# SWEEPING POLYGONS WITH A VARIABLE-LENGTH LINE SEGMENT

**Kien C. Huynh**
Department of Computer Science
Stony Brook University
USA
kchuynh@cs.stonybrook.edu

**Joseph S. B. Mitchell**
Department of Applied Mathematics and Statistics
Stony Brook University
USA
joseph.mitchell@stonybrook.edu

**ABSTRACT**

In this abstract, we consider the problem of sweeping a polygonal domain $P$ using a mobile, variable-length line segment. A point in the domain is swept if it comes into contact with any point of the line segment. Subsequently, a domain $P$ is completely swept if every point in it is touched by the line segment at least once. The main objective of this problem is to find a sweeping schedule that sweeps every point in the domain as quick as possible. During the entire solution, we require the line segment to always stay connected. An 8-approximation algorithm is provided when $P$ is a simple polygon. For a polygon with holes, we prove that this problem is NP-hard and provide a constant-factor approximation algorithm.

## 1 Introduction

Given a polygonal domain $P$, possibly with a set of polygonal obstacles $\mathcal{H}$, we want to completely sweep every point in $P$, including points on the boundary $\partial P$, using a line segment, $ab$, whose endpoints move within $P$ while remaining visible to each other. A point in $P$ is swept if it touches the (variable length) line segment at least once. At the beginning, we assume (for now) points $a$ and $b$ coincide at a given point in $P$. Points $a$ and $b$ move with variable speed, but with maximum speed 1. Our objective is to determine trajectories for points $a$ and $b$ in order to minimize the time it takes for the segment $ab$ to sweep $P$, i.e. minimizing the makespan. Given any complete sweeping schedule, this objective is equivalent to minimizing the maximum time traveled by $a$ or $b$.

Related to the above problem is that of sweeping a polygon with a chain of guards introduced in [1]. The sweeping mechanism here is similar to the problem statement given above, but there can be more than just two endpoints and they are required to formed a chain with two end guards restricted to stay on the boundary of $P$. Because of this particular restriction, the chain can always maintain an uncontaminated region which is helpful in the context of pursuing a evading target in the domain. If the goal is to minimize the maximum distance between two guards or the sum of the distance traveled, a schedule can be computed in polynomial time [4][3]. Another closely related problem is the watchman route problem, [2], which involves finding a minimum tour or path of a point guard in a polygon so that he sees everything.

## 2 Sweeping a simple polygon

**Theorem 2.1.** *Given a simple polygon $P$, we can, in polynomial time, compute a sweeping schedule with makespan at most 8 times that of an optimal sweeping schedule.*

Our algorithm relies on the following lemmas:

**Lemma 2.1.** *To sweep a convex vertex, at least one endpoint of the sweeping line segment must visit it.*

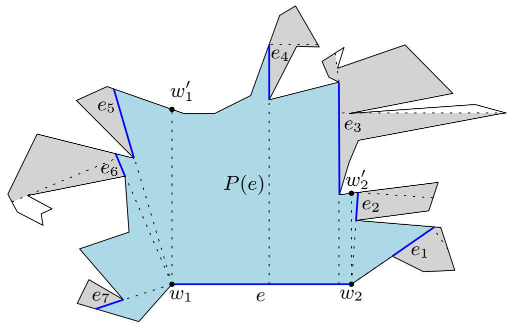



Figure 1: The light blue region is the special visibility polygon $P(e)$ with base edge $e$. The additional chords $e_1, e_2, e_3$, etc., also have visibility polygons (grey) associated with them.

**Lemma 2.2.** *Let $\pi_r = (v_i, v_{i+1}, \dots v_j), i < j$ be the longest induced subpath of $\partial P$ between two consecutive convex vertices $v_i, v_j$. If $j > i+1$ then $\pi_r$ also includes the chain of reflex vertices between $v_i$ and $v_j$. Then,*

$$OPT \geq \frac{1}{2}|\partial P \setminus \pi_r|,$$

*where $OPT$ is the optimal makespan.*

Given that $a$ and $b$ start at the same point initially, the union of their paths in any solution cannot be shorter than the minimum TSP path that visits all convex vertices, i.e. $\partial P \setminus \pi_r$.

