Quadrangulations of Planar Sets

Godfried Toussaint
School of Computer Science
McGill University
Montreal, CANADA

ABSTRACT

Given a set S such as a polygon or a set of points, a quadrangulation of S is a partition of the interior of S, if S is a polygon, or the interior of the convex hull of S, if S is a set of points, into quadrangles (quadrilaterals) obtained by inserting edges between pairs of points (diagonals between vertices of the polygon) such that the edges intersect each other only at their end points. Not all polygons or sets of points admit quadrangulations, even when the quadrangles are not required to be convex (convex quadrangulations). In this paper we briefly survey some recent results concerning the characterization of those planar sets that always admit quadrangulations (convex and non-convex) as well as some related computational problems.

1. Introduction

In the field of computational geometry a *triangulation* of a finite planar set such as a set of points, line segments or polygon, is a well studied structure [O'R94], [PS85]. For one thing, a triangulation always exists and for another, triangulations have a wide variety of applications in practice. Two of the most important applications in engineering are (1) mesh generation for finite element methods [SNTM], [ZT89], [Ho88], [SP89], [BE92] and (2) scattered data interpolation [QS90], [LS94]. Triangulations have also been heavily applied to other application areas such as pattern recognition, computer graphics, solid modelling and geographic information systems [AHMS], [BS94], [OBS92], [SS88], [To86], [Wa92], [WA86], [Yo75]. However, in the study of finite element methods and scattered data interpolation, it has recently been shown that quadrangulations of point sets may be more desirable objects than triangulations. A quadrangulation of a set of points S is a planar subdivision whose vertices are the points of S, whose outer face is the convex hull of S, and every face of the subdivision (except possibly the outer face) is a quadrilateral. Compared to triangulations, quadrangulations have received scant attention in the computational geometry literature.

The classical problem in scattered bivariate data interpolation can be stated as follows. Given a set $V = (v_1, v_2,..., v_n)$ of n points in the plane along with an elevation z_i, $i = 1,2,..., n$ associated with each point v_i, determine a function f such that $f(v_i) = z_i$, for all i. There is a large body of literature on this subject describing a variety of methods that yield functions with different properties. Most of these methods start with a triangulation of V which is subsequently refined in some way [Wa92]. Since a triangulation of a point set always exists there is no problem starting in this way. Recently Lai [La94] and Lai & Schumaker [LS94] showed that there are certain advantages that can be obtained by starting with a quadrangulation rather than a triangulation. They also gave algorithms for computing such interpolation functions starting from quadrangulations.

The above applications provide motivation for the study of quadrangulations of point sets from the computational geometry point of view. We remark that quadrangulations of *polygons*, on the other hand, have been investigated in the computational geometry literature for some time, mostly in the context of guarding or illumination problems. In this paper we briefly survey some known results concerning the characterization of those planar sets that always admit quadrangulations (convex and non-convex) as well as some related computational problems. Furthermore we concentrate on the recent results on (1) quadrangulating sets of points rather than polygons and (2) "converting" triangulations to quadrangulations.

2. Quadrangulating Orthogonal Polygons

One of the earliest results on quadrangulations concerns orthogonal polygons, i.e., polygons whose sides are parallel to two orthogonal axes. The motivation here was not the quadrangulation itself but rather its application to locating $n/4$ guards (lights) to cover (illuminate) the interior of the polygon. Sack and Toussaint [ST81] showed that a star-shaped orthogonal polygon of n vertices can always be decomposed into *convex* quadrangles in $O(n)$ time. Kahn, Klawe and Kleitman [KKK83] proved this result for arbitrary simple orthogonal polygons. However, their existential proof did not lead to an algorithm for quadrangulating the polygon. Later, Sack and Toussaint [Sa82], [ST88] and independently Lubiw [Lu85] obtained constructive proofs of the [KKK83] result that led to an $O(n \log n)$ time algorithm. Lubiw [Lu85] also showed under a fairly general computational model that $\Omega(n \log n)$ time is a lower bound on the complexity of this problem. Some work has also been done on computing minimum "ink" quadrangulations of simple orthogonal polygons. In this setting one is interested in the convex quadrangulation whose total length (sum of diagonal lengths) is a minimum. Independently, Keil & Sack [KS85] and Lubiw [Lu85] showed this could be done in $O(n^4)$ time and $O(n^2)$ space. A very special case of convex quadrangulations, of great interest in VLSI, is that of decomposing an orthogonal polygon into *rectangles*. For example, Lipski et al. [LLLMP79] and Ohtsuki [Oh82] give polynomial-time algorithms for partitioning orthogonal polygonal regions into the minimum number of rectangles.

