Finding an Approximate Minimum-Link Visibility Path Inside a Simple Polygon

Muhammad H. Alsuwaiyel
Department of Information and Computer Science
King Fahd University of Petroleum & Minerals
Dhahran 31261, Saudi Arabia
e-mail:facp079@saupm00.bitnet

D.T. Lee[1]
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208
e-mail: dtlee@eecs.nwu.edu

Keywords: Computational geometry, approximation algorithm, simple polygon, visibility path, minimum link path.

1 Introduction

In [1] Alsuwaiyel and Lee showed that computing a minimum link path π inside a simple polygon with n vertices such that the interior of the polygon is weakly visible from π is NP-hard. The authors also presented an $O(n^3 \log n)$ time algorithm for producing an approximate solution that was claimed to have no more than 3 times the number of links of an optimal solution. In [2] Arkin et al. show that the problem of finding a minimum-link watchman route in polygonal domains (with holes)[2] is NP-complete and give a polynomial-time approximation algorithm with performance ratio of $\log n$. In fact the algorithm in [1] gives a feasible solution with no bound guarantee. Here we describe in a more precise way the approximation algorithm that constructs a watchman path as well as a tour in time $\max\{m^2, mn\} = O(n^2)$, where m is the link-length of an optimal path, and show that indeed a constant performance ratio of no more than 4 can be attained in a simple polygonal domain (i.e., no holes). We also show that the performance ratio can be improved to 3.5 with time bound $\max\{m^3, mn\} = O(n^3)$.

2 Preliminaries

Let x be any reflex vertex adjacent to another vertex u in a simple polygon P. Let H be the infinite half line originating at x in the direction from u to x. Let y be the intersection of H and the boundary of P closest to x. Then, the *directed* line segment $w = \overline{xy}$ is called a *window*

[1] Supported in part by the National Science Foundation under the Grant CCR-9309743, and by the Office of Naval Research under the Grant No. N00014-93-1-0272.

[2] The Minimum-link watchman route problem is to find a *tour*, instead of a *path*, such that the entire polygonal domain is weakly visible.

and $\overline{ux}$ the edge *generating* w. Its *start* and *end* points will be denoted by $\mathcal{X}(w)$ and $\mathcal{Y}(w)$, respectively. If w is a window generated by $\overline{ux}$, then it partitions P into two parts and that part which contains u is called the *region of* w, denoted by $P(w)$. If w_1 and w_2 are two windows such that $P(w_1)$ is contained entirely within $P(w_2)$, then w_2 is called *redundant*, otherwise it is *nonredundant*. Since we will be working mainly with nonredundant windows, the qualifier "nonredundant" will be dropped throughout, unless a distinction is explicitly made.

Given a simple polygon P, let $\mathcal{W}$ denote its set of nonredundant windows. The windows in W, $W \subseteq \mathcal{W}$, are called *independent* if $\forall\, w, w' \in W, P(w) \cap P(w') = \phi$. W is a *maximal independent set of windows* if $\forall\, w \in \mathcal{W} - W\ \exists\, w' \in W$ such that w and w' are not independent. A set of windows H is said to be a *maximum independent set of windows* if it is a maximal independent set of windows of maximum cardinality.

A *visibility path* π for a given simple polygon P is a polygonal path contained in P with the property that any point in P is visible from at least one point on π. We say that P is weakly visible from π. The following theorem and lemma are basic to finding a visibility path. The proof of the theorem is a direct generalization of the case when the visibility path is a line segment. The proof of this special case can be found in Ke[8]. The problem of finding a shortest visibility segment can be solved in linear time[3] improving the previously known $O(n \log n)$ time algorithm due to Ke[8].

Theorem 1 *Let π be a polygonal path inside a simple polygon P. Then, π is a visibility path for P if and only if it intersects all the regions $P(w)$ of nonredundant windows w inside P.*

We will assume throughout that no three vertices in P are colinear. This assumption is essential in order to guarantee a constant performance ratio.