Our algorithm first partitions the polygon into subpolygons, each of which has a simple sweeping strategy. The partition results in a set of subpolygons whose dual is a tree, which allows the sweeping line segment to sweep each subpolygon, traversing the tree recursively. To understand the partition scheme, consider any edge $e = w_1 w_2$ of the polygon. We first compute the histogram polygon with base $e$, $HP(e)$, which is defined as the union of all interior chords perpendicular to $e$ having an endpoint on $e$. Let $(w_1, w_1')$ be such a chord anchored at $w_1$. We then compute the visibility polygon at point $w_1$ restricted to be left of $(w_1, w_1')$; call it $VP(w_1)$. Similarly, we compute another visibility polygon to the right of $(w_2, w_2')$, called $VP(w_2)$. Then $P(e) = HP(e) \cup VP(w_1) \cup VP(w_2)$ is a subpolygon in our partitioning scheme. See Figure 1 for an example of $P(e)$. Notice that the boundary of $P(e)$ includes some new "shadow" chords, each will be used as the histogram base for the subsequent subpolygons in the partition. The process is complete when no shadow chord is created. For the first subpolygon, we find the longest reflex chain of $P$ and compute histogram polygons for all edges on this chain, along with the visibility polygons of the vertices of the chain but with angles restricted by the two perpendicular chords of the two edges incident to it. The union of them will be the first subpolygon in our partition.

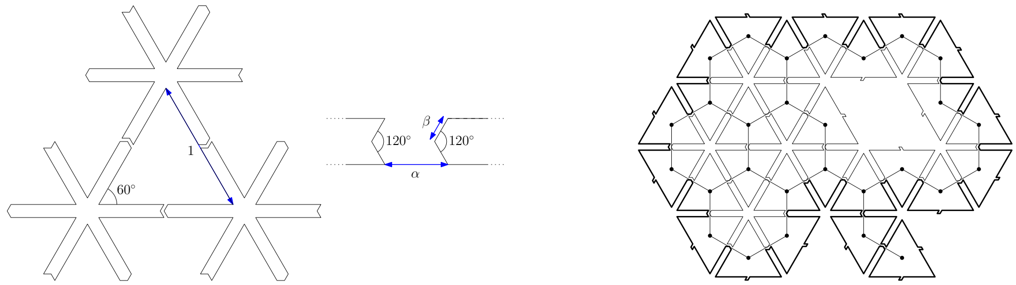
The schedule begins with the line segment rotating clockwise with one endpoint anchoring at $w_1$ until it lines up with $(w_1, w_1')$, this will sweep $VP(w_1)$. To sweep $HP(e)$, one endpoint will move along the top polygonal path of it while the other move along $e$. Finally, the segment will rotate clockwise around $w_2$ to sweep $VP(w_2)$ and move to the nearest chord of an adjacent unswept subpolygon. It costs $|\partial P(e) \setminus e|$ to sweep $P(e)$. The sweeping schedule continues by visiting adjacent subpolygons in a depth-first order until all of them are swept. This whole process incurs a cost that is double of the sum of perimeters of all subpolygons, minus the length of $\pi_r$. With the established partition algorithm the total length of the additional chords is no longer than $|\partial P \setminus \pi_r|$. Therefore, the final makespan is $4|\partial P \setminus \pi_r| \leq 8OPT$. The straightforward running time of the partitioning is $O(n^2)$; with more care, it can be computed in time $O(n)$ (as in computing window partition trees). This is because each time $P(e)$ is computed, we eliminate all reflex vertices seen by it and creating an equal amount of chords. The cost of computing each visibility polygon is $O(n)$.

## 3 Polygon with holes

### 3.1 Hardness

**Theorem 3.1.** *The problem of finding an optimal sweeping schedule in a polygon with holes is NP-hard.*

We give a summary of our reduction from Hamilton path in hexagonal grid graph. To prove the hardness, we employ a polygonal gadget illustrated in Figure 2a. Each hole has six thin blades, each tip has two edges of length $\beta$ and together they either form a reflex or convex vertex. Three holes can be aligned so that the long edges of their blades form an equilateral triangle with side length 1, and the tips of their blades make arbitrarily thin gaps of length $\alpha$. By combining multiple hexafans and merging them when needed, a polygon can be built from any hexagonal grid graph See Figure 2b for such an example. The main result we want to prove is the following:




(a) (b)

Figure 2: On the left is the hexafan gadget we used in our NP-hardness proof. On the right is a polygon constructed from a hexagonal grid graph using the hexafan gadget. The original grid graph is superimposed on top of the polygon.

