Hiding Disks in Folded Polygons

Therese C. Biedl* Erik D. Demaine†‡ Martin L. Demaine†
Anna Lubiw†‡ Godfried T. Toussaint*

June 30, 1998

Abstract

This paper considers the problem of finding a simple fold of a given polygon P that "hides" (covers both sides of) the largest possible disk. We solve this problem by giving a polynomial-time algorithm to find the largest pair of equal-radius non-overlapping disks in a polygon P. The desired fold is then the perpendicular bisector of the centers of these two disks. We also present some conjectures for the more general multiple-fold case when P is a square.

1 Introduction

The gift wrapping problem asks whether a given polyhedron can be wrapped up, or hidden, using a given piece of paper. We consider this problem one dimension down: Suppose you are given a piece of paper, in the shape of some polygon P. Given another shape Q, can you *hide* Q by folding P around it? By "hiding," we mean that both sides of Q must be covered by the paper. We allow the paper to stick out beyond Q, but insist that it be folded flat. This decision problem can be generalized to an optimization problem: what is the largest scaling of the shape Q that can be hidden by folding P around it?

In this paper we consider the case where Q is a disk. We consider only *simple folds* that fold the paper P along one straight line. Our starting point is the special case where we allow only one simple fold. The problem then becomes that of packing two non-overlapping copies of the largest disk possible into P. To our knowledge, this problem has not been studied before.

Related to the two-disk packing problem is the largest-empty-circle problem: find the largest (single) disk that fits in a given polygon P. This problem is traditionally solved by noting that the center of the disk must be at a vertex of the medial axis [9]. It can thus be

*School of Computer Science, McGill University, 3480 University Street, Montréal, Québec H3A 2A7, Canada, email: {therese,godfried}@cs.mcgill.ca.

†Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, email: {eddemaine,mldemaine,alubiw}@uwaterloo.ca.

‡Supported by NSERC.

computed in linear time [4]. A simpler algorithm [5] solves this problem for convex polygons without using the medial axis.

In Section 2 we use the medial axis to obtain a polynomial-time algorithm to find the largest disk that can be hidden by one simple fold of an input polygon P. In Section 3 we present conjectures for hiding a disk using sequences of simple folds when P is a square. Section 4 describes other interesting open problems.

2 One-Fold Case

Consider hiding a disk using one simple fold. Draw the outline of the disk through all layers of the folding, as if the paper were carbon paper. Then unfold the paper to restore the original polygon P (see Figure 1). The result is two equal-radius non-overlapping disks inside P. The fold is the perpendicular bisector of the two centers.

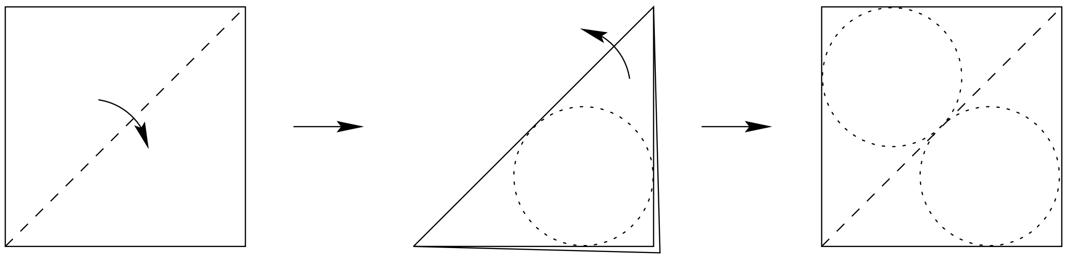

Figure 1: *Illustration of the one-fold case.*

Indeed, the disk-hiding problem for one simple fold is equivalent to the following packing problem: given a polygon P, find the largest pair of equal-radius non-overlapping disks that fit in P. To our knowledge, this packing problem has not been studied before for general polygons. Here we solve the problem in polynomial time.

