Stabbing Polygons by Monotone Chains

Laxmi P. Gewali

Department of Computer Science
University of Nevada, Las Vegas
Las Vegas, NEVADA 89154

Abstract. Consider the problem of computing a stabber for polygonal objects. Given a set of objects S, an object that intersects with all of them is called the stabber of S. Polynomial time algorithms for constructing a line segment stabber for polygonal objects, if one exists, have been reported in the literature. We introduce the problem of stabbing polygonal objects by monotone chains. We show that a monotone chain that stabs the maximum number of given obstacles can be computed in $O(n^2 \log n)$ time. We also prove that the maximum number of monotone chains required to stab all polygons can be computed in $O(n^{2.5})$ time. The main tool used in developing both results is the construction of a directed acyclic graph induced by polygonal objects in a given direction. These results have applications for planning collision-free disjoint paths for several mobile robots in a manufacturing environment.

I. Introduction.

An important combinatorial problem having several applications in robotics and computational geometry is the construction of a stabber for a given set of objects [GS90, EM*82]. For example, algorithms for stabbing parallel line segments with a convex polygon can be used for forming an image of parts moving in a conveyor belt [GS90]. In a typical setting of the problem we are given a set of objects, and our goal is to construct, if possible, an object that intersects with all objects in the set. Many solutions have been developed for specialized versions of the problem. Edelsbrunner *et al* [EM*82] developed an $O(n \log n)$ algorithm for stabbing general line segments by a straight line. Algorithms for stabbing polygonal objects by line segments have also been considered. Atallah and Bajaj [AB87] have given a sub-quadratic algorithm that finds a straight line stabber for a given set of polygonal objects. In general, it may not be possible to stab all given objects by a stabber. In such situations a stabber that stabs the maximum number of objects is defined. An $O(n \log n)$ algorithm for computing a line that stabs the maximum number of given line segments is reported in [EG86]. A related result is given in [JK90].

In this paper, we consider the problem of stabbing polygonal objects by monotone chains. Note that a chain (simple path) is said to be monotone with respect to a given line ℓ if lines orthogonal to ℓ intersect the chain in at most one point. The first problem we consider is the problem of computing a monotone chain (with respect to a given direction) that stabs the maximum number of polygons. The

This research is supported in part by grants from Cray Research Inc.

second problem is the computation of the minimum number of monotone chains that can stab all polygons. One of the main motivations for considering these problems is due to their applications for planning collision-free monotone paths for mobile robots in a manufacturing environment [ACM89]. In section II, we prove that a monotone chain that stabs the maximum number of polygons can be computed in $O(n^2 \log n)$ time. In section III, we show that the minimum number of monotone chains required to stab all polygons can be computed in $O(n^{2.5})$ time. We conclude by discussing extensions and applications of these problems.

II. Stabbing the maximum number of polygons.

Consider a two dimensional scene consisting of disjoint polygons $P_1, P_2, \ldots, P_m$. Let n be the size of the scene. Here, by size of the scene we mean the total number of edges in all polygons. Let ℓ_i denote the left most point of P_i (the point of P_i having the smallest x-coordinate). Similarly, the right most point r_i of P_i is defined.

An **x-monotone chain** is a chain of line segments having monotonicity along x-axis. An x-monotone chain C is called a **stabber** of the polygon P_i if (i) C enters P_i from the point ℓ_i and leaves from the point r_i, and (ii) the portion of C between ℓ_i and r_i lies wholly within O_i (Figure 1).

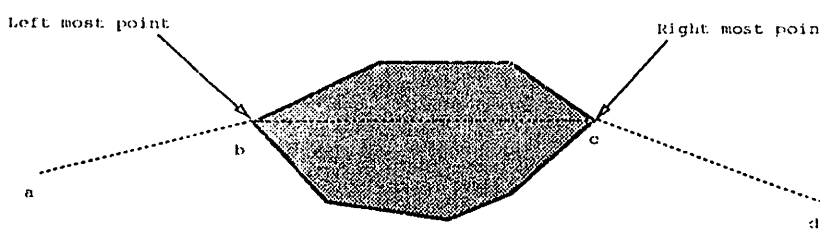


Figure 1: Illustrating the Stabber of an Obstacle

We are interested in computing a x-monotone chain that stabs all polygons. Such a stabber may not always exist. In Figure 2 only two polygons can be stabbed by any x-monotone chain. We, therefore, define the **maximal stabber** to be an x-monotone chain that stabs the maximum number of polygons. We call the problem of computing such a stabber the **maximal stabber problem** (MSP).

Note that if a polygon is not convex then no monotone chain may stab it. Hence, in the sequel we assume that all polygons are convex.

