Shortest Watchman Tours in Weak Visibility Polygons

P.Pradeep Kumar*and C.E.Veni Madhavan†
Department of Computer Science and Automation
Indian Institute of Science, Bangalore, India

June 17, 1993

Abstract

An $O(n^4 loglogn)$ algorithm for shortest watchman tour (*SWT*) problem for simple polygons, given a starting point on the boundary of the polygon is proposed in Chin and Ntafos [CN1]. The problem of finding the *SWT* in general polygons when the starting point is not specified is open. We develop an $O(n^{10})$ algorithm for the *SWT* problem in weak visibility polygons, with no assumption on the starting point.

1 Introduction

The watchman tour problem was introduced by Chin and Ntafos [CN2]. Earlier, many researchers considered the problem of positioning stationary watchmen in a gallery so that every point in the gallery can be seen by atleast one watchman. The problem of finding the minimum number of watchmen needed is equivalent to that of covering the polygon with minimum number of star-shaped components.

Chin and Ntafos [CN2] considered the problem of finding a *watchman tour*, which is a tour within a polygon with the property that every point in the polygon is visible from atleast one point along the tour . They present $O(n)$ and $O(nlogn)$ algorithms to find a shortest watchman route in a simple rectilinear monotone and simple rectilinear polygon, respectively. Here n is the number of vertices in the polygon. Chin and Ntafos [CN1] developed an $O(n^4 loglogn)$ algorithm to find a shortest watchman route in a simple polygon *given a starting point* on its boundary. An interesting incremental algorithm for the *SWT* problem gas recently been proposed [HIT].

We present a polynomial time algorithm for finding the shortest watchman tour, with no assumption on the starting point , for a class of polygons called *weak visibility polygons* .

Consider a simple polygon P. Two points in P are said to be *visible* if the line segment joining them lies totally inside P. A point p is said to be *weakly visible* from an edge st if there is a point z in the interior of st such that p and z are visible. If every point in P is weakly visible from st then P is called a *weak visibility polygon*. If a polygon P is weakly visible from an edge, then that edge is called a *weak visibility edge*. Pal [SP90] presents characterizations of weak visibility polygons, and based on them, an algorithm for recognizing weak visibility polygons with running time proportional to the size of the

*presently at CS dept. Ohio State Univ., Columbus, USA
†presently visiting DIMACS Center, Rutgers University, USA, cevm@dimacs.rutgers.edu

visibility graph of the input polygon. Here, visibility graph is the graph with the vertices of the polygon as the vertex set, and two vertices are adjacent if they are visible from each other in the polygon.

Assume that the simple polygon P is given as a clockwise sequence of vertices $v_1, v_2, ... v_n$ with their x and y coordinates. $\overline{v_i v_{i+1}}$ are the edges. The direction of edge $\overline{v_i v_{i+1}}$ is assumed to be from v_i to v_{i+1}. v_i is said to be the predecessor of v_{i+1}, denoted by $pred(v_{i+1})$ and v_{i+1} is said to be the successor of v_i denoted by $succ(v_i)$. It is assumed that no three consecutive vertices are collinear. Denote the boundary of P by $bd(P)$, a weak visibility edge by WVE and the shortest watchman tour by SWT. Assume that a WVE of P is given and that P is aligned with the WVE along the x-axis.

2 Properties of SWT and related results.

In what follows we develop through a sequence of definitions and propositions, the properties satisfied by candidate tours which include the SWT. These propositions characterize the corners or turning points and the segments of such tours. We suppress the proofs in this version.

