithms that compute proximity drawings see [JT92].

Existing algorithms, however, often accomplish the asymptotically optimal efficiency at the expenses of simplicity; also they often rely on simplifying assumptions on the input configuration (no three points are collinear and no four points are co-circular) that make them fail when implemented in practice. For example, the linear-time algorithm of [JKY] is based on complex UNION-FIND data structures.

On the other hand, issues of robustness, experimentation, and implementation have become crucial to the development of geometric algorithms. In this context, the study of *geometric checkers*, i.e. algorithms that verify the correctness of the geometric output computed by a particular code execution, has been recently receiving increasing attention (see,e.g.,[BK95, MNS$^+$96, DLTP97]). The development of geometric checkers is motivated by the intrinsic structural complexity of implementing most geometric algorithms and by the expectation that it is usually easier to verify the properties of a geometric structure than the correctness of the code producing it. The study of geometric checkers is further justified by the large availability of uncertified geometric software over the Internet and by the emerging client-server distributed models of geometric computing over the web (see, e.g. [BCLT95]). Also, there is an increasing consensus about developing robust geometric libraries that are equipped with checking facilities (see, e.g., LEDA [MN95, MN96]). Geometric checkers for Delaunay triangulations, convex hulls, Voronoi diagrams, planar subdivisions, and convex polytopes have been developed in [BK95, MNS$^+$96, DLTP97].

In this note, we focus on the problem of robust computing and checking proximity graphs, where proximity is defined in terms of circle-based β-neighborhood [KR85, Rad88]. In the context of methodologies intended to confer robustness to geometric computation, we analyze the efficiency of the proposed algorithms within a realistic model, in which the arithmetic precision that is needed to carry out exact computation is relevant. To this aim, we adopt the concept of *degree* of a geometric algorithm [LPT97], which characterizes, up to a small additive constant, the number of bits needed by a geometric algorithm to be error free, i.e. not affected by degenerate configurations of the input and not affected by errors in the flow of control due to round-off approximations.

The main contributions are listed below.

- We show lower bounds on the degree of checking and computing the β-skeleton graph of a set of points on the plane under the Euclidean metric ($\infty \geq \beta \geq 1$). In order to achieve the lower bounds, we exploit techniques in [LPT97], based on analyzing the intrinsic complexity of the geometric predicates that must be evaluated by any algorithm that correctly solves the problem. We provide the following result: The problem of checking a Gabriel graph (i.e. the β-skeleton graph for $\beta = 1$) has degree at least 2, while the problem of checking a circle-based β-skeleton graph has degree at least 4 for values of β such that $\infty \geq \beta > 1$.

*Research supported in part by the ESPRIT LTR Project no. 20244 - ALCOM-IT.

†Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", via Salaria 113, 00198 Roma, Italy. liotta@dis.uniroma1.it.

- We show a new characterization of circle-based β-skeleton graphs ($\infty \geq \beta \geq 1$) in terms of the underlying Delaunay triangulation. For the value $\beta = 1$, the characterization generalizes a result proved in [BLL96] to solve a graph drawing problem. We show two different applications of our characterization. As a first application, we present robust algorithms for extracting Gabriel graphs and circle-based β-skeleton graphs (where $\infty \geq \beta > 1$) from Delaunay diagrams. The algorithms are optimal with respect to both asymptotic number of operations and degree. The algorithm for computing Gabriel graphs outperforms in terms of degree the well-known algorithm in [MS80], that is usually considered as "the algorithm" for this problem. The algorithm for circle-based β-skeleton graphs is alternative to the one in [KR85] and its time complexity is affected by a lower constant factor.
- As a second application of the characterization, we present linear time checkers for Gabriel and circle-based β-skeleton graphs ($\infty \geq \beta > 1$) that have degree 4, which is the optimal degree for circle-based β-skeleton graphs. The input of our checkers consists of: A two-dimensional geometric graph Γ with its circular ordering of the edges around the vertices, the parameter β ($\infty \geq \beta > 1$), the set S of point for which Γ is claimed to be the β-skeleton graph, and a subdivision of $D(S)$ which is claimed to be the Delaunay diagram of S. The assumption that $D(S)$ is part of the input is justified by the fact that all efficient algorithms that compute β-skeleton graphs, (for $\infty \geq \beta \geq 1$) compute first $D(S)$ and then construct the wanted proximity graph by deleting edges of $D(S)$. Supporting a checker with an additional input in order to simplify the output verification is advocated by the program checking paradigm pioneered by Blum and Kannan [BK95].