Our basic approach is to examine all maximal pairs of disks, which clearly include the maximum pair. A pair is *maximal* if the disks cannot be displaced by continuous motion and then expanded (both to the same size) while remaining inside P and not overlapping.

Maximal pairs of disks are related to the *medial axis* [2, 7] of P. Conceptually, if we light a prairie fire along the boundary of the polygon, then the medial axis is where the fire meets and burns itself out. In other words, the medial axis is the set of points minimally equidistant to at least two boundary points of P. Its *vertices* are points that are minimally equidistant to at least three boundary points of P. The edges of the medial axis split into two kinds: line segments, which trace points that are equidistant from two edges of P, and sections of parabolas, which trace points that are equidistant from an edge and a reflex vertex of P. Technically, an edge of the medial axis may consist of multiple line-segment and parabolic pieces, but we consider the edges to be split at these points of change.

Lemma 1 *Given a maximal pair of equal-radius non-overlapping disks in P, they can be translated so that both centers lie on the medial axis of P.*

Proof: Let (D_1, D_2) be a maximal pair of equal-radius non-overlapping disks. Define an *object* to be D_1, D_2, or an edge of P. Consider first the case of one disk. Since disks are strictly convex, a disk can touch another object at only one point. If a disk touches no objects or just one object, it can be expanded slightly. Suppose that a disk with center c touches two objects at points p and q, respectively. If p, q, and c are not collinear (i.e., p and q are not directly opposite on the disk), then the disk can be expanded slightly. If the points are collinear, then we can translate the disk along a direction perpendicular to that line, until the disk touches another object and/or stops touching objects at directly opposite points. In either case, we can translate the disk to touch three objects, or expand it slightly.

Applying this result to the maximal pair of disks, either we can translate one of the disks to touch three objects, or expand them both slightly. Since the pair is maximal, we must be able to translate one disk (say D_1) to touch at least three objects. Since D_1 can only touch the other disk D_2 once, D_1 must touch two edges of P. In other words, D_1's center must be minimally equidistant to two boundary points of P, which is exactly the definition of D_1's center being on the medial axis of P.

Now fix D_1 and consider D_2. We can translate it so that it touches at least two objects. If it touches two edges of P, we are done. Assume that D_2 only touches one edge of P, and so it touches D_1. Then we can roll D_2 along (i.e., translate D_2 while touching) the boundary of P, in an appropriate direction so that D_2 moves away from D_1. How can this process terminate? If D_2 ever tries to roll to touch a convex vertex of P, D_2 will get "stuck" in this corner; in other words, we will have translated it to touch two edges of P, as desired. If D_2 comes back to touch D_1 again, it must have reached a convex vertex of P beforehand. P must have some convex vertices, so this must happen eventually. Therefore, we will translate D_2 to touch two edges of P, and hence its center must be on the medial axis of P. □

This lemma leads to the following algorithm to solve the packing problem for P.

1. Compute the medial axis M of P.
2. For each pair (e_1, e_2) of edges in M (each either a line segment or a section of a parabola):
   (a) Find p_1 on e_1 and p_2 on e_2 to give the largest pair of equal-radius non-overlapping disks in P centered at p_1 and p_2, respectively.
   (b) Add this pair of disks to the list of candidates.
3. Choose the largest candidate pair of disks.

For Step 1, the medial axis of an n-vertex polygon can be computed in $O(n \log n)$ time using the practical divide-and-conquer algorithm of Lee [6]. It was a long-standing open question whether the medial axis could be found in linear time. This was answered positively with more complex algorithms, initially for convex polygons [1] and eventually for general polygons [4].

The number of edges in the medial axis is $O(n)$, and hence the number of iterations in Step 2 is $O(n^2)$. It remains to show how to do Step 2 in polynomial time. In fact, it is simpler to split Step 2 into two cases: when the resulting disks touch, and when they are disjoint. We can handle the latter case as follows.