Lemma 1 *If no three vertices of P are colinear, then no line segment inside P intersects with more than two independent windows.*

Proof. Let $L = \overline{ab}$ be a line segment inside P that intersects with more than two independent windows. Without loss of generality, we may assume that for some two independent windows w_1 and w_2 that L intersects with, $a \in P(w_1)$ and $b \in P(w_2)$. Let w_3 be another independent window intersected by L. Since $P(w_1), P(w_2)$ and $P(w_3)$ are pairwise disjoint, L cannot intersect with the interior of $P(w_3)$. Moreover, if $L \cap w_3$ contains more than one point, then we must have $L \subseteq w_3$ since w_3 is a chord in P. Consequently, $L \cap w_3$ consists of exactly one point, say c. It is not hard to see that the only possibility is that $c = \mathcal{Y}(w_3)$ and indeed a *reflex* vertex of P. But this implies that c and the endpoints of the edge generating w_3 are colinear.

Corollary 1 *Let $H \subset \mathcal{W}$ be any* independent set of windows*. Then any window $w \in \mathcal{W}$ intersects with at most two windows in H.*

Corollary 2 *Let* $M \subseteq \mathcal{W}$ *be a* maximum independent set of windows. *Then the link-length of any optimal visibility path* π *is at least* $|M| - 1$.

Proof. By Theorem 1, π must intersect all members of M. By Lemma 1, at least one line segment is needed in order for π to go from one window in M to the next.

The idea of finding an approximate minimum-link visibility path inside P consists of computing a set of points, which we will call *special points*. These points have the property that if a polygonal path π passes through all of them then, as will be shown later, it is a visibility path. In Section 3 we show how to compute a set S of special points. In Section 4, we give the approximation algorithm for computing an approximate visibility path.

3 Construction of the special points

The following procedure constructs a maximum independent set of windows. It is essentially the same as that given in [9]. The proof that this algorithm does produce an independent set of maximum cardinality is similar to that of finding a maximum independent set of circular-arcs, given in [9, 10] and is omitted. Pick an arbitrary window w and let it be numbered w_1. Recall that the endpoints of a window w_i are $\mathcal{X}(w_i)$ and $\mathcal{Y}(w_i)$. Assume that the endpoints are sorted in counterclockwise direction. The endpoint of a window w visited the first time as we scan them in this direction is designated as $\mathcal{X}(w)$. Let *next* be a function from $\mathcal{W}$ to itself such that $next(w_i) = w_j$, where $\mathcal{X}(w_j)$ lies after $\mathcal{Y}(w_i)$, and between $\mathcal{Y}(w_i)$ and $\mathcal{X}(w_j)$ there exists no endpoint, $\mathcal{X}(w_k)$ for any k.

1. Let $w = w_1$ and set $M \leftarrow \{w\}$.
2. Set $w' \leftarrow next(w)$.
3. While $w' \neq w$ and $w' \cap w = \emptyset$ do the following:
   Set $M \leftarrow M \cup \{w'\}$; $w' \leftarrow next(w')$.
4. If $w' = w$ then halt.
5. Set $M \leftarrow M - \{w\} \cup \{w'\}$.
6. Set $w \leftarrow next(w)$; $w' \leftarrow next(w')$ and go to Step 3.

The above procedure finds a set of independent windows in a greedy manner. It will terminate when function *next* maps the set of windows in M to itself. Note that when all windows intersect, $\forall w \in \mathcal{W}$, $next(w) = w'$ for some $w' \in \mathcal{W}$, and M consists of exactly one window, i.e., $\{w'\}$. It is not hard to see that the time taken by this procedure is $O(n)$ as w is reassigned in Step 6 at most n times, and w' never bypasses w as they move in the counterclockwise direction.

An important property of a maximum independent set of windows $M = \{w_1, w_2, \ldots, w_k\}$ constructed in this way is the following, whose proof is omitted.

Lemma 2 *Let $M = \{w_1, w_2, \ldots, w_k\}$ be a maximum independent set constructed in the greedy algorithm. Then for any $w \in \mathcal{W} - M$ there is a window $w' \in M$ such that $\mathcal{X}(w)$ lies in the open interval defined by $(\mathcal{X}(w'), \mathcal{Y}(w'))$.*

Let $w \in \mathcal{W} - M$. Then w intersects some $w_i \in M, 1 \leq i \leq k = |M|$. By Corollary 1 there are three possibilities depending on whether w intersects w_{i-1} or w_{i+1} or neither (see Figure 1(a) through (c)). Note that $w_0 = w_k$, and $w_{k+1} = w_1$. The case in Figure 1(d) cannot happen (Lemma 2). In this figure, the boundary of P is represented by a circle.

