# Optimal Polygon Placement

Prosenjit Bose *  Jason Morrison †


## Abstract

Given a simple polygon $P$ with $m$ vertices and a set $S$ of $n$ points in the plane, we consider the problem of finding a rigid motion placement of $P$ that contains the maximum number of points in $S$. We present two solutions to this problem that represent time versus space trade-offs. The first algorithm runs in $O(n^3m^3)$ expected time using $O(n^2m^2)$ space. The second algorithm runs in $O(n^3m^3 \log(nm))$ deterministic time and $O(nm)$ space. While these algorithms represent a substantial improvement in the time bounds of previous work the main contribution is that the approach is extendible to related rigid motion placement problems including polygonal annulus placement.


## 1 Introduction

This paper studies the problem of finding a rigid motion placement of a simple polygon $P$ on a point set $S$ such that the maximum number of points of $S$ are contained within $P$. The approach taken in the paper is to extend work done on rigid motion placement of convex polygons by Dickerson and Scharstein [4]. In this paper we introduce a technique that decomposes the simple polygon into a series of constrained convex polygon placements that are analyzed using the methodologies and results of Dickerson and Scharstein. As this is an extended abstract full details are available in [7]. The technique used decomposes a simple polygon into convex polygonal pieces and can be generalized to decompose polygonal annuli and other geometric structures defined by line segments. Such a generalization would extend results here to rigid motion placement of polygonal annuli etc. We further note that the techniques presented here to determine the placement which maximizes the number of contained points can also be used to find the placement that minimizes the number of contained points.

The problem of finding the rigid motion placement of a polygon to include the maximum or minimum number of points is motivated by several applications. In fabric cutting the placement of the desired shape to be cut on a specific sheet of fabric can be done such that the most number of features (*e.g.* pattern locations) are included. Alternatively the shape can be placed and hence cut with the least number of fabric flaws included. In the area of geometric tolerancing the use of polygon placement can determine if a given manufactured part "covers" the important tolerance points. In the *robot localization problem* (see *e.g.* [2, 6]) a robot first detects the distance to points on the walls of the room it is in with a range finder. The robot then uses the measured points and its internal polygonal map of the room to determine how the "room" is oriented and positioned around it.

## 2 Previous Work and Definitions

Under differing rigid motions a polygon may rotate, translate or some combination of the two but may not scale. Formally a *planar rigid motion* $\rho$ is an affine transformation of the plane that preserves distance and hence preserves angles and areas as well. Given this definition the *optimum polygon placement problem* studied here determines a rigid motion $\rho$ that when applied to a given polygon $P$ will maximize the number of points of a given set $S$ contained in $\rho(P)$.

We immediately observe that every rigid motion placement of a polygon $P$ on a point set $S$ partitions $S$ into exactly two sets: the set $S_{(\rho,ext)}$ is all of points of $S$ not contained in $\rho(P)$ and the set $S_{(\rho,int)}$ is all of points of $S$ contained in $\rho(P)$ including the boundary of $\rho(P)$. We classify two rigid motion placements $\rho_1(P)$ and $\rho_2(P)$ as *combinatorially equivalent* if $S_{(\rho_1,ext)} = S_{(\rho_2,ext)}$ and hence $S_{(\rho_1,int)} = S_{(\rho_2,int)}$, otherwise the placements are *combinatorially distinct*. To solve the optimal polygon placement problem it is sufficient to examine at most one placement for each set of combinatorially equivalent placements. To enumerate a suitable set of placements to examine, Chazelle [3] defines *stable placements* to be a placement where three points of $S$ are on three segments of $P$.

The key to proving that there are a finite number of stable placements and that examining these placements is sufficient for solving our problem lies in Lemmata 1 & 2, which were proven by Chazelle [3]. Lemma 1 bounds the number of stable placements and the time to compute them.