3. Quadrangulating Simple Polygons

Whereas an orthogonal polygon always admits a *convex* quadrangulation, an arbitrary simple polygon does not necessarily admit a quadrangulation, even if convexity of the resulting quadrangles is not a requirement. In fact, Lubiw [Lu85] has shown that the problem of deciding whether a polygon with holes admits a quadrangulation is *NP*-complete. On the other hand, if we are allowed to add new points that act as vertices (usually called Steiner points) then, depending on how many such points we are allowed to add, we may always obtain even convex quadrangulations. However, it is not difficult to construct polygons that require $\Omega(n)$ Steiner points in order to complete a quadrangulation. A transparently simple proof and algorithm for obtaining a convex quadrangulation of a simple polygon is due to Mark de Berg [ELOSU] His algorithm first triangulates the polygon and is illustrated in Figure 1, which is self-explanatory. This method however uses many Stainer points to yield many quadrilaterals. Inspired by de Berg's method Everett et al. [ELOSU] have shown that a simple polygon can always be quadrangulated into $5(n-2)/3$ strictly convex quadrangles and that n-2 quadrangles are sometimes necessary. For polygons with holes, they showed that a polygon on n vertices with h holes can always be decomposed into $8(n+2h-2)/3$ strictly convex quadrangles.

Some work has also been done on computing minimum "ink" quadrangulations of simple polygons. Conn & O'Rourke [CO90] showed this could be done in $O(n^3 \log n)$ time and $O(n^3)$ space. To close this section we mention a special case of convex quadrangulations of polygons that has also been studied before and this concerns the problem of decomposing a polygon into *trapezoids*. In fact, many polygon triangulation algorithms start out by first obtaining a trapezoidization of the polygon and subsequently converting this trapezoidization

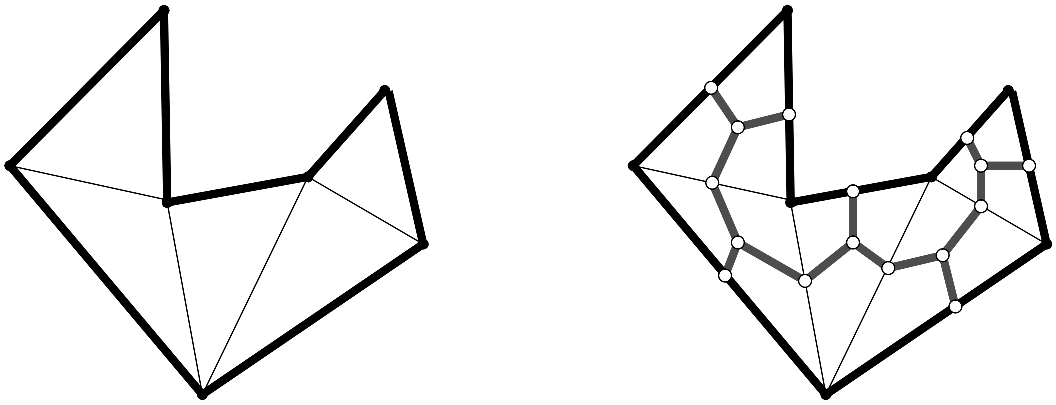

Fig. 1 Example of de Berg's construction of a convex quadrangulation from an arbitrary triangulation of a polygon.

into a triangulation [FM84]. Here the trapezoidization is merely a step towards another goal. Furthermore, we should point out that in the work on trapezoids, triangles are allowed as degenerate trapezoids. On the other hand, trapezoidizations are used as the main goal in electron-beam lithography systems [SS79] where subsequent processing time is proportional to the number of trapezoids in the decomposition. Asano, Asano and Imai [AAI86] have shown that a partition of a polygonal region into the minimum number of trapezoids can be obtained in $O(n^2)$ time.

4. Quadrangulating Point Sets

One characterization of those sets of points that admit a quadrangulation is via *matchings* [RRT95]. A set S of points (not all on a line) admits a quadrangulation if, and only if, there exists a triangulation of S such that its dual graph contains a *perfect* matching. Of course this characterization is not useful for obtaining algorithms to quadrangulate S because there are too many triangulations to enumerate. On the other hand, for the problem of converting triangulations to quadrangulations by only deleting edges in the triangulation, this characterization is most useful as it allows us to apply the known results on matchings to solve a variety of optimization problems concerning quadrangulations [RRT95].