**Lemma 3.1.** *Given a polygon constructed from any hexagonal grid graph with $N$ vertices, there exists a sweeping solution of makespan $2N(1-\alpha+\beta)+\alpha-\beta$ if and only if there is a Hamilton path in the original graph.*

The intuition here is that it takes at least $2(1-\alpha+\beta)$ for the line segment to enter a triangle, sweep it, and move on to another triangle. Because of the way the gaps are constructed, the line segment cannot be in two different triangles so it is impossible to sweep more than one triangle at the same time. If the segment enters and leaves the same triangle multiple times it will add at least $0.5-\alpha+\beta$ to the makespan, which exceeds the upper bound $2N(1-\alpha+\beta)+\alpha-\beta$ by a noticeable margin since $\alpha$ and $\beta$ can be arbitrarily small.

### 3.2 Approximation algorithm

In this subsection we give a brief overview of our approximation algorithm. The algorithm involves two phases. First, we compute a minimum spanning tree, $MST^*$, within $P$ that spans all connected components of the boundary, $\partial P$, of $P$ (both the holes and the outer boundary). This tree, along with the boundaries, can be seen as bounding a degenerate simple polygon $P'$. In the second phase, we apply the algorithm from Section 2 to compute a sweeping solution for $P'$.

**Theorem 3.2.** *For a polygon $P$ with holes, there is a polynomial-time algorithm to produce a sweeping schedule with makespan $O(OPT)$.*

We utilize the following lemmas:

**Lemma 3.2.** *Let $c$ be any interior chord in $P$, $a_1b_1$ be a configuration where $a_1b_1 \cap c = \emptyset$, and $a_2b_2$ be a configuration where $a_2b_2 \cap c = x, x \neq a, x \neq b$, then to move between the two configurations, at least one endpoint has to cross $c$.*

**Lemma 3.3.** *Given any arbitrary interior chord $c$ in $P$, to completely sweep $P$, at least one endpoint of the line segment $ab$ has to touch $c$.*

The proof of Theorem 3.2 relies on a lower bound on $OPT$: we show that $OPT$ must be $\Omega(|\partial P'|)$; $\partial P'$ includes the boundaries of the holes and $P$ and the edges of the spanning tree. To prove it, we first notice that any complete sweeping schedule $\pi$ consists of a pair of paths, for the endpoints of the sweeping segment; this pair of paths forms an arrangement that partitions the interior of $P$ into faces $F_1, F_2, \ldots$ Each face $F_i$ could contain one or more holes. Let $CH_1$ be the geodesic convex hull of all holes $H_j$ w.r.t. the boundary of $F_i$. We observe that the boundary of $CH_1$ has to collapse onto itself at some point, otherwise it is impossible to sweep the holes in $F_i$. If $CH_1$ does not collapse onto itself, then apart from the sections where it touches the supporting holes, its boundary will be made of reflex chains (Figure 3). By definition, the endpoints of the line segment never enters $F_i$, i.e. the line segment has to sweep all holes in $F_i$ while both endpoints are moving outside of it. Because $F_i$ is itself a polygon, $[a, b]$ must intersect at least two reflex chains.

Let $H_j$ be any hole in $F_i$, there are three main cases where $[a, b]$ can sweep it as illustrated in Figure 4. In case (a) and (b), the line segment has to be intersecting with some reflex chains and then not so that it can sweep all sides of $H_j$. To achieve this, $a$ or $b$ must either cross $\partial F_j$ or make a discontinuous jump over some holes which is not possible. In case (c), where all of the extensions of the edges of $H_j$ intersect the same pair of reflex chain, it still requires $[a, b]$ to make a discontinuous movement over $H_j$. Since it is impossible to sweep $H_j$ with such a geodesic convex hull, the boundary of $CH_1$ must collapse onto itself at some point(s), splitting the set of holes in $F_i$ into two or more subsets, each lies inside a separate pocket of $CH_1$. Let $CH_2, CH_3, \ldots$ be the geodesic convex hulls of each of these sets of holes w.r.t. $F_i$. They can be seen as children of $CH_1$. Notice that their total perimeter $|\partial CH_2| + |\partial CH_3| \ldots \leq |\partial CH_1|$. Now, for any convex hull $CH_k$ we can reapply the same argument as above which creates children of $CH_k$ whose sum of

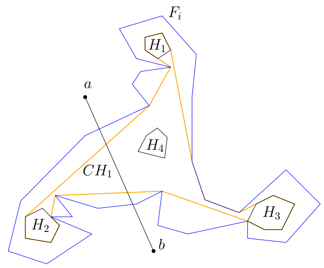



Figure 3: A face $F_i$ in the arrangement of $\pi$. The geodesic convex hull, $CH_1$, of all holes is colored in orange.