*Research supported in part by NSERC. School of Computer Science, Carleton University, Ottawa, Canada, jit@scs.carleton.ca

†Biosystems Engineering, University of Manitoba, Winnipeg, Canada, jason_morrison@umanitoba.ca

**Lemma 1 (Chazelle 1)** *Given any three points $A, B, C$ of $S$ and any segments $a, b, c$ of $P$, there are at most two stable placements involving $A$ on $a$, $B$ on $b$, and $C$ on $c$. Moreover, these placements can be computed in constant time.*

Lemma 2 establishes that examining the stable placements is sufficient to solving our problem.

**Lemma 2 (Chazelle 3)** *Let $S$ be a planar point set and $P$ be a simple polygon. If there exists a rigid motion placement $\rho(P)$ which contains $S_{(\rho,int)}$, then there exists a rigid motion placement $\rho^*(P)$ such that it is a stable placement and $S_{(\rho^*,int)} \subseteq S_{(\rho,int)}$.*

Thus all stable placements of an $m$ vertex polygon $P$ on a planar point set $S$ with $n$ points can be calculated in $O(n^3m^3)$ time. For each placement it is possible to determine the number of points contained in the placed polygon in $O(n \log(m))$ time. Thus Chazelle's brute-force technique [3] requires $O(n^4m^3 \log(m))$ time and is the best known result for the optimal polygon placement problem.

Dickerson and Scharstein [4] present analysis and algorithms for the specific case of the optimal polygon placement problem where the polygon is convex. They begin by noting that Chazelle's brute force technique improves to $O(n^4m^2 \log(m))$ time, due to fewer stable placements. Then they note that the convex problem is solvable in $O(n^2k^3m^2 \log(m))$ time using the concepts of Eppstein and Erickson [5], where $k$ is the maximum number of points contained by the polygon. After these initial musings Dickerson and Scharstein [4] improve on Chazelle's results with algorithms that use their new notion of rotation diagrams. A critical observation regarding the brute force approach that is addressed by rotation diagrams is that stable placements should be examined in an order where the difference between successive sets of $S_{int}$'s and $S_{ext}$'s is minimal. Specifically, Dickerson and Scharstein [4] present the concept of rotation diagrams and give two algorithms that use them to solve the convex problem in $O(n^2km^2 \log(mn))$ time or $O(nk^2m^2 \log(mk))$ time, each using $O(m + n)$ space where $k$ is the maximum number of points contained by the polygon.

To solve a given problem Dickerson and Scharstein examine $n$ rotation diagrams. Given a point $p \in S$ and a convex polygon $P$, the *rotation diagram* $R_{(P,p)}$ has a containing region $C_{(P,p,p_i)}$ for every point $p_i \in S$ such that $p_i \neq p$. Every *containing region* $C_{(P,p,p_i)}$ is defined to be the set of rigid motions for which $p$ is on the boundary of $P$ ($\partial P$) and $p_i$ is contained in $P$ (Note: throughout this paper $\partial P \subset P$).

The primary concept of $R_{(P,p)}$ is to use a planar arrangement to characterize many stable placements of $P$ on $S$. Each of the possible rotations and translations with $p \in \partial P$ are characterized by $p$'s position $L$ on the perimeter of $P$ and the angular rotation $\Theta$ of $P$. If the perimeter of $P$ is $W$ then $O \leq L \leq W$ and $0 \leq \Theta \leq 2\pi$. Thus the rotation diagram $R_{(P,p)}$ plots the containment region of all $p_i$ when $p \in \partial P$, with respect to $(L, \Theta)$. The containment region of $p_i$ is plotted as a region using these variables. Dickerson and Scharstein also prove that due to convexity of $P$ this containment region can be represented by $O(m^2)$ vertical line segments and trigonometric curve segments. An example of a rotation diagram is given in Fig. 1.