A much more useful characterization of those sets of points that admit a quadrangulation was obtained by Bose & Toussaint [BT95]. They show that S admits a quadrangulation if and only if S does not have an odd number of extreme points (vertices on the convex hull of S). If S admits a quadrangulation, they present an algorithm that computes a quadrangulation of S in $O(n \log n)$ time. Furthermore, their algorithm can handle the presence of collinear points with trivial modifications. If S does *not* admit a quadrangulation, then their algorithm can quadrangulate S with the addition of no more than *one* Steiner point, which is optimal. Their algorithm is conceptually simple, but to achieve the $O(n \log n)$ time complexity, they need to use some complicated data structures. However, from the conceptual description of the algorithm, a very simple $O(n^2)$ time algorithm is implied which may be more desirable from a practical point of view than the $O(n \log n)$ algorithm. They also point out an $\Omega(n \log n)$ time lower bound for the problem. Finally, their results imply that a set of points S admits a k-angulation for any k with the addition of at most k-3 Steiner points. A k-angulation of a set of points S is a planar subdivision whose vertices are the points of S, whose outer face is the convex hull of S, and every face of the subdivision (except possibly the outer face) is a simple polygon with k vertices. Below we give a brief description of their method.

Their existence proof immediately implies the following Sequential Insertion (SI) Algorithm. Compute the convex hull of S in $O(n \log n)$ time with any of several algorithms available (see [O'R94]). If the number of convex hull vertices is even, partition the convex hull into quadrilaterals in $O(n)$ time by joining one vertex to every other vertex of the polygon in a clockwise fashion. Subsequently, the remaining points are inserted one at a time. The insertion stage of the algorithm is the crucial part of the entire procedure. The obvious method of inserting the points results in an $O(n^2)$ time algorithm. However, a variety of algorithms are available in the literature for point insertion [BJM94]. The main difficulty in obtaining efficient implementations of the SI algorithm is determined by the fact that the data structure that supports fast point location must be dynamic so that subsequent point locations after an insertion remain efficient. If the quadrangulation is maintained as a triangulation one can exploit the additional structure over arbitrary planar subdivisions. Furthermore in our context we do not need fully dynamic algorithms since we are not interested in deletions. If we demand $O(n)$ storage space then the best relevant algorithms available, from the worst-case complexity point of view, are the algorithms of Preparata & Tamassia [PT89] and Cheng & Janardan [CJ92]. The former algorithm has both a point location query time and an insertion (update) time of $O(\log^2 n)$. The latter algorithm is slightly better because it has a point location query time of $O(\log^2 n)$ but only an $O(\log n)$ insertion time. However, in the present context every point location query is followed by an insertion and hence the worst-case complexities of both algorithms are the same. Therefore the Sequential Insertion (SI) Algorithm for quadrangulating S runs in worst-case time $O(n \log^2 n)$ and uses $O(n)$ storage space when implemented in this manner. This approach yields a rather complicated algorithm which also has $O(n \log^2 n)$ expected time complexity. However, by embedding the quadrangulation in a triangulation (as above) but using a very simple randomized triangulation algorithm [AHMS], one can obtain a very simple quadrangulation algorithm with $O(n^2)$ worst-case and $O(n \log n)$ expected time complexities.

Another implementation of the SI Algorithm is to use a sweep-line approach (see [O'R94] or [PS85] for examples of sweep-line algorithms). Once the convex hull has been partitioned into quadrilaterals (say $Q_1,..., Q_j$), determine into which quadrilaterals Q_i, the remaining points fall. A simple sweep-line can be used to quadrangulate the set of points lying inside each Q_i, $i = 1,..., j$ in $O(n_i \log n_i)$ time where n_i is the number of points inside Q_i. Therefore the total complexity of the SI algorithm implemented in this way is $O(n \log n)$.

From a theoretical computational complexity point of view, the SI algorithm is optimal. However it has its drawbacks. For one thing, the SI algorithm with random insertion of the points, fails if points are not in general position, although Joe Mitchell has shown that inserting the points using a sweep-line approach, allows a non-trivial modification of the algorithm that can handle collinearities. The main drawback however, as Figure 2 illustrates, of the sequential insertion algorithm is that it yields quadrangulations that are not desirable in practice since they tend to yield long non-convex quadrangles when fat convex quadrangles are desired. For this reason, Bose & Toussaint [BT95] propose another algorithm which yields much nicer quadrangles, and easily accommodates points in general position.