Lemma 2 *One disk of a maximal pair of equal-radius non-touching disks is centered at a vertex of the medial axis. Furthermore, it is the largest disk in P with that center.*

Proof: As we saw in the proof of the previous lemma, one disk in the maximal pair must touch at least three objects. By assumption, the maximal pair of disks do not touch, so the three objects must all be boundary edges of P. Hence, this disk is centered at a medial-axis vertex. Since it touches three boundary edges of P, it is the largest disk with its center. □

Hence, we modify Step 2 as follows.

2. For each edge e_2 of M:
   (a) For each vertex v of M:
      i. Find the largest disk D_1 in P centered at v.
      ii. Find (if possible) p on e_2 such that the disk D_2 with center p and the same radius as D_1 is in P and does not overlap D_1.
      iii. Add (D_1, D_2) to the list of candidates, if an appropriate D_2 was found.
   (b) For each edge e_1 of M:
      i. Find (if possible) p_1 on e_1 and p_2 on e_2 to give the largest pair of touching equal-radius non-overlapping disks in P centered at p_1 and p_2, respectively.
      ii. Add the resulting pair of disks to the list of candidates.

2.1 Parameterizing by Radius

Next we will parameterize a point on an edge of the medial axis by the radius of the largest disk in P centered at that point. This will allow us to consider two points on two respective medial-axis edges using the same parameter. Such a parameterization requires that only one point on an edge has a particular radius. To achieve this, we must cut the medial axis at its intersection with the angular bisector of every reflex vertex of P, if this intersection point lies on a parabolic edge; see Figure 2. As we have already indicated, we also separate parabolic pieces from line-segment pieces of medial-axis edges.

Consider an edge (v, w) of the medial axis; it will be one of two kinds.

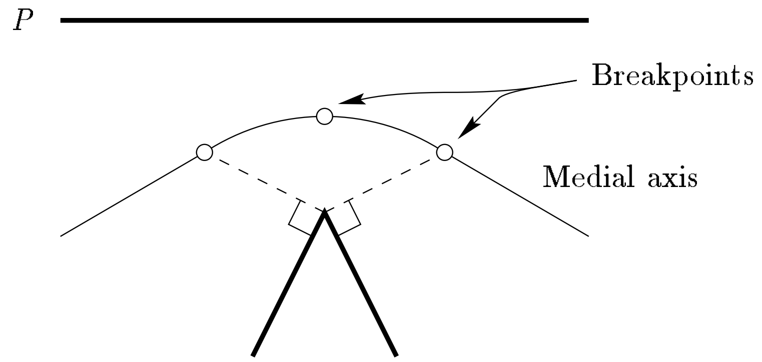

Figure 2: *Splitting the edges of the medial axis.*

Line-segment edges. Extend (v,w) and the two edges of P it bisects; see Figure 3. Assume that the boundary edges are not parallel, so that (v,w)'s extension meets their extensions at a point a as an angular bisector (we justify this assumption below). Let 2θ be the bisected angle. The radius r from a point p on (v,w) is the length of the perpendicular to either of the bisected edges of P. Then

$$\sin\theta = r/d(p,a), \quad \text{so} \quad d(p,a) = r/\sin\theta,$$

where $d(p,a)$ denotes the Euclidean distance between p and a. Hence, $p = p(r) = a + (r/\sin\theta)\vec{x}$, where $\vec{x} = (v-a)/d(v,a)$ is the unit vector in the direction from a to v. (If $a = v$, swap the labels of v and w.) This parameterizes p in terms of r.

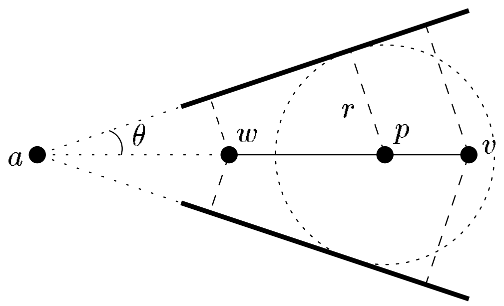

Figure 3: *Illustration of parameterization of line-segment edges.*

We do not have to consider the case when the two bisected edges of P are parallel. If we had a pair of maximal disks (D_1, D_2), with D_1 centered at the middle of such a line-segment edge, we could translate D_1 to either v or w (avoiding D_2 by choosing between v or w appropriately), not changing the radius since the edges of P are parallel. Hence, a pair of disks of equal radius has already considered by Step 2(a).