Our approach to solve MSP is to convert an instance of MSP to an instance of finding the longest path between two nodes in a directed acyclic graph. We define two points in free space to be **mono-connectable** if they can be connected by a collision-free x-monotone chain. The directed graph $G(V, E)$ induced by the collection of polygons is constructed as follows. The vertices in V correspond

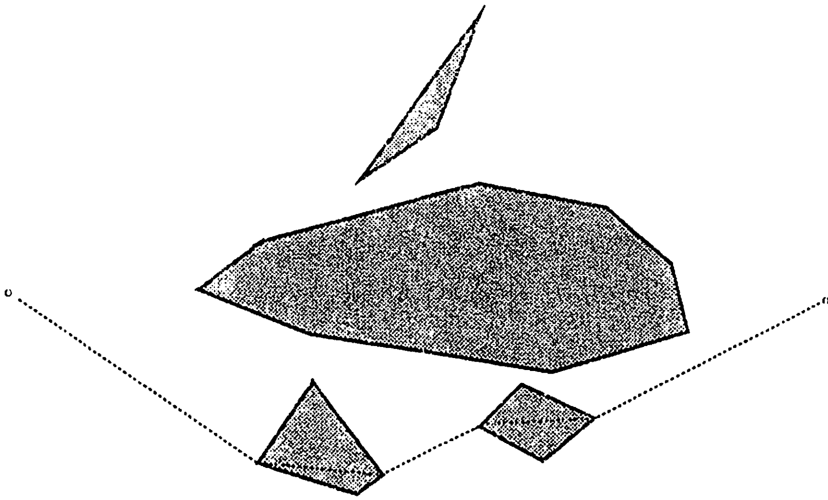


Figure 2: Illustrating Maximal Stabber

to each polygon in the scene (that is, $V = \{v_i \mid O_i$ is an polygon$\}$). Vertices v_i and v_j are connected by a directed edge $\overline{v_i, v_j}$ (directed from v_i to v_j) if (i) the x-coordinate of r_i is less than the x-coordinate of ℓ_j and (ii) r_i and ℓ_j are mono-connectable. Figure 3 shows an example of such a construction.

Lemma 1. *The directed graph $G(V, E)$ induced by polygons (as defined above) can be constructed in $O(n^2 \log n)$ time.*

Proof: The left most points ℓ_i's and the right most points r_i's of each polygon can be determined in $O(n)$ time by simply scanning the boundary of each polygon. Collision-free x-monotone paths connecting a vertex of a polygon to all other vertices can be found in $O(n \log n)$ time by using the monotone path planning algorithm given in [ACM86]. Using this algorithm at most n times we can obtain all edges of the DAG in $O(n^2 \log n)$. ∎

Lemma 2. *The directed graph $G(V, E)$ as constructed above is acyclic, that is, it does not contain any cycle.*

Proof: Assume to the contrary that a cycle exists in the graph. Consider monotone chains C_i and C_{i+1} corresponding to successive edges e_i and e_{i+1}, respectively, (e_i is the predecessor of e_{i+1}) in the cycle. By construction, chain C_{i+1} lies entirely to the right of chain C_i. Thus, the monotone chain C_j corresponding to any edge e_j that can be encountered by following the directed path starting at C_i must entirely lie to the right of C_i. If we traverse the cycle, starting at C_i, we eventually encounter C_i, which implies that C_i lies entirely to the right of itself — a contradiction. ∎