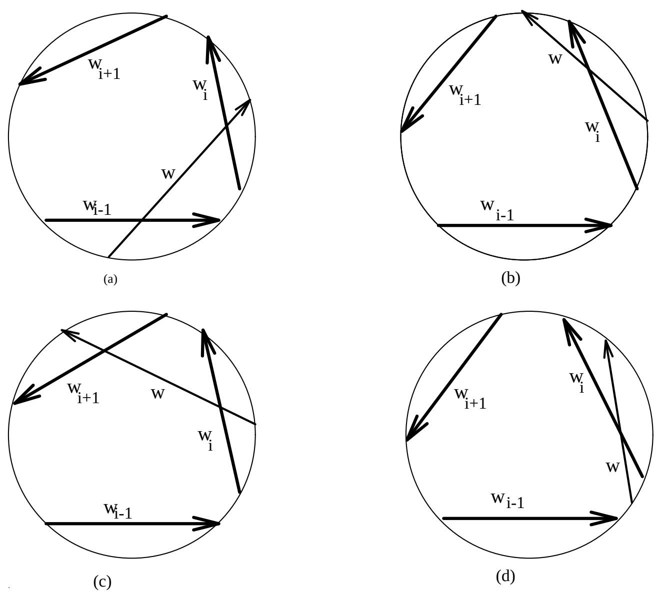

Figure 1: Possibilities of intersections of a window w with other windows in a maximum independent set.

Now, we partition $\mathcal{W} - M$ into k groups as follows.

$$W_i = \{w \in \mathcal{W} - M \mid \mathcal{X}(w) \in P(w_i)\},\ 1 \leq i \leq k.$$

Figure 2 provides an example of this grouping. In this figure, $a \in W_k, b, c \in W_1$ and $d \in W_2$. Similarly, all four windows originating from within $P(w_i)$ belong to W_i.
Finally, we define a set $S = \{p_1, p_2, \ldots, p_k\}$ of k *special points*, where $p_i = \mathcal{Y}(w_i)$, for $w_i \in M, 1 \leq i \leq k$. The intention is that these points will be the vertices of the visibility path to be constructed (see Figure 2 for example). The set S thus constructed will be referred to as the set of special points *associated with* the maximum independent set of windows M. Note that, as implied by Corollary 2, the length of any optimal visibility path π is at least $k - 1$.

Theorem 2 *Let π be a polygonal path that visits all the special points computed as described above. Then π is a visibility path.*

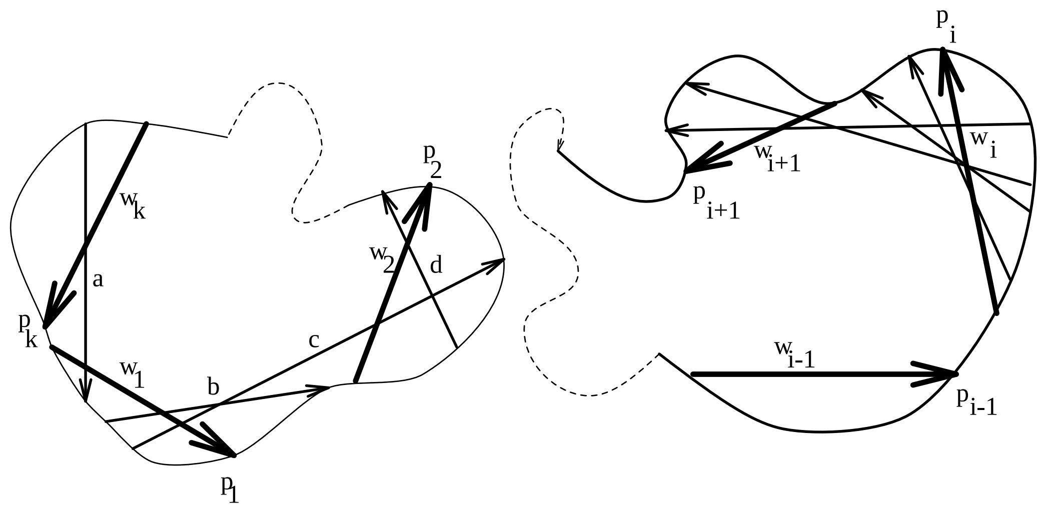

Figure 2: An example of a set of special points in a simple polygon with respect to a maximum independent set of windows (boldlines).

