Matching Points to a Convex Polygonal Boundary
EXTENDED ABSTRACT

Matthew T. Dickerson* Michael T. Goodrich†

1 Introduction

A set of problems studied recently by Barequet *et al.* in [BBDG] and [BBD] involve the optimal placement of a *polygonal annulus region*, defined as the difference region between two *offset* or scaled copies of a polygon P. There are several optimization criteria, two of which are to maximize the number of contained points and to minimize the size of the annulus. In many applications, including those dealing with manufacturing processes, defining an annulus in terms of offset polygons is more natural than scaling. For example, the relative error of a production tool (milling head, laser, etc.) is independent of the location of the produced feature relative to some artificial reference point (the origin), and thus a tool is more likely to allow (and expect) local errors bounded by some tolerance, rather than scaled errors relative to some (arbitrary) center. For this reason, a study of the polygon offset operation, of the related distance function and its Voronoi diagram, and of annulus-placement problems for offset-polygons, are particularly interesting. Theoretical considerations were studied in [BDG] and [BBD], and were used in these papers as well as in [BBDG] in solutions to the offset versions of several annulus placement problems. Many ideas of [BBD] extended work of [BBBR] from circular annuli to polygonal annuli.

A constrained polygon annulus placement problem arises when one of the two annulus boundaries (inner or outer) is fixed. If a manufactured object is to fit inside a sleeve, for example, then its outer boundary has an absolute maximum, whereas if a part must fit over a peg then its inner boundary must be fixed. Thus fixing the inner or outer boundary is itself an important aspect of quality control, and was studied for circular annuli in [BBBR]. The following problem, which fixes inner polygon and seeks to minimize the outer, was thought to be the computational more difficult of the two constrained problems.

Problem 1: *Given a convex polygon P and a set of points S, find a translation τ that minimizes an outer offset O of P, such that all points in S lie in the annulus region between P and O.*

In [BBD], Barequet *et al.* give several competing algorithms for both constrained annulus problems (fixed inner and fixed outer), some of which are based on the approach of [BBBR]. Their algorithm for Problem 1, optimizing the outer offset, requires $O(nm\log(nm))$ time. In this paper we extend some ideas developed in [BBD] for the smallest enclosing polygon problem in order to present an algorithm to solve Problem 1 in $O(m + n\log n\log m)$ time. In addition to improving the algorithm of [BBD] by almost a linear factor when m and n are both large, our algorithm is considerably simpler in that it does not use the offset distance function Voronoi diagram.

2 Preliminary Observations

The *outer* δ-offset of a convex polygon P is obtained by translating each edge $e \in P$ by δ in a direction orthogonal to e and by extending it until it meets the translations of the edges neighboring to e. In other words, the new edge e' lies on a line parallel to that containing e and at distance δ from it on the outside of P. The edge e' is trimmed by the lines parallel and at distance δ (outside of P) of

*Dept. of Math and Computer Science, Middlebury College, Middlebury, VT 05753. E-mail: dickerso@middlebury.edu. Work by this author was supported by the National Science Foundation (NSF) Grant CCR-99-02032.

†Center for Algorithm Engineering, Dept. of Computer Science, Johns Hopkins Univ., Baltimore, MD 21218. E-mail: goodrich@acm.org. The work of this author was supported by NSF Grants CCR-9732300 and PHY-9980044.

the neighboring edges of e. We denote by $O_{P,\delta}$ the version of P offset outward by δ.

2.1 Containing Translations

We now present some terminology and additional properties of the offset-polygon distance function. Throughout the paper we assume polygon P has a fixed reference point. For a given point q in the plane, by a *translation* of P to q we mean a translation of P and its reference point so that the reference point coincides with q. Similarly, when we speak of the *reflection* of P, we mean the rotation of P by π around its reference point. The translation of a reflection of P to a point q translates the polygon so that the reference of the reflected copy is mapped to q.

Note that for offset-polygons, the natural reference is the offsetting center, which is the point to which the inner polygon collapses when the polygon is offset inward. (This point is the center of the medial axis of the polygon [BDG].) In the degenerate case, the offset center is a segment, so we arbitrarily select a point of it as the center, say, the median of the segment.

We say that a polygon P *contains* point q if q lies on the boundary or in the interior of P. We say that a polygon P *properly contains* point q if q lies strictly in the interior of P. The annulus region for a polygon P and a positive distance δ is defined as the set of points contained in the outer offset $O_{P,\delta}$ but not properly contained in P.

The following observations, reworded from [BBD], are fundamental in our algorithm:

1. A translation of a polygon P to a point q contains all the points of a set S if and only the intersection of the n copies of the reflection of P translated to the points of S contains q.
2. A translation of a polygon P to a point q properly contains none the points of a set S if and only q is not properly contained in the union of n copies of the reflection of P translated to the points of S.

From the previous observations, we see that to solve Problem 1 we are interested in finding a translation of P and some $O_{P,\delta}$ (for a δ to be determined) that lies in the intersection of n reflected of $O_{P,\delta}$, but not properly in the union of n reflected copies of P.