Their second approach is to show that a set of points S always admits a simple spanning polygonal chain (whose vertex set is precisely the set of points in S) that in turn always admits a *serpentine* triangulation. In such a triangulation the diagonals (and hence triangles) are ordered in accordance to the ordering of the vertices of the dual chain. By removing every other diagonal starting at one end of the chain, they obtain the desired quadrangulation except perhaps a single triangle at the end. In such an eventuality they add one Steiner point outside the convex hull of S to convert the final triangle to a quadrangle. Then they show that the

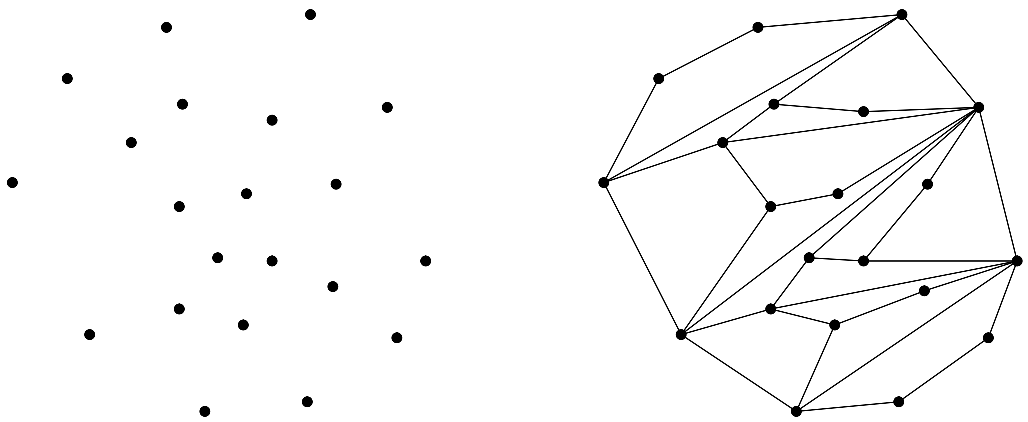

Fig. 2 A set of points S and the quadrangulation obtained with a variant of the Sequential Insertion Algorithm.

Steiner point is necessary if, and only if, the number of points h on the convex hull of S is odd.

In a different context concerned with fast rendering in computer graphics, Arkin et al. [AHMS] proposed an $O(n \log n)$ time algorithm for obtaining a serpentine triangulation (Hamiltonian in their terminology) of a set of points. However, their algorithm is based on sequentially inserting triangles in a triangulation of the points and hence, like the SI algorithm described above, its application to the quadrangulation problem leads to poor quadrangles similar to those in Figure 2. The serpentine triangulation algorithm of Bose & Toussaint [BT95] promisees to yield very nice and usually convex quadrilaterals.

A simple spanning polygonal chain most convenient for this purpose is the convex spiral of S. For the set of points in Figure 2 the resulting convex spiral is illustrated in Figure 3. This structure has appeared in the literature for some time and is closely related to the *onion peeling* of a set or the *convex layers* [Ch85], [To86], [Wi80]. In fact one can compute the spiral and the convex layers, one from the other, in $O(n)$ time [PS85] with a trivial scan of either structure. Bose & Toussaint [BT95] show that the convex spiral of a set S admits a serpentine triangulation. They then triangulate the spiral polygonal region with a variant of the *rotating calipers* [To83].

The Spiraling Rotating Caliper (SRC) Algorithm produces a serpentine triangulation. Therefore the diagonals and triangles of $T(P)$ are ordered according to the order of the vertices comprising the dual chain of $T(P)$. Let $D = (d_1, d_2,..., d_m)$ denote the diagonals in this order. If m is even, then if we delete from the triangulation every other diagonal d_2, d_4,... and so on, we obtain a quadrangulation. If m is odd we may delete every other diagonal starting from the last diagonal d_m. This will quadrangulate P except for the presence of one triangle on the boundary of the convex hull. Finally, by inserting one Steiner point just outside the convex hull of S near this triangle we may convert the triangle to a quadrangle.