Visibility polygon of a vertex v is the portion of the given polygon P, denoted $VP(v)$, such that v is visible to all points in $VP(v)$. The boundary of $VP(v)$ consists of segments through the interior of P, alternating with chains of boundary segments of P. The segments through the interior of P, if extended through the interior of the polygon, will intersect at v. $VP(v)$ has atleast one point on the WVE, since every vertex v is visible from the WVE. **Adjacent segments of a vertex** are the two segments on the $bd(P)$ meeting at the vertex. A **chord** is a segment connecting two points on $bd(P)$ through the interior of P. **Supporting Chords** of a vertex v, are the extensions of the adjacent segments of v, away from it through the interior of P until they meet the boundary of the polygon. In some cases, one or more supporting chords may not exist, in which case, they are assumed to have zero length. **Supporting Chains** of a vertex v are the bounding chains of the visibility polygon of v, starting from v_i and v_j where $v_i v$ and $v v_j$ are the adjacent segments of v. The supporting chains can contain the boundary edges of the polygon P. They necessarily include the supporting chords of v.

Proposition 2.1 *A supporting chain of any vertex v can intersect a segment $\overline{v_1 v_2}$ lying entirely within the interior of P, in atmost one point.*

Directions of edges and segments of supporting chains: To each edge on $bd(P)$, we can assign a direction as described before. Each supporting chord is assigned a direction identical to that of the adjacent segment, which when extended, gives the supporting chord. Each segment in a supporting chain C has direction identical to that of the adjacent segment from which C starts.

Inner region of a chord is the region in P to the right, while travelling along its assigned direction. Analogously,the **outer region** is the region to the left. **Inner region of a chain of segments** is the intersection of the inner regions of all the segments of the chain. **Outer region** of a chain is the region in P other than the inner region. It can be seen that the visibility polygon of a vertex is the intersection of the inner regions of supporting chains of the vertex and the boundary of the polygon. A chord s of P is said to **cover** a chord t if t lies entirely in the outer region of s. **r-supporting chord** of a vertex v above the WVE is that supporting chord of v whose direction is towards v and **l-supporting chord** is one whose direction is away from v. For a vertex v below the WVE, the r and l supporting chords are those supporting chords whose directions are away from v and towards v, respectively. Two supporting chords are said to be of the **same kind** if both are l-supporting or r-supporting. Two segments are said to be of the same kind if they are parts of supporting chords of the same kind. They are said to be of the *opposite* kind, otherwise. The set of **l-extreme chords** is the subset of l-supporting chords which are not covered by any other l-supporting chord. Similarly, the set of **r-extreme chords** is the subset of r-supporting chords which are not covered by any other r-supporting chord.

Proposition 2.2 *In a weak visibility polygon, the cover relation is asymmetric and transitive for supporting chords of the same kind . It means that if an l- supporting chord s covers an l- supporting chord t, t cannot cover s, and hence, s should be in the inner region of t. Also, if s covers t and t covers v, then s should cover v. Similar results hold for r- supporting chords.*
If the outer regions of two extreme chords of the same kind intersect, then either one of them covers the other or the two chords intersect.

Proposition 2.3 *The entire polygon P is visible from a continuous tour in P iff all the convex vertices of P are visible from the tour.*
The extreme chords cover all other chords of the supporting chains of every convex vertex.
All the convex vertices and hence the whole polygon is visible to a watchman tour visiting the visibility polygons of vertices whose supporting chains contain at least one extreme chord.

A **corner** of a tour is defined to be a point on the tour at which the tour changes direction. We label the corners of a tour in increasing order in *clockwise* direction. A corner of a tour is said to be **convex** if the angle swept by rotating the segment vv_i to the segment vv_j, about v in a counterclockwise direction is less than 180 degrees. Here v_i and $v_j (i < j)$ are the corners preceding and succeeding v in the tour. It is said to be **reflex**, otherwise. A tour is said to **reflect** at a point p on a supporting chord s if p is on the chord and both corners of the tour adjacent to p are in the **outer region** of s. The segments of the l-extreme and r-extreme chords on which a SWT can reflect are termed **l-reflecting**, and **r-reflecting segments**, respectively.

Proposition 2.4 *Each convex corner of the SWT is a point of reflection on a supporting chain or a point on the boundary of the polygon, and any reflex corner of the tour is a point on the boundary of the polygon.*

Proposition 2.5 *A SWT reflecting on a supporting chord should lie entirely in the outer region of the chord. It cannot touch or intersect the chord at any point other than the point of reflection. (The SWT