2.2 Efficiency of Some Substeps

In [BBD], several algorithms were given to compute the intersection of n copies of a convex polygon. Asymptotically, the most efficient of these requires $O(n \log h + m)$ time where h is the number of points on the convex hull of S.

Since translated copies of the same convex polygon together define a set of pseudo-disks, the complexity of the union of n translated copies of a convex m-gon P (e.g. the total number of edges and vertices possible on its boundary) is $O(nm)$ [KLPS]. However only $O(n)$ of these vertices represent intersections between two copies of P, while the rest of the vertices are just translated vertices of one copy of P. In other words, the boundary can be represented as a sequence of n chains, where each chain is just a translated copy of a contiguous portion of P. Given a reference copy of P, each such chain can be stored in $O(1)$ space. In this way, we can store a *compact representation* of the union of n translated copies of a convex m-gon in $O(n + m)$ space (which also follows from the bound of Kedem *et al.* [KLPS] regarding pseudo-disks). Moreover, by a simple divide-and-conquer algorithm, we can construct such a compact representation in $O(m + n \log n \log m)$ time.

3 The Algorithm

We now present our algorithm for Problem 1. The idea is to find some δ large enough to guarantee a containing annulus translation. We then compute the intersection of n reflected of $O_{P,\delta}$ translated to the points of S. We call this intersection $I_{P,\delta,S}$, or just I when P, δ, and S are clear from context.

We also compute the union of n reflected copies of P translated to the points of S which we denote $U_{P,S}$, or just U when P and S are clear from context. If q be any point that is contained in $I_{P,\delta,S}$ but not in properly contained in $U_{P,S}$, then a translation to q of the original P and $O_{P,\delta}$ gives a containing placement of the annulus region for set S. However it is not yet a solution to Problem 1 because δ is not minimized. We want to shrink δ back down to the smallest value such that $I_{P,\delta,S}$ has a non-empty intersection with the boundary or exterior of $U_{P,S}$—or, equivalently, grow $I_{P,S}$ outward from its central point to the first such δ.

The following lemma is helpful to that end.

Lemma 1: *Let x be the first point on the boundary of $U_{P,S}$ that is hit by an expanding offset of $I_{P,\delta,S}$ that begins at its center point c and grows outward. Then x is either a reflex vertex of U, or is on intersection of $SLL(I_{P,\delta,S})$, the straight-line skeleton of $I_{P,\delta,S}$, and the boundary of $U_{P,S}$.*

This leads to the following algorithm:

Algorithm 1

1. Compute an offset (or scaled) version of P (denoted as $P^* = O_{P,\delta}$, for some $\delta > 0$) large enough so that there exists a containing placement of the annulus region between P and P^*.
2. Compute the intersection I of n reflected copies of P^* translated to the points of S.
3. Compute the union U of n reflected copies of P translated to the points of S.
4. (a) Compute $SLL(I)$, the straight-line skeleton of I. Let c be the center of $SLL(I)$.
   (b) Determine whether c is properly contained in U. If it is not, then we are done. We let α be the amount by which we offset I inward until it degenerates to a point c. Then $\delta_{min} = \delta - \alpha$ is the width of the smallest annulus, and c is the translation of P and $O_{P,\delta-\alpha}$ that contains S.
   (c) If c is not properly contained in U, then we compute (using Lemma 1) the smallest inner offset α of I that contains a point x not properly contained in U. Our optimal annulus region is $\delta - \alpha$ and its containing translation is given by x.

3.1 Analysis

Before giving more precise detail about the algorithm, we make a few further observations. In the first step we need to compute a δ large enough that $I_{P,\delta,S}$ is not empty–that is, a δ that guarantees an annulus region large enough to contain S. (We can't shrink the annulus down if it isn't big enough to start with.) To this end we note that for at any vertex v of P, there is a semi-circle of radius δ centered at v that lies in the annulus region between P and $O_{P,\delta}$. So let w be the width of some boundary square whose sides are parallel to the $x-$ and $y-$ axes and that contains all of S. Then for $\delta = w\sqrt{5}/2$ there is a semi-circle centered at the leftmost vertex of P that lies in the annulus region between P and $O_{P,\delta}$ and which is large enough to contain the bounding square around S. Step 1 requires $O(n+m)$ time. $O(n)$ time is required to compute bounding square of S and $O(m)$ to offset P by this much. In step 2, we can compute the intersection of n copies of a convex polygon in $O(n \log n + m)$ time, or $O(n \log h + m)$ time where h is the size of the convex hull of S [BBD].

In step 3, the union of n copies of a convex polygon has complexity $O(nm)$ and an explicit version can be computed in time $O(nm \log(nm))$ using a divide-and-conquer strategy. The straight-line skeleton of step 4 can be computed in time $O(m)$ using the technique of [AGSS].

The last two parts of step 4 are the most complex. We can perform point location of c in time $O(log(nm))$. We then use ray-shooting for each of the m edges of $SLL(I)$ to determine where it intersects U. Conversely, we test each of the n reflex vertices of U to determine which region of $SLL(I)$ it falls in and then compute the offset at which the edge sweeps through it.