Consider now the complexity of the SRC algorithm. The first step of the algorithm is to compute the convex spiral of the given set of points. This can be done via the convex layers of S in $O(n \log n)$ time using the algorithm of Chazelle [Ch85] or the algorithm of Hershberger and Suri [HS90]. This is the most difficult step in the algorithm, as both these algorithms are fairly involved. From the convex layers, we can compute a convex spiral P of S in $O(n)$ time with the procedure of Preparata and Shamos [PS85]. The spiraling rotating caliper algorithm for obtaining a serpentine triangulation of P runs in $O(n)$ time since no backtracking is

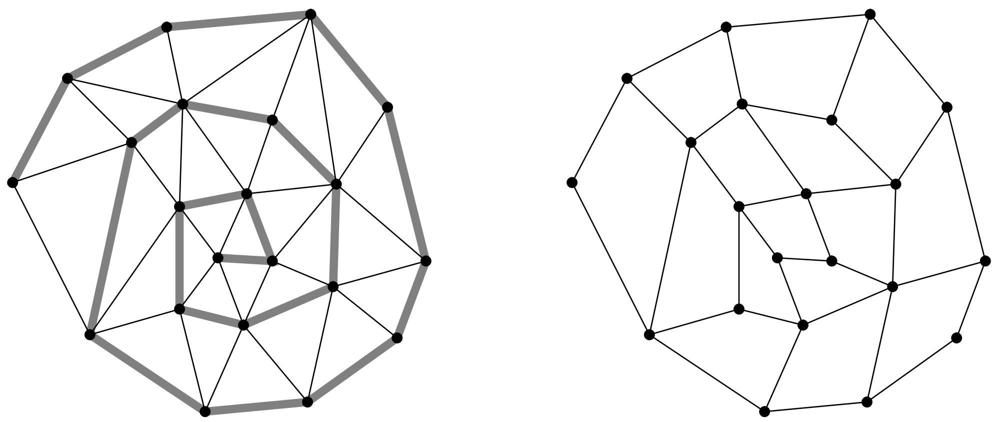

Fig. 3 Illustrating the convex spiral, serpentine triangulation and resulting quadrangulation of the set of points in Figure 2.

involved. Therefore the entire algorithm runs in $O(n \log n)$ time.

5. Converting Triangulations to Quadrangulations

Since little is known about computing quadrangulations, whereas triangulations have been well studied for a long time [Be92], engineers have devoted some attention to the problem of converting quadrangulations to triangulations [He83], [JSK91]. These methods are heuristic, conceptually cumbersome and require many Steiner points. For example, Johnston et al. integrate several heuristics into a system that automatically converts a triangular mesh into a quadrangular mesh which runs in $O(n^2)$ time and adds $O(n)$ Steiner points in the process. The problem of converting triangulations to quadrangulations has recently been investigated formally from graph-theoretic and computational geometric points of view by Ramos, Ramaswami & Toussaint [RRT95]. They have obtained a variety of characterizations for when a triangulation (of some structure such a polygon, set of points, line segments or planar subdivision) admits a quadrangulation without using Steiner points (or with a bounded number of Steiner points). They also propose efficient algorithms for accomplishing these tasks. For example, they point out (as mentioned earlier) that a triangulated polygon admits (contains) a quadrangulation *without* adding Steiner points if and only if the dual tree admits a *perfect matching*. They can also obtain quadrangulations with certain properties by applying *maximum weighted* matching algorithms on the dual graphs of the given triangulations. They also have several results on obtaining quadrangulations with bounded numbers of Steiner points. For example they show that a triangulated polygon can always be quadrangulated by adding at most $\lfloor n/3 \rfloor$ Steiner points *external* to the polygon and that there exist polygons that require this many *external* Steiner points. Furthermore, these Steiner points can be located in $O(n)$ time. The sufficiency of $\lfloor n/3 \rfloor$ Steiner points follows immediately from Fisk's proof [Fi78] for Chvátal's art gallery theorem [Chv75]. In his proof, Fisk showed that every triangulation of an n-gon P can be partitioned into $\leq \lfloor n/3 \rfloor$ *fans* (a *fan* is a triangulation where one vertex, called the *fan center*, is shared by all the triangles). Note that these fans start and end at edges of the polygon.

Consider now a vertex v of P that is a fan center. Vertex v defines a sequence of triangles in the triangulation. These triangles can be paired up to form quadrilaterals. If the number of such quadrilaterals is odd, we will be left with one triangle, one of whose edges is an edge of the polygon. One of the end-points of e is v; let the other be v'. We can convert this to a quadrilateral by adding a Steiner point p outside e, deleting the edge e and connecting p to the two vertices v and v'.

Thus we need to add at most one Steiner point per fan. Since P can be partitioned into $\leq \lfloor n/3 \rfloor$ fans, it follows that $\lfloor n/3 \rfloor$ outer Steiner points are always sufficient to quadrangulate a simple polygon.