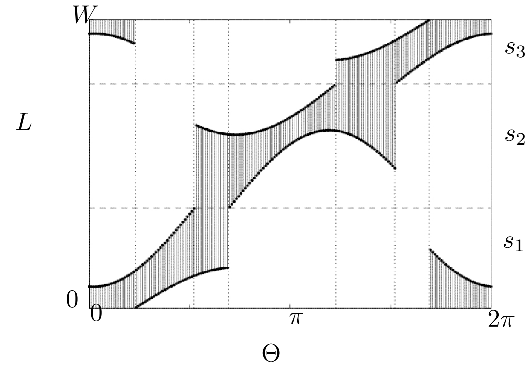


Figure 1: Rotation Diagram $R_(P, p)$ with a containment region and $P$ is a triangle comprised of three edges $(s_1, s_2, s_3)$ (adapted from [4]).

## 3 Rotation Diagrams for Simple Polygons

We define a rotation diagram for a simple polygon to be analogous to the definition for convex polygons given by Dickerson and Scharstein [4] with one minor change. A rotation diagram $R_{(P,p,s)}$ for a simple polygon is a plot that assumes that some point $p \in P$ is on segment $s$ of the boundary of the polygon $P$. Dickerson and Scharstein's rotation diagrams for convex polygons only assume that $p$ is on the boundary of the polygon $P$.

Since the previous work provides a framework, our goal is to divide the optimal polygon placement problem into a set of problems involving convex polygons. We achieve this by first triangulating the polygon into a set of triangles $T = \{t_1, t_2, ...\}$. We subsequently redefine containment regions $C_{(P,p,s,p_i,t_j)}$ such that not only is $p \in s$ and $p \in P$ but also $p_i \in t_j$. Our approach to computing this new containment region is to first compute the convex hull $c$ of $s$ and $t_j$ and compute its containment region using the methods of [4]. From this we have to subtract the containment region for the polygon $u$ where $u = c - t_j$ to ensure the regions only represent placements where $p_i \in t_j$.

The first difficulty of this approach is when $s$ lies within $c$. In this case we create a constant number of triangles $\{t'_1, ...\}$ by partitioning $t_j$ and creating their containment regions in place of $t_j$. These triangles are

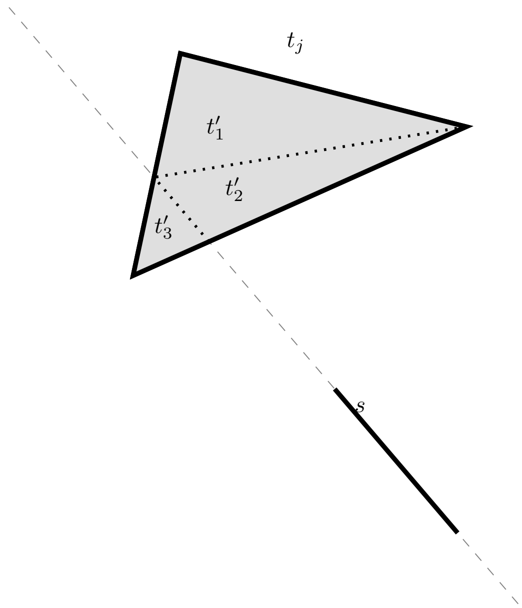


Figure 2: Splitting $t_j$ with $s$.

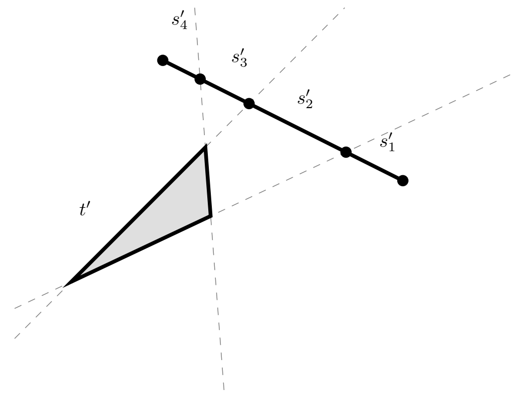


Figure 3: Splitting $s$ with a triangle.

generated by the line containing $s$ and arbitrarily dividing the remaining quadrilateral (See Fig. 2).