Proof. By Theorem 1, it suffices to show that for any window $w \in \mathcal{W} - M$ there is a special point $p \in S$ such that $p \in P(w)$. Let $w \in W_i$, where $W_i, 1 \leq i \leq k$, is a window group as explained in the construction. By construction, $\mathcal{X}(w) \in P(w_i)$. This implies that p_i is to the right of w, i.e., $p_i \in P(w)$. Note that $P(w)$ of each window $w \in \mathcal{W} - M$ contains exactly one special point.

4 Construction of an approximate minimum-link visibility path

Given a simple polygon P, the following algorithm computes an approximate minimum-link visibility path π for P.

1. Compute the set of nonredundant windows $\mathcal{W}$ of P.

2. Sort the endpoints in $\mathcal{W}$ in counterclockwise order.

3. Compute a maximum independent set of windows $M = \{w_1, w_2, \ldots, w_k\}$ and its associated set of special points $S = \{p_1, p_2, \ldots, p_k\}$ as described in Section 3. It is important to use such a construction so as to ensure that Lemma 2 holds.

4. Find $\pi_{i,j}$, a polygonal path of minimum link-length between each pair of windows $w_i, w_j \in M$.

5. Let G be the complete graph whose node set is M; to each edge of G assign a (positive integer) cost equal to the link distance between the corresponding windows.

6. Construct a minimum cost spanning tree T for the graph G. Convert T into an Eulerian graph G' by doubling its edges. Starting from a leaf node of T construct a tour t that visits all the nodes in M and bypasses all previously visited nodes. This is exactly the well-known approximation algorithm for the geometric traveling salesman problem (GTSP). Convert t into a Hamiltonian path $p = v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ by removing an edge in t (of maximum cost, say).

7. Let $w_{i_1}, w_{i_2}, \ldots, w_{i_k}$ be the corresponding sequence of windows. Rename the windows in this sequence as $w_1, w_2, \ldots, w_k$. Let $\Pi = \pi_{1,2}, \pi_{2,3}, \ldots \pi_{k-1,k}$. Π is the sequence of polygonal paths corresponding to the sequence of edges in p.

8. Convert Π into a connected polygonal path π as follows. For each window $w_i, 2 \leq i \leq k-1$, let the two polygonal paths $\pi_{i-1,i}$ and $\pi_{i,i+1}$ intersect w_i at points u_i and v_i respectively. Insert the two line segments $\overline{u_i p_i}$, and $\overline{p_i v_i}$ into the sequence Π. Also, add the two line segments $\overline{p_1 u_1}$ and $\overline{u_k p_k}$ to the beginning and end of the sequence, where u_1 and u_k are the intersections of w_1 and w_k with $\pi_{1,2}$ and $\pi_{k-1,k}$ respect Let π be the resulting polygonal path

$$\overline{p_1 u_1}, \pi_{1,2}, \overline{u_2 p_2}, \overline{p_2 v_2}, \pi_{2,3}, \overline{u_3 p_3}, \overline{p_3 v_3} \ldots \pi_{k-2,k-1}, \overline{u_{k-1} p_{k-1}}, \overline{p_{k-1} v_{k-1}}, \pi_{k-1,k}, \overline{u_k, p_k}.$$

π is the desired path.