In order to see that $\lfloor n/3 \rfloor$ outer Steiner points are sometimes necessary to quadrangulate a polygon it suffices to consider the polygon in Figure 4 (this is similar to an example of a polygon that requires $\lfloor n/3 \rfloor$ guards). There is only one way to triangulate this polygon, as shown in the figure. A simple case analysis shows that $\lfloor n/3 \rfloor$ outer Steiner points are necessary in order to quadrangulate the polygon.

To see that these Steiner points can be located in O(n) time, consider the following: P can be triangulated in O(n) time by Chazelle's algorithm [Ch91]. The triangulated polygon can then be three-colored in linear time (Kooshesh and Moret [KS92]). The edge on which a guard is placed gives us the fan-arm e outside which we place the Steiner point. To find an appropriate placement of the Steiner point, we may triangulate the simple polygon(s) that lie outside P and within the convex hull of P, which can also be done in linear time. The Steiner point for e can be placed anywhere inside the triangle incident on e (and outside P). If e is an edge of the convex hull, then it is not difficult to find a placement for the Steiner point. It follows therefore that all Steiner points can be located in O(n) time.

■

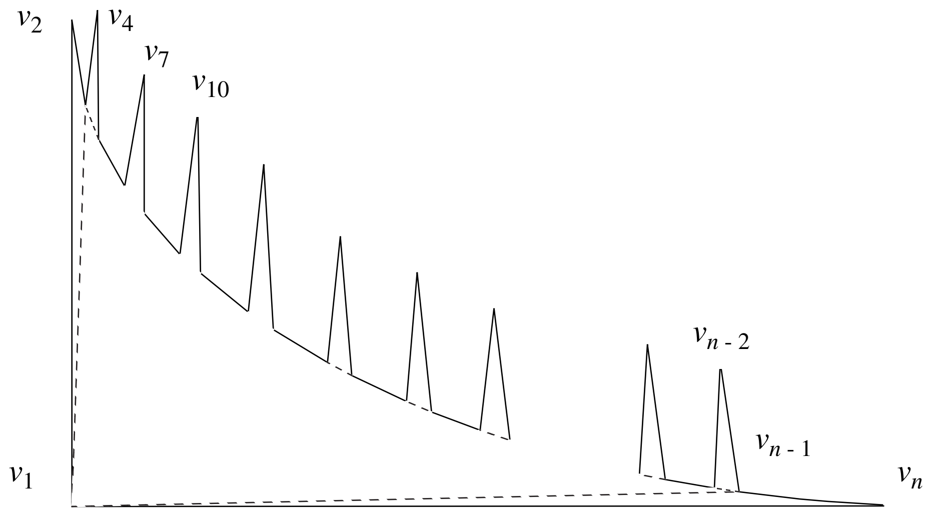

Fig. 4 A quadrangulation of this polygon requires $\lfloor n/3 \rfloor$ outer Steiner points. This polygon admits only one triangulation (as shown).

6. Open Problems

It is easy to see that a set of points does not always yield a quadrangulation where every quadrangle is convex. This observation leads to the following interesting open questions. Can one decide if a set of points admits a quadrangulation where every quadrangle is convex? This question was raised by Joe Mitchell at the MSI Workshop in Stonybrook in 1993. Can one compute the quadrangulation of a set of points that has the maximum number of convex quadrilaterals? Since Lubiw [Lu85] has shown that the decision problem is *NP*-complete for polygons that allow holes, it seems reasonable to believe that the same might hold for point sets. There is an intriguing open question concerning the SI (Sequential Insertion) algorithm for quadrangulating point sets when collinearities are allowed. Joe Mitchell has shown that the SI algorithm can be modified so that it works in spite of collinearities if we insert the points in order, say, sorted by x-coordinate. Does an optimal algorithm exist for points inserted in arbitrary order? Finally, we mention that paper-and-pencil examples suggest that the rotating-caliper algorithm gives significantly nicer quadrangulations than the SI algorithm. It would be interesting to compare these two algorithms both quantitatively and experimentally to determine the extent of the improvement afforded by the rotating-caliper algorithm for different measures of the quality of a quadrangulation.

7. Acknowledgments

The author is grateful to Larry Schumaker for introducing him to quadrangulations, and to Prosenjit Bose, Francisco Gomez, Joe Mitchell, Suneeta Ramaswami, Pedro Ramos and Xun Xue for interesting discussions on the topic.

8. References

[AAI86] Takao Asano, Tetsuo Asano and H. Imai, "Partitioning a polygonal region into trapezoids," *Journal of the A.C.M.*, vol. 33, No. 2, April 1986, pp. 290-312.