Parabolic edges. Let u and e be the reflex vertex and edge of P, respectively, such that points on the medial-axis edge (v,w) are equidistant to u and e; see Figure 4. Let a and b be the (orthogonal) projections of v and w, respectively, onto e. The radius r of a point p on (v,w) is $d(u,p)$, which equals the length of the perpendicular from p to e.

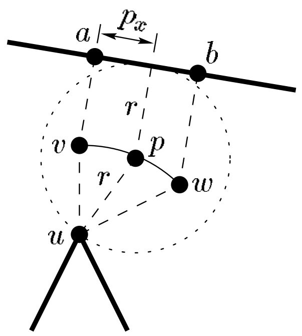

Figure 4: *Illustration of parameterization of parabolic edges.*

Consider the coordinate frame based at a and with coordinate vectors $\vec{x} = (b-a)/d(b,a)$ and $\vec{y} = (v-a)/d(v,a)$. These are orthogonal vectors, since a is an orthogonal projection of v onto e. We want to compute $r = d(u,p)$, so let us consider the coordinates of u and p in this frame. Let u_x and u_y be the constants such that $u = a + u_x\vec{x} + u_y\vec{y}$. Let p_x and p_y be such that $p = a + p_x\vec{x} + p_y\vec{y}$, and note that $p_y = r$, since this is exactly the distance from point p to edge e. Now we can compute

$$r = d(p,u) = d(a + p_x\vec{x} + r\vec{y}, a + u_x\vec{x} + u_y\vec{y}) = \sqrt{(p_x - u_x)^2 + (r - u_y)^2}.$$

Squaring both sides, we have that

$$r^2 = (p_x - u_x)^2 + (r - u_y)^2, \quad \text{i.e.,} \quad p_x^2 - 2u_x l + u_x^2 - 2ru_y + u_y^2 = 0,$$

$$\text{so} \quad p_x = u_x + \sqrt{u_y(2r - u_y)}.$$

But $p = a + p_x\vec{x} + r\vec{y}$, so this parameterizes p in terms of r.

2.2 Solving Step 2

Recall that Step 2 of our algorithm was concerned with finding a pair of maximal equal-radius non-overlapping disks, with one center on a given edge e_2 and the other center at a given vertex v (Step 2(a)) or on an edge e_1 (Step 2(b)). Let $e(r)$ denote the parameterization of a medial-axis edge e in terms of the radius r, as described in the previous section. Note that an edge e has an implicit range, denoted $R(e)$, of radii it can achieve.

Let $r(p)$ denote the radius of the largest disk in P centered at medial-axis point p. Then Step 2(a) is easy to solve. First, compute the subset e_2' of points p on e_2 with $r(p) \geq r(v)$. Note that e_2' is a subedge of e_2, and that its ends can be computed using $e_2^{-1}(r(v))$. If e_2' is non-empty, check whether the disk centered at either end of e_2' with radius $r(v)$ overlaps the disk centered at v with the same radius. If not, add the resulting pair of disks to the list of candidates.

Now let us consider Step 2(b). We need to solve the touching equation, $2r = d(e_1(r), e_2(r))$, where r is allowed to be in the range $R(e_1) \cap R(e_2)$. There are three cases, depending on the types of the edges e_1 and e_2.

Case 1. Both e_1 and e_2 are line segments. Then the equation is

$$2r = d(a_1 + (r/\sin\theta_1)\vec{x_1}, a_2 + (r/\sin\theta_2)\vec{x_2}),$$

which is a quadratic polynomial in r. Hence, we can apply the quadratic equation and find r in constant time. Note that this is the only case we have to consider if P is convex.