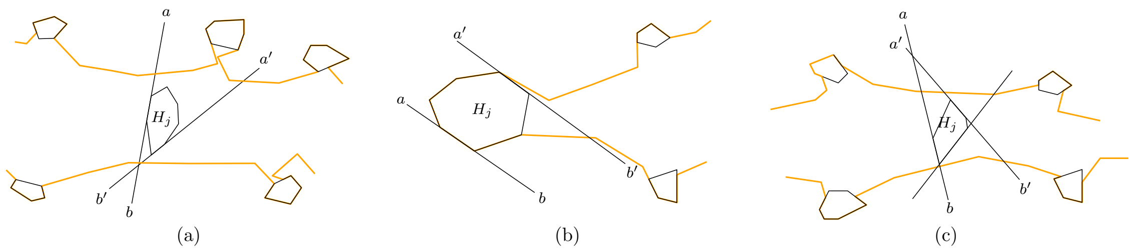



Figure 4: Different cases where a the line segment has to make discontinuous movements to sweep all sides of a hole $H_j$. In all cases, it is not possible to move from $[a, b]$ to $[a', b']$ and vice versa without making discontinuous movements.

perimeter is also less than or equal to $|\partial CH_k|$. We want to repeat this until each convex hull has either one or two holes in it. In the first case, it is trivial to see that $|\partial H_j| \leq |\partial CH_k|$ where $H_j \in CH_k$. When there are exactly two holes left, their geodesic convex hull must collapse onto itself as well, due to Lemma 3.3 forcing $OPT$ to block any straight chord between the two holes. Therefore, the total perimeter of the two holes is at most as large as $|\partial CH_k|$. At the end, we have a set of geodesic convex hulls $\mathcal{S}$, each containing one or two holes of $P$ whose total perimeter obeys the inequality: $\sum_{H \in CH, \forall CH \in \mathcal{S}} |\partial H| \leq \sum_{CH \in \mathcal{S}} |\partial CH| \leq |\partial CH_1| \leq |\partial F_i|$. By noticing that the common boundary between any two adjacent faces in the arrangement is shared only by those two, we can derive the lower bound:

**Lemma 3.4.**

$$\sum_{H \in \mathcal{H} \cup \{P\}} |\partial H| \leq 2|\pi^*| \leq 4OPT$$

With the above reasoning, any convex hole $H_j$ belongs to its own pocket of some geodesic convex hull $CH_k$ when $CH_k$ collapses onto itself. Because $CH_k$ is relative w.r.t. $F_i$, it means that $F_i$ also comes into contact with a point $x_j$ where the collapsing occurs. Therefore $F_i$ must also touch the pocket of $CH_k$ containing $H_j$ at $x_j$. We can then connect each hole $H_j$ to its corresponding $x_j$ by adding a new edge. This edge has length less than $0.5|\partial CH_k|$. For non-convex holes, $\pi^*$ must visit all of their convex vertices. With the observation that $\pi^*$ and the new edges serve as a tree spanning boundaries of $P$, we can conclude the following which proves our theorem about the approximation algorithm:

**Lemma 3.5.**

$$|MST^*| \leq 2|\pi^*| \leq 4OPT$$

## References

[1] Alon Efrat, Leonidas J Guibas, Sariel Har-Peled, David C Lin, Joseph S B Mitchell, and T M Murali. Sweeping simple polygons with a chain of guards. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 927–936, 2000.

[2] Joseph SB Mitchell. Approximating watchman routes. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 844–855. SIAM, 2013.

[3] Xuehou Tan and Bo Jiang. Optimum sweeps of simple polygons with two guards. In *International Workshop on Frontiers in Algorithmics*, pages 304–315. Springer, 2010.

[4] Xuehou Tan and Bo Jiang. Minimization of the maximum distance between the two guards patrolling a polygonal region. *Theoretical Computer Science*, 532:73–79, 2014.