The next difficult point occurs if $t$ does not have an edge on $c$. In this case $s$ can be decomposed into a constant number of constant sized pieces that each have this property (See Fig. 3). After this procedure, each of the generated convex polygons $c$ is generated by a segment $s'$ and a triangle $t'$ and $u' = c - t'$, where $t' \in t_j$, $s' \subseteq s$, $s' \subset \partial c$ and either $u'$ is convex or all vertices of $u'$ are visible from $s'$. Since $c$ is convex and $s' \subset \partial c$ we can determine the containing region for which $p_i \in t'$ and $p \in s'$, using rotation diagrams for convex polygons [4]. Furthermore this is possible in constant time as $|c| = O(1)$. All that remains is to subtract out the regions for which $p_i \in u'$. This is also easily accomplished as $u'$ is either convex or contains only one reflex vertex (see Fig. 4) and can be covered with a constant number of constant sized convex polygons.

Thus for a given triangle $t_j$, segment $s$ and point $p_i$ the containment region can be calculated in $O(1)$ time. For any one diagram there are at most $O(mn)$ such regions as each point $p_i$ may lie within any triangle $t_j$ with $O(1)$ complexity each. The arrangement of curve segments for this diagram can be calculated in $O(nm \log(nm) + K)$ expected time using Mulmuley's result [8] or $O((nm + K) \log nm)$ deterministic time using Bentley and Ottmann's result [1] where $K$ is the number of intersections between boundaries of containment regions.

A total of $O(n^2m^2)$ intersections are possible in the resulting rotation diagram. This corresponds directly to the number of stable placements of $p \in s$. We note that the analysis of the following algorithms assumes

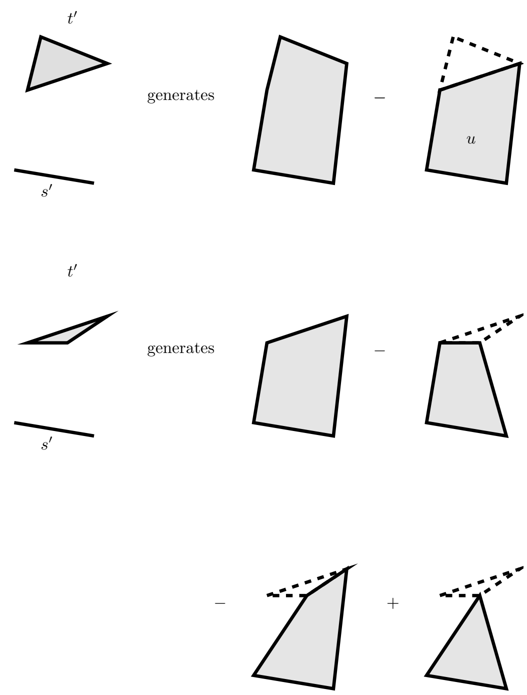


Figure 4: $t'$, $s'$ configurations and "subtraction" of $u'$.

the points are in general position such that no placement exists in which more than three points lie on the boundary of $P$. Lastly we note that this method of generating rotation diagrams did not make use of the nature of the simple polygon except to guarantee the number of triangles generated when partitioning it. Thus this method is extendible to polygonal annuli and other geometric structures.

## 4 Optimal Polygon Placement Algorithms

The two algorithms described below represent a time versus space tradeoff. The first algorithm runs in $O(n^3m^3)$ expected time with $O(n^2m^2)$ space while the second algorithm $O(n^3m^3 \log nm)$ deterministic time with $O(nm)$ space.