Beside the above results, open problems about checking and computing proximity graphs are discussed, and trade-offs between computational efficiency and degree are shown.

References

[BCLT95] J. E. Baker, I. F. Cruz, G. Liotta and R. Tamassia, *A New Model for Algorithm Animation Over the WWW*, ACM Comput. Surv., 27, no. 4, 568–572, 1995.

[BK95] M. Blum and S. Kannan, *Designing Programs that Check Their Work*, J. ACM, 42, no. 1, 269–291, Jan. 1995.

[BLL96] P. Bose, W. Lenhart and G. Liotta, *Characterizing Proximity Trees*, Algorithmica, 16, 83–110, 1996, (special issue on Graph Drawing, edited by G. Di Battista and R. Tamassia).

[CT76] D. Cheriton and R. E. Tarjan, *Finding minimum spanning trees*, SIAM J. Comput., 5, 724–742, 1976.

[DLTP97] O. Devillers, G. Liotta, R. Tamassia and F. Preparata, *Cecking the Convexity of Polytopes and the Planarity of Subdivisions*, In Algorithms and Data Structures (Proc. WADS 97), vol. 1272 of *Lecture Notes Comput. Sci.*, pp. 186–199. Springer-Verlag, 1997.

[JK87] J. W. Jaromczyk and M. Kowaluk, *A note on relative neighborhood graphs*, In Proc. 3rd Annu. ACM Sympos. Comput. Geom., pp. 233–241, 1987.

[JKY] J. Jaromczyk, M. Kowaluk and F. Yao, *An Optimal Algorithm for constructing β-skeletons in the L_p metric*, SIAM J. On Computing, to appear.

[JT92] J. W. Jaromczyk and G. T. Toussaint, *Relative Neighborhood Graphs and Their Relatives*, Proc. IEEE, 80, no. 9, 1502–1517, Sept. 1992.

[KR85] D. G. Kirkpatrick and J. D. Radke, *A framework for computational morphology*, In G. T. Toussaint, editor, Computational Geometry, pp. 217–248. North-Holland, Amsterdam, Netherlands, 1985.

[Lin94] A. Lingas, *A Linear-time Construction of the Relative Neighborhood Graph from the Delaunay triangulation*, Comput. Geom. Theory Appl., 4, 199–208, 1994.

[LPT97] G. Liotta, F. P. Preparata and R. Tamassia, *Robust Proximity Queries: an Illustration of Degree-Driven Algorithm Design*, In Proc. 13th Annu. ACM Sympos. Comput. Geom., pp. 156–165, 1997.

[MN95] K. Mehlhorn and S. Näher, *LEDA: a platform for combinatorial and geometric computing*, Commun. ACM, 38, 96–102, 1995.

[MN96] K. Mehlhorn and S. Näher, *Checking Geometric Structures*, Dec. 1996, Manual.

[MNS+96] K. Mehlhorn *et al.*, *Checking Geometric Programs or Verification of Geometric Structures*, In Proc. 12th Annu. ACM Sympos. Comput. Geom., pp. 159–165, 1996.

[MS80] D. W. Matula and R. R. Sokal, *Properties of Gabriel graphs relevant to geographic variation research and clustering of points in the plane*, Geogr. Anal., 12, 205–222, 1980.

[Rad88] J. D. Radke, *On the shape of a set of points*, In G. T. Toussaint, editor, Computational Morphology, pp. 105–136. North-Holland, Amsterdam, Netherlands, 1988.

[Sup83] K. J. Supowit, *The relative neighborhood graph with an application to minimum spanning trees*, J. ACM, 30, 428–448, 1983.

[Tou80] G. T. Toussaint, *The relative neighbourhood graph of a finite planar set*, Pattern Recogn., 12, 261–268, 1980.