[AHMS] Arkin, E., M. Held, J. Mitchell, and S. Skiena, "Hamiltonian triangulations for fast rendering," *Algorithms-ESA'94*, J. van Leeuwen, ed., Utrecht, NL, LNCS 855, pp. 36-47, September 1994.

[BE92] Bern, M. and Eppstein, D., "Mesh generation and optimal triangulation," in *Computing in Euclidean Geometry*, F. K. Hwang and D.-Z. Du, eds., World Scientific 1992.

[BJM94] Baumgarten, H., Jung, H. and Mehlhorn, K., "Dynamic point location in general subdivisions," *Journal of Algorithms*, vol. 17, 1994, pp. 342-380.

[BS94] Barequet, G., and M. Sharir, "Piecewise-linear interpolation between polygonal slices," *Proceedings of the 10th Annual Symposium on Computational Geometry*, pp. 93-102, 1994.

[BT95a] Bose, P., and G. Toussaint, "No Quadrangulation is Extremely Odd," technical report #95-03, Dept. of Computer Science, University of British Columbia, January 1995.

[BT95b] P. Bose and G. T. Toussaint, "On computing quadrangulations of planar point sets," *10th Colloquium on Graph Theory, Combinatorics and Applications,* Feb. 27 - March 3, 1995, Xalapa, Mexico.

[BT95c] P. Bose and G. T. Toussaint, "Generating quadrangulations of planar point sets," accepted for publication in *Computer Aided Geometric Design*.

[Ch85] Chazelle, B., "On the convex layers of a convex set," *IEEE Transactions on Information Theory*, vol. IT-31, 1985, pp. 509-517.

[Ch91] B. Chazelle, "Triangulating a simple polygon in linear time," *Discrete Comput. Geom.*, vol. 6, 1991, pp. 485-524.

[Chv75] V. Chvátal, "A combinatorial theorem in plane geometry," *J. Combin. Theory Ser. B*, vol. 18, 1975, pp. 39-41.

[CJ92] Cheng, S. W. and Janardan, R., "New results on dynamic planar point location," *SIAM Journal on Computing*, vol. 21, 1992, pp. 972-999.

[CO90] H. E. Conn and J. O'Rourke, "Minimum weight quadrilateralization in O(n^3 log n) time," *Proc. of the 28th Allerton Conference on Comm. Control and Computing*, October 1990, pp. 788-797.

[ELOSU] Everett, H., W. Lenhart, M. Overmars, T. Shermer, and J. Urrutia, "Strictly convex quadrilateralizations of polygons," in *Proceedings of the 4th Canadian Conference on Computational Geometry*, pp. 77-83, 1992.

[Fi78] S. Fisk, "A short proof of Chvátal's watchman theorem," *J. Combin. Theory Ser. B*, vol. 24, 1978, pp. 374.

[FM84] A. Fournier and D. Y. Montuno, "Triangulating simple polygons and equivalent problems," *ACM Transactions on Graphics*, vol. 3, No. 2, 1984, pp. 153-175.

[He83] Heighway, E., "A mesh generator for automatically subdividing irregular polygons into quadrilaterals," *IEEE Transactions on Magnetic*s, 19, 6, pp. 2535-2538, 1983.

[Ho88] Ho-Le, K., "Finite element mesh generation methods: A review and classification," *Computer Aided Design*, 20, pp. 27-38, 1988.

[HS90] Hershberger, J., and S. Suri, "Applications of a semi-dynamic convex hull algorithm," *Proceedings of the second S.W.A.T., Lecture Notes in Computer Science 447*, Bergen, Sweden, pp. 380-392, 1990.

[JSK91] Johnston, B. P., Sullivan, J. M. and Kwasnik, A., "Automatic conversion of triangular finite meshes to quadrilateral elements," *International Journal of Numerical Methods in Engineering*, vol. 31, No. 1, 1991, pp. 67-84.

[KKK83] Kahn, J., M. Klawe, D. Kleitman, "Traditional galleries require fewer watchmen," *SIAM J. Algebraic Discrete Methods*, 4, pp. 194-206, 1983.

[KM91] A. A. Kooshesh and B. M. E. Moret, "Three-coloring the vertices of a triangulated simple polygon," *Pattern Recognition*, vol. 25, 1992.

[KS85] J. M. Keil and J.-R. Sack, "Minimum decompositions of polygonal objects," in *Computational Geometry*, Ed., G. T. Toussaint, North-Holland, Amsterdam, pp. 197-216.

[La94] Lai, M. J., "Scattered data interpolation and approximation by using C^1 piecewise cubic polynomials," submitted for publication.