It should be noted that the complexities of the rotation diagrams for simple polygons are overestimated. The worst case analysis does not consider the relative distance between points. In fact there can be a number of containment regions that are empty because there exists no placement of $P$ which covers $p$ and $p_i$. Dickerson and Scharstein use this fact along with the convexity of the polygon to decrease the number of containment regions in a given rotation diagram. Unfortunately their technique does not apply to simple polygons.

The first algorithm begins by constructing the arrangement of containment regions. Once the arrangement has been constructed a single stable placement is calculated, and the number of points lying on the interior are counted, which requires $O((n+m)\log m)$ time. Then the arrangement can be walked. At each intersection of the arrangement a single point switches from being contained to being uncontained (or vice versa). This information is easily maintained in $O(1)$ time and the optimality of the placement is also checked. Thus the entire set of stable placements involving $p$ on $s$ is checked in $O(n^2m^2)$ time.

Since $nm$ different rotation diagrams exist, all arrangements can be checked in $O(n^3m^3)$ expected time or $O(n^3m^3 \log nm)$ deterministic time. This improves on the time complexity of the previous algorithm by $O(n \log m)$ in the expected case but uses more space (*i.e.*, $O(n^2m^2)$).

The second algorithm does not construct the rotation diagram. Instead it computes all of the containment regions and sorts their boundary segments by endpoint from left to right. The goal is to scan a vertical line from left to right through the rotation diagram maintaining the maximum point of overlap. This is accomplished using a segment tree as in [4].

This technique is very similar to the arrangement construction technique by Bentley and Ottmann [1] but does not produce the arrangement. Instead at every event point (boundary intersection or boundary endpoint) the information on the number of containment regions is updated. Since this information is represented as a single number at each elementary interval this is possible in $O(\log nm)$ time per event.

Since each rotation diagram contains at most $O(nm)$ boundaries and an arrangement complexity of $O(n^2m^2)$ this algorithm runs in $O(n^2m^2 \log nm)$ time and $O(nm)$ space per diagram. Since $O(nm)$ such diagrams exist, the program runs in $O(n^3m^3 \log nm)$ time and $O(nm)$ space.

## 5 Conclusions

The algorithms presented here extend rotation diagrams from use on convex specific problems to the non-convex case. The improvement of this technique yields a deterministic performance of $O(n^3m^3 \log nm)$ versus the previous best algorithm's $O(n^4m^3 \log m)$. Furthermore we present an algorithm which runs in $O(n^3m^3)$ expected time which is $O(1)$ expected time per possible stable placement. While the analysis provided accounts only for the worst case number of stable placements the algorithm is actually sensitive to the input and only examines the actual stable placements. Unfortunately the analysis of the sensitivity to inputs is beyond the scope of this abstract and is left to the full version of the paper. The technique described here is also extendible to placement of polygonal annuli and other polygonal planar objects.

## References


[1] J. Bentley and T. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computing*, C-28:643–647, 1979.

[2] P. Bose, A. Lubiw, and J. Munro. Efficient visibility queries in simple polygons. *Computational Geometry: Theory and Applications*, 23(3):313–335, 2002.

[3] B. Chazelle. The polygon placement problem. In F. Preparata, editor, *Advances in Computing Research*, volume 1, pages 1–34. JAI Press, 1983.

[4] M. Dickerson and D. Scharstein. Optimal placement of convex polygons to maximize point containment. *Computational Geometry: Theory and Applications*, 11(1):1–16, Aug. 1998.

[5] D. Eppstein and J. Erickson. Iterated neatest neighbors and finding minimal polytopes. *Discrete and Computational Geometry*, 11:321–350, 1994.

[6] L. Guibas, R. Motwani, and P. Raghavan. The robot localization problem. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *Algorithmic Foundations of Robotics*, pages 269–282. A K Peters Ltd., 1995.

[7] J. Morrison. *Geometric placement problems.* PhD thesis, School of Computer Science, Carleton University, 2002.

[8] K. Mulmuley. A fast planar partition algorithm II. *Journal of the ACM*, 38(1):74–103, Jan. 1991.