Lemma 3 *Let π be constructed as described in the algorithm above. Then $|\pi| < 4|\pi'|$, where π' is any optimal path.*

Proof. Using an analysis similar to that of the well-known approximation algorithm for the GTSP, and noting that link-distances obey the triangle inequality, it is easy to see that

$$|\Pi| \ = \sum_{\pi_{i,j} \in \ \Pi} |\pi_{i,j}| \ < 2|\pi'|.$$

By Corollary 2, $|\pi'| \geq k-1$, and since exactly $2k-2$ line segments have been added to the sequence of polygonal paths, we have

$$|\pi| \ = \ |\Pi| + 2k - 2 < 2|\pi'| + 2k - 2 \leq 4|\pi'|.$$

The time complexity of the above algorithm is computed as follows. Step 1 takes $O(n)$ time[4]. Step 2 takes $O(n \log n)$ time. Step 3 takes $O(n)$ time[9]. Step 4 takes $O(kn)$[5, 6, 12], where $k = |M|$. By Corollary 2, $k < m$, where m is the link-length of an optimal path. Hence, the time taken by this step is $O(mn)$. The cost of building the complete graph G in Step 5 is $O(k^2)$. The total time required by Step 6 is dominated by that of finding a minimum cost spanning tree, i.e., $O(k^2) = O(m^2)$. Finally, the time taken by Steps 7 and 8 is $O(n)$. Hence, the overall time taken by the algorithm is $\max\{m^2, mn\} = O(n^2)$.

In[7], it is shown that Christofides' heuristic can be modified so that it applies to paths as well. It is not hard to see that modifying Step 6 accordingly results in a performance ratio of 3.5. In this case, the running time becomes $\max\{m^3, mn\} = O(n^3)$. To obtain a tour instead of a path, we need not remove any edges of the tour t constructed in Step 6 of the algorithm. This proves the following theorem.

Theorem 3 *Let P be a simple polygon, in which no three vertices are colinear. It is always possible to construct a watchman path (tour) whose link-length is no more than 4 times that of an optimal watchman path (tour) in time $\max\{m^2, mn\} = O(n^2)$, where m is the length of an optimal path (tour). In time $\max\{m^3, mn\} = O(n^3)$, the performance ratio can be improved to 3.5.*

5 Acknowledgement

The first author would like to thank The British Council for supporting this research during his summer visit to the U.K. He would especially like to thank the School of Mathematics and Statistics at the University of Birmingham for their hospitality. Thanks to Joseph Mitchell for bringing to our attention a gap in the proof of our earlier approximation algorithm described in [1] and for his comments on the earlier draft that help improve the readability of the paper.

References

[1] M. H. Alsuwaiyel and D. T. Lee, "Minimal Link Visibility Paths inside a Simple Polygon," *Computational Geometry: Theory and Applications*, 3, 1, (1993) 1-26.

[2] E. M. Arkin, S. P. Fekete, J. S. B. Mitchell, and C. D. Piatko, "Optimal Covering Tour Problems," Manuscript, 1994.

[3] G. Das, and G. Narasimhan, "Optimal Linear-Time Algorithm for the Shortest Illuminating Line Segment in a Polygon," *Proc. 10th Annual Symp. on Computational Geometry,* Stony Brook, NY, June, 1994, 259-266.

[4] D. Das, P. Heffernan and G. Narasimhan, "LR-visibility in Polygons," *Proc. 5th Canadian Conf. on Computational Geometry,* 1993, 303-308.

[5] S. K. Ghosh, "Computing the visibility polygon from a convex set and related problems," *J. Algorithms*, 12 (1991), 75-95.

[6] J. Hershberger and J. Snoeyink, "Computing minimum length paths of a given homotopy class," *Computational Geometry: Theory and Applications*, 4, 2, (June 1994), 63-97.

[7] J. A. Hoogeveen, "Analysis of Christofides' heuristic: Some paths are more difficult than cycles", *Operations Research Letters*, Vol 10 No 5, 1991, 291-295.

[8] Y. Ke, "Detecting the weak visibility of a simple polygon and related problems," Technical report, Johns Hopkins University, 1987.

[9] D. T. Lee, M. Sarrafzadeh and Y.F. Wu, "Minimum Cut for Circular-arc Graphs," *SIAM J. Computing,* 19,6 Dec. 1990, 1041-1050.

[10] S. Masuda and K. Nakajima, "An Optimal Algorithm for Finding a Maximum Independent Set of a Circular-Arc Graph," *SIAM J. Computing,* 17,1 Feb. 1988, 41-52.

[11] J. Mitchell, *personal communication.*

[12] S. Suri, "A linear time algorithm for minimum link-paths inside a simple polygon," *Computer Vision, Graphics and Image Processing* 35, 1986, 99-110.