Case 2. e_1 is a line segment and e_2 is parabolic. Then the equation is

$$4r^2 = \left(d\left(a_1 + (r/\sin\theta)\vec{x_1}, a_2 + r\vec{y} + \left(u_x + \sqrt{u_y(2r - u_y)}\right)\vec{x_2}\right)\right)^2,$$

which is of the form

$$4r^2 = \left(a + br + c\sqrt{r-k}\right)^2 + \left(d + er + f\sqrt{r-k}\right)^2,$$

which in turn is of the form

$$gr^2 = h + ir + (j + mr)\sqrt{r-k}.$$

Bringing all but the $\sqrt{r-k}$ term to the left-hand side and squaring, we obtain

$$(gr^2 - h - ir)^2 = (j + mr)^2(r-k),$$

which is an equation of degree four in r. This can be solved exactly in constant time.

Case 3. Both e_1 and e_2 are parabolic. With a derivation similar to Case 2, the resulting equation has degree eight. This can be solved using the bisection method and/or the Newton-Raphson method [8, pp. 274–292]. For these methods to work, we first need to find a *bracket* of a root, that is, two inputs for which the polynomial gives opposite signs. In other words, we need to find two radii parameters that give disks that are disjoint and overlapping, respectively.

This can be done as follows. Recall that we are only interested in solutions in the range $R = R(e_1) \cap R(e_2)$. We claim that R brackets a root, provided the touching equation has a solution. Suppose to the contrary that the disks at radius $\min R$ are overlapping, as are the disks at radius $\max R$. Then the disks will overlap at all radii in R, so there is no solution. Similarly, if both pairs of disks (at radii $\min R$ and $\max R$, respectively) are disjoint, then they will continue to be disjoint at all radii in R, so again the equation has no solution. Hence, R being a bracket is equivalent to the touching equation having a solution.

Thus, to solve Step 2(b), we check whether R is a bracket. If it is not, we skip this pair of edges. If it is, we apply one of the root-finding methods mentioned above, using this bracket. The running time of the bisection method is linear in the desired number of bits for the solution. A hybrid with the Newton-Raphson method improves the performance in practice, but has the same worst-case bound.

In all, the one-fold disk-hiding problem can be solved in polynomial time under an appropriate model of computation. For convex polygons, we only need a real RAM with the ability to take square roots. For general polygons, square roots are not enough; instead, we compute the solution to the desired number of bits, in time that is polynomial in this number of bits.

3 Square Case

We now turn to the multiple-fold question: given a polygon P and a positive integer k, what is the largest disk you can hide by folding P with k simple folds? One way to simplify the problem is to place the restriction of allowing at most two layers of paper at a point. At first we thought that, under this restriction, an n-vertex polygon could be folded at most n times. However, this is not true, not even for rectangles. You can make arbitrarily many folds with a long-enough rectangle. Indeed, the number of folds required to hide the largest disk can be arbitrarily large for sufficiently long constant-width rectangle; see Figure 5.

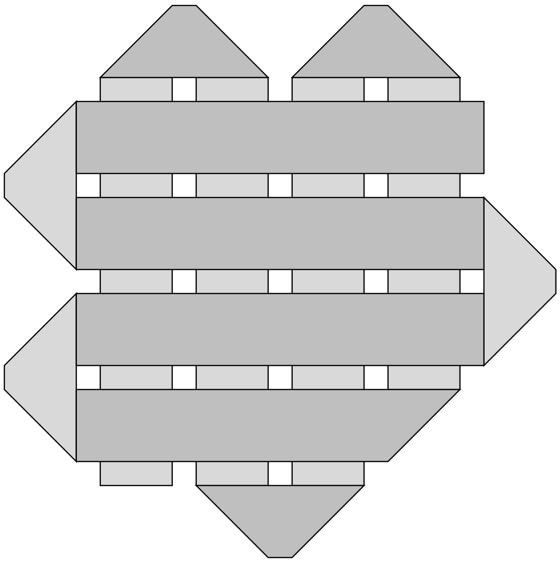

Figure 5: *Hiding an arbitrarily large disk using a sufficiently long constant-width rectangle and many folds (loosened for illustration).*