[LLLMP79] W. Lipski, Jr., E. Lodr, F. Luccio, C. Mugnal and L. Pagli, "On two dimensional data organization II, *Fundanmenta Informaticae*, vol. 2, 1979, pp. 245-260.

[LS94] Lai, M. J. and Schumaker, L. L., "Scattered data interpolation using C^2 piecewise polynomials of degree six," *Third Workshop on Proximity Graphs*, Mississippi State University, Starkville, Mississippi, December 1-3, 1994.

[Lu85] Lubiw, A., "Decomposing polygonal regions into convex quadrilaterals," *Proc. Symposium on Computational Geometry*, 1985, pp. 97-106.

[OBS92] Okabe, A., B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, Chichester, England, 1992.

[Oh82] T. Ohtsuki, "Minimum dissection of rectilinear regions," in *Proc. IEEE International Symposium on Circuits and Systems*, Rome, 1982, pp. 1210-1213.

[O'R94] O'Rourke, J., *Computational Geometry in C*, Cambridge University Press, 1994.

[PS85] Preparata, F. P. and Shamos, M. I., *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.

[PT89] Preparata, F. P. and Tamassia, R., "Fully dynamic point location in a monotone subdivision," *SIAM Journal on Computing*, vol. 18, 1989, pp. 811-830.

[QS90] Quak, E. and Schumaker, L. L., "Cubic spline fitting using data dependent triangulations." *Computer-Aided Geometric Design*, vol. 7, 1990, pp. 293-301.

[RRT95] Ramaswami, S., Ramos, P. and G. T. Toussaint, "Converting triangulations to quadrangulations," manuscript in preparation.

[Sa82] Sack, J. R., "An $O(n \log n)$ algorithm for decomposing simple rectilinear polygons into convex quadrilaterals," *Proc. 20th Annual Conf. on Communications, Control and Computing,* Allerton, 1982, pp. 64-74.

[SNTM] Srinivasan, V., Nackman, L. R., Tang, J.-M. and Meshkat, S. N., "Automatic mesh generation using the symmetric axis transformation of polygonal domains," *Proceedings of the IEEE*, vol. 80, No. 9, September 1992, pp. 1485-1501.

[SP89] Sapidis, N. and Perucchio, R., "Advanced techniques for automatic finite element meshing from solid models," *Computer Aided Design*, vol. 21, No. 4, May 1989, pp. 248-253.

[SS79] N. Sugiyama and K. Saitoh, "Electron-beam exposure system AMDES," *Computer Aided Design*, vol. 11, 1979, pp. 59-65.

[SS88] Schroeder, W., and M. Shephard, "Geometry-based fully automatic mesh generation and the Delaunay triangulation," *International Journal for Numerical Methods in Engineering*, 24, pp. 2503-2515, 1988.

[ST81] Sack, J.-R. and Toussaint, G. T., "A linear-time algorithm for decomposing rectilinear star-shaped polygons into convex quadrilaterals," *Proc. 19th Annual Conf. on Communications, Control and Computing,* Allerton, 1981, pp. 21-30.

[ST88] Sack, J.-R. and Toussaint, G. T., "Guard placement in rectilinear polygons," in *Computational Morphology*, Ed., G. T. Toussaint, North-Holland, 1988, pp. 153-175.

[To83] Toussaint, G. T., "Solving geometric problems with the *rotating calipers," Proc. IEEE MELECON 83*, Athens, Greece, 1983, pp. A10002/1-4.

[To86] Toussaint, G. T., "New results in computational geometry relevant to pattern recognition in practice," in *Pattern Recognition in Practice II*, E. S. Gelsema and L. N. Kanal, Eds., North-Holland, 1986, pp. 135-146.

[Wa92] Wang, T., "A C^2-quintic spline interpolation scheme on triangulation," *Computer Aided Geometric Design,* vol. 9, 1992, pp. 379-386.

[WA86] Wang, Y., and J. Aggarwal, "Surface reconstruction and representation of 3-d scenes," *Pattern Recognition*, 19, pp. 197-207, 1986.

[Wi80] Wismath, S. K., "Triangulations: An algorithmic study," Tech. Report 80-106, Queens University, Kingston, Canada, July 1980.

[Yo75] Yoeli, P., "Compilation of data for computer-assisted relief cartography," in *Display and Analysis of Spatial Data*, J. Davis, and M. McCullagh, eds., John Wiley & Sons, New York, 1975.

[ZT89] Zienkiewicz, O. C. and Taylor, R. L., *The Finite Element Method*, vol. I, McGraw-Hill, London, 1989.