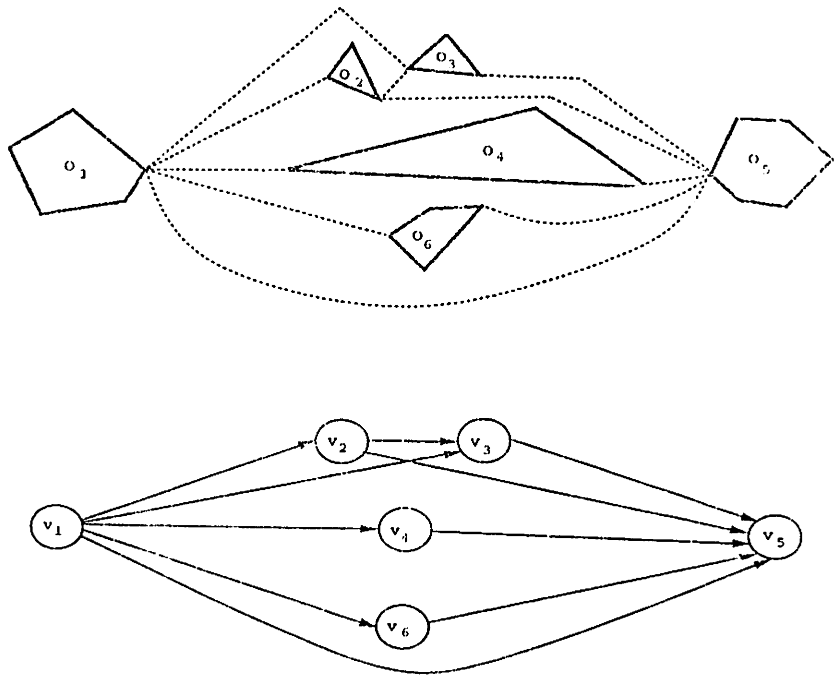


Figure 3: Illustrating the Construction of DAG

Two x-monotone stabbers are said to be **equivalent** if they stab the same set of polygons in the same order. The following Lemma directly follows from the construction of the directed graph.

Lemma 3. *For each x-monotone stabber that stabs polygons in the order O_{i_1}, $O_{i_2}, \ldots, O_{i_m}$ there is a corresponding directed path in the directed graph connecting the vertices in the order $v_{i_1}, v_{i_2}, \ldots, v_{i_m}$.*

Observe that all equivalent stabbers are mapped to the same directed path in the DAG. Hence, a maximal stabber corresponds to a longest path in the DAG (the length of the path is the number of edges in the path).

Theorem 1. *An x-monotone chain that stabs the maximum number of given polygons can be computed in $O(n^2 \log n)$ time.*

Proof: The directed graph can be constructed in $O(n^2 \log n)$ time (by Lemma 1). Finding a longest path in a directed acyclic graph (DAG) is like topological sorting and can be computed in $O(|E| + |V|)$ time [AHU74]. ∎

III. Stabbing all polygons.

Since it may not always be possible to stab all polygons by a single monotone chain we are motivated to define the following problem.

Stabber Cover Problem (SCP). *Find the minimum number of disjoint x-monotone chains to stab all polygons such that no polygon is stabbed by more than one chain.*

A minimum path cover of the induced DAG gives the minimum number of x-monotone chains that stab all polygons. Stabbing chains obtained in this way may stab the same polygon more than once. We, therefore, have to make use of the minimal vertex disjoint path cover of the induced DAG . In this way, we make sure that every polygon is stabbed exactly once. Still, the chains may intersect in the interior of their segments. If that happens we can introduce a new vertex at each point of intersection and make the resulting chains disjoint. Consider chains C_1 and C_2 whose intersecting segments are $\overline{a,b}$ and $\overline{c,d}$. Let I be the point of intersection. We modify these segments into two chains by introducing a new vertex I_1 as shown in Figure 4. By modifying each intersection point in this way, the required disjoint chains are obtained. Note that the total number of chains is not altered by this construction. Figure 5 shows the construction of disjoint chains from intersecting ones.

Theorem 2. *The minimum number of disjoint x-monotone chains required to stab all polygons can be obtained in $O(n^{2.5})$ time.*

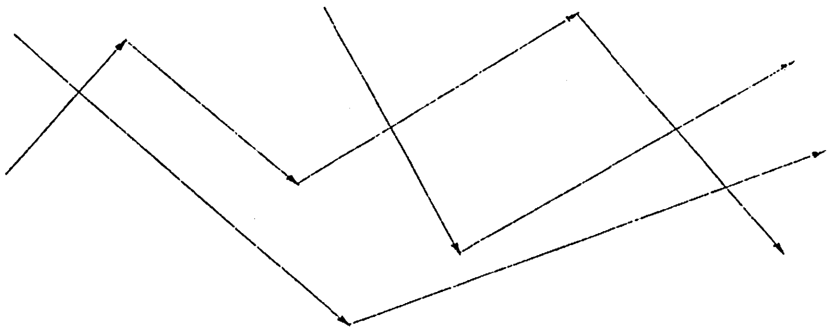
Proof: The DAG induced by the collection of polygons with respect to the x-axis can be computed in $O(n^2 \log n)$ (Lemma 1). A minimal vertex disjoint path cover of any DAG can be computed by using bipartite matching in $O(n^{2.5})$ [BG77]. Intersecting chains can be modified to disjoint chains by sweeping the plane by a vertical line in $O(n \log n + K)$ time [PS85], where K is the total number of intersections between chains. ∎