More surprising is that this is not even true for a square; an example of five folds of a square is given in Figure 6. We conjecture that a square cannot be folded more than five times, without having more than two layers at a point. We also conjecture that five folds do not allow us to hide any bigger disks than four folds. It would also be interesting to consider $k \geq 5$ folds without the two-layer constraint.

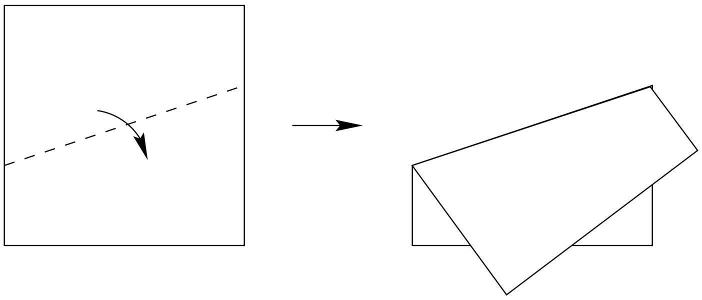

Figure 6: *A fold in a square that allows four more folds without overlap.*

Now let us consider P a square, and $k \in \{1, 2, 3, 4\}$. The best "non-trivial" foldings we have found are shown in Figure 7. By non-trivial we mean that the k-fold does not approximate a j-fold for $j < k$.

What we find interesting is that, while three and four folds permit bigger disks than one fold, it appears that two folds do not. Of course, you could get arbitrarily close to the

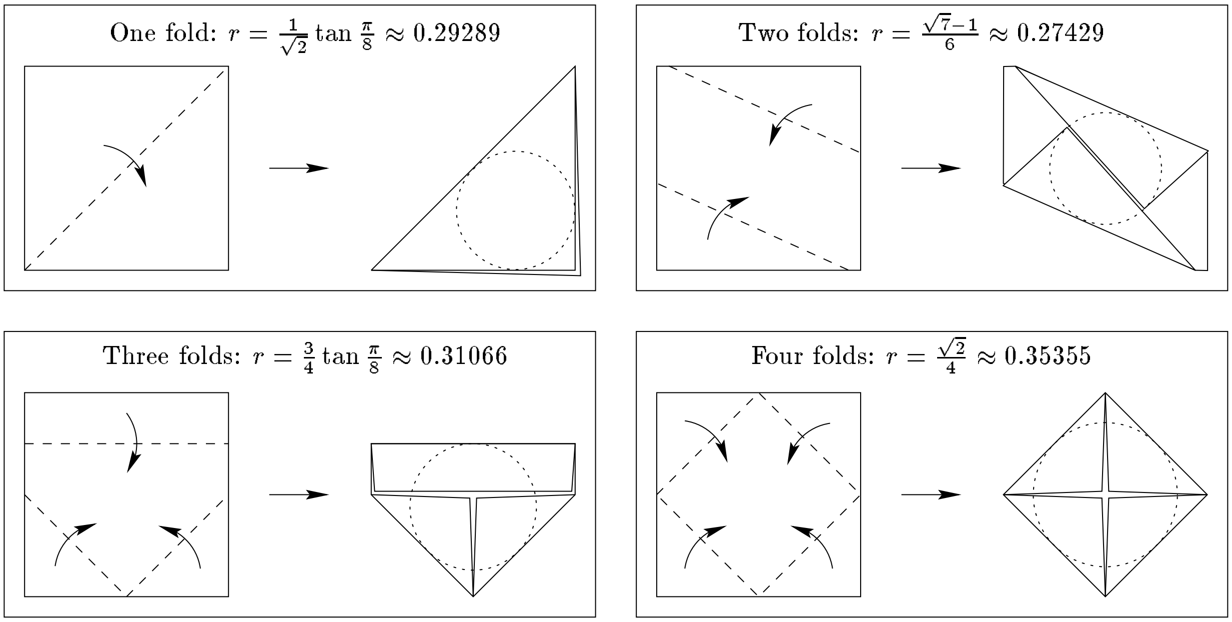

Figure 7: *Best known non-trivial 1-, 2-, 3-, and 4-fold disk hidings for a square. The disk radius r is specified relative to the square's side.*