3.2 Improving the Algorithm with a Compact Diagram

The run time of Algorithm 1 can be improved by almost a linear factor (in the case when m and n are both large) by using a compact representation of $U_{P,S}$ as described in Section 2.2. As mentioned, $U_{P,S}$ has complexity $O(nm)$, but only $O(n)$ vertices represent the intersection of two copies, since the copies of P form a family of n pseudo-disks [KLPS]. Furthermore, all reflex vertices of $U_{P,S}$ are of these $O(n)$ intersection-type vertices. The rest of the vertices are from some copy of the reflection of P. This compact diagram U^* can be computed in $O(m + n \log n \log m)$ time, by using a simple divide-and-conquer strategy (where in the merge step we compute crossing points between two copies of P using appropriate "binary search" strategies). The reflex vertices needed in Step (4c) (see Lemma 1) are explicitly stored in U^*. It is only slightly more complex to compute the intersection of $SLL(I)$ with $U_{P,S}$. We do a ray-shooting query on U^* to determine which portion of a polygon the ray from $SLL(I)$ passes through, and then a second ray-shooting query on that particular portion of a polygon.

Thus, we have shown the following:

Theorem 2: *Given a convex m-gon P and a set S of n points in the plane, we can determine the translation for the minimum outer offset of P that contains all the points of S in $O(m + n \log n \log m)$ time.*

References

[AA] O. Aichholzer and F. Aurenhammer, Straight skeletons for general polygonal figures in the plane, *Proc. 2nd COCOON*, 1996, 117–126, *LNCS 1090*, Springer Verlag.

[AAAG] O. Aichholzer, D. Alberts, F. Aurenhammer, and B. Gärtner, A novel type of skeleton for polygons, *J. of Universal Computer Science* (an electronic journal), **1**, 1995, 752–761.

[AGSS] A. Aggarwal, L.J. Guibas, J. Saxe, and P.W. Shor, A linear-time algorithm for computing the Voronoi diagram of a convex polygon, *Discrete Computational Geometry*, 4 (1989), 591–604.

[BBD] G. Barequet, P. Bose, and M.T. Dickerson, Optimizing constrained offset and scaled polygonal annuli, *Proc. 6th Workshop on Algorithms and Data Structures*, Fredericton, Newfoundland, Canada, *Lecture Notes in Computer Science*, 1663, Springer-Verlag, 62–73, 1999.

[BBDG] G. Barequet, A.J. Briggs, M.T. Dickerson, and M.T. Goodrich, Offset-polygon annulus placement problems, *Computational Geometry: Theory and Applications*, 11 (1998), 125–141.

[BDG] G. Barequet, M. Dickerson, and M.T. Goodrich, Voronoi Diagrams for Polygon-Offset Distance Functions, *Proc. 5th Workshop on Algorithms and Data Structures*, Halifax, Nova Scotia, Canada, *Lecture Notes in Computer Science*, 1272, Springer Verlag, 200–209, 1997.

[BDP] G. Barequet, M. Dickerson, and P. Pau, Translating a convex polygon to contain a maximum number of points, *Computational Geometry: Theory and Applications*, 8 (1997), 167–179.

[BBBR] M. de Berg, P. Bose, D. Bremner, S. Ramaswami, and G. Wilfong, Computing constrained minimum-width annuli of point sets, *Computer-Aided Design*, 30 (1998), 267- -275.

[BKOS] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, Computational Geometry: Algorithms and Applications, Springer-Verlag, Germany, 1997.

[CD] L.P. Chew and R.L. Drysdale, Voronoi diagrams based on convex distance functions, Technical Report PCS- TR86-132, Dept. of Computer Science, Dartmouth College, Hanover, NH 03755, 1986; Preliminary version appeared in: *Proc. 1st Ann. ACM Symp. on Computational Geometry*, Baltimore, MD, 1985, 235–244.

[F] L.W. Foster, GEO-METRICS II: The application of geometric tolerancing techniques, Addison-Wesley, 1982.

[GMR] L. Guibas, R. Motwani, and P. Raghavan, The robot localization problem, in: *Algorithmic Foundations of Robotics*, A.K. Peters, Ltd., 1995, 269–282.

[KLPS] K. Kedem, R. Livne, J. Pach, and M. Sharir, On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles, *Discrete Comput. Geom.*, **1**:59–71, 1986.

[KN] J.L. Kelley and I. Namioka, Linear Topological Spaces, Springer Verlag, 1976.

[KS] D. Kirkpatrick and R. Seidel, The ultimate planar convex hull algorithm, *SIAM J. Computing*, 15 (1986), 287–299.

[KSn] D. Kirkpatrick and J. Snoeyink, Tentative prune-and-search for computing fixed-points with applications to geometric computation, *Fundamental Informaticæ*, 22 (1995), 353–370.

[KW] R. Klein and D. Wood, Voronoi diagrams based on general metrics in the plane, *Proc. 5th Symp. on Theoretical Computer Science*, 1988, 281–291, *LNCS 294*, Springer Verlag.

[MKS] M. McAllister, D. Kirkpatrick, and J. Snoeyink, A compact piecewise-linear Voronoi diagram for convex sites in the plane, *Discrete Computational Geometry*, 15 (1996), 73–105.