Figure 4: Modifying Crossing Segments

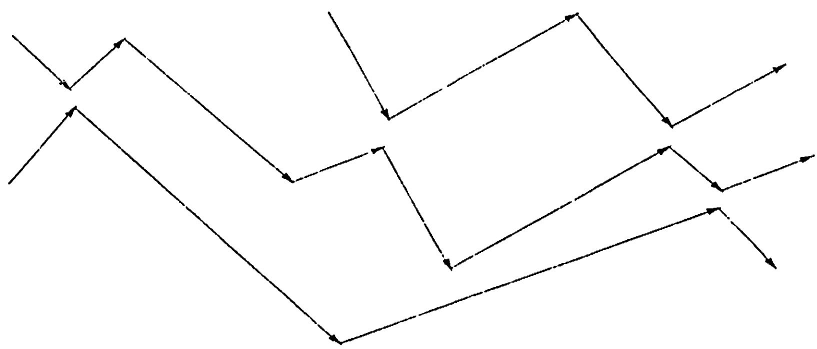
IV. Discussions.

We proved that a maximal stabber having monotonicity in a given direction can be computed in $O(n^2)$ time. We also established that the minimum number of monotone chains required to stab all obstacles can be computed in $O(n^{2.5})$.

Computation of collision-free paths has been an active area of research in robotics [SY87]. Recently, the importance of planning monotone collision-free paths has



(a): Intersecting Chains



(b): Disjoint Chains

Figure 5: Construction of Disjoint Chains from Intersecting Chains

been pointed out by several authors [ACM89, KS90]. Consider the cost of traveling along a path. If the cost of travel is charged only in a given direction then a path having monotonicity in that direction minimizes the total cost of the travel [ACM89]. In situations where we want to avoid turns, monotone paths are better (monotone paths tend to have lesser number of turns). Monotone paths have also applications in robotics for deciding the separability of polygonal objects [T85].

Imagine a manufacturing environment consisting of several work-sites. Mobile robots are required to visit the work-sites to perform the assigned tasks. Robots can enter a work-site from a fixed entrance point and leave from a fixed exit point. If polygons are used to model work-sites then a stabber of the polygons gives a path to be followed by a robot to visit the sites. The first result of this paper (Theorem 1) can be used to plan collision-free paths so that a robot can be instructed to perform tasks in the maximum number of sites and yet keep the path monotone. The second result (Theorem 2) can be used to plan the motion of several robots amidst the work-sites. If the paths are disjoint then the problem of scheduling and coordination is eliminated.

We showed how to compute a maximal stabber having monotonicity in a given direction. More polygons may be stabbed if some other direction is considered for monotonicity. It would be interesting to solve the problem by allowing the stabbing chain to have monotonicity in any direction.

Acknowledgement

The author would like to thank the anonymous referee for many helpful suggestions.

References

AB87 M.J. Alallah and C. Bajaj, *Efficient algorithms for common traversals*, Information Processing Letters 25 (1987), 87–91.

ACM89 E.M. Arkin, R. Connelly, and J.S.B. Mitchell, *On monotone paths among obstacles, with applications to planning assemblies*, Proc. 5th Annual Symposium on Computational Geometry (1989).

AHU74 A.V. Aho, J.E. Hopcroft, and J.D. Ullman, "The Design and Analysis of Computer Algorithms", Addison Wesley Publishing Company, 1974.

BG77 F.T. Boesch, and J.F. Gimpel, *Covering the points of a digraph with the point-disjoint paths and its applications to code optimization*, Journal of the ACM 24, No. 2 (1977), 192–198.

EG86 H. Edelsbrunner and L.J. Guibas, *Topologically sweeping an arrangement*, Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computation (1986), 389–403.

EM*82 H. Edelsbrunner, H.A. Maurer, F.P. Preparata, A.L. Rosenburg, E. Wezel, and D. Wood, *Stabbing line segments*, BIT 22 (1982), 274–281.

GS90 M.T. Goodrich and J. Snoeyink, *Stabbing parallel segments with convex polygons*, Computer Vision, Graphics, and Image Processing 49 (199), 152–170.

JK90 B. Jones and Yan Ke, *Maximal segment stabbing of segments in the plane*, Proceedings of the Second Canadian Conference in Computational Geometry (1990).

KS90 S. Kundu and R. Sridhar, *An* $O(kN \log N)$ *algorithm for decomposing a set of polygons into D-separable components*, Pattern Recognition 23 (1990), 735–743.

PS85 F.P. Preparata and M.I. Shamos, "Computational Geometry — An Introduction", Springer-Verlag, 1985.

SY87 J.T. Schwartz and C. Yap (Editors), "Algorithmic and Geometric Aspects of Robotics", Lawrence Erlbaum Associates, 1987.

T85 G.T. Toussaint, *Movable separability of sets*, in "Computational Geometry", Ed. by G.T. Toussaint, North-Holland Publishing Company, 1985.