one-fold radius with two folds, by slightly shifting the diagonal and having a tiny second fold. Also, if you remove the restriction of at most two layers, you can add a tiny second fold and get exactly the one-fold radius. The two folds shown in Figure 7 form the most competitive "different" solution.

We know from our results in Section 2 that our one-fold solution is optimal. However, it is open whether our two-, three-, or four-fold solutions can be improved. In particular, we would like to know whether you can exploit two folds to hide a bigger disk than with one fold.

4 Open Problems

We conclude with a list of related open problems.

1. Improve the two-, three-, or four-fold solutions for the square. The two-fold case is particularly interesting.

2. Generalize our one-fold theory to more folds. This is a natural way to attack the first open problem.

3. Give an efficient algorithm to decide, given polygons P and Q, whether Q can be hidden by P.

   We note that there is relevant work on packing a single copy of Q in the polygon P (see [3] and the references therein), but even the single-fold problem is no longer equivalent to packing two disjoint copies of Q in P—we need to reflect one copy of Q and ensure that the two copies are symmetric through a line.

4. How does the problem change if we hide circles instead of disks, that is, we only need to hide the boundaries of the disks?

5. Higher dimensions: we could imagine folding n-dimensional polytopes through $n+1$ dimensions, where a simple fold is a reflection through an $(n-1)$-dimensional object. A maximal n-ball must now touch at least $n+1$ objects, so n of them must be edges of P. Hence, we can again argue that the centers are on the medial axis, and this part of our one-fold theory carries over. Now how do we find the largest pair of equal-radius non-overlapping n-balls?

Acknowledgments

The research reported here was initiated at the International Workshop on Wrapping and Folding, co-organized by Anna Lubiw and Sue Whitesides, at the Bellairs Research Institute of McGill University, January 31–February 6, 1998. We wish to thank Joe O'Rourke and Steve Robbins for valuable discussions and their comments on the paper.

References

[1] Alok Aggarwal, Leonidas J. Guibas, James Saxe, and Peter W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete and Computational Geometry*, 4(6):591–604, 1989.

[2] Harry Blum. A transformation for extracting new descriptors of shape. In W. Whaten-Dunn, editor, *Proceedings of the Symposium on Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.

[3] L. P. Chew and K. Kedem. Placing the largest similar copy of a convex polygon among polygonal obstacles. *Computational Geometry: Theory and Applications*, 3:59–89, 1993.

[4] Francis Chin, Jack Snoeyink, and Cao An Wang. Finding the medial axis of a simple polygon in linear time. In *Proceedings of the 6th Annual International Symposium on Algorithms and Computation*, number 1004 in Lecture Notes in Computer Science, pages 382–391, 1995.

[5] Ferran Hurtado, Vera Sacristán, and Godfried T. Toussaint. Constrained facility location. *Studies in Locational Analysis, Special Issue on Computational Geometry in Locational Analysis*, to appear in 1998.

[6] D. T. Lee. Medial axis transformation of a planar shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4):363–369, July 1982.

[7] Joseph O'Rourke. *Computational Geometry in C*, pages 193–195. Cambridge University Press, 1994.

[8] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 1992.

[9] Michael I. Shamos and Dan Hoey. Closest-point problems. In *Proceedings of the 16th Annual IEEE Symposium on Foundations of Computer Science*, pages 151–162, Berkeley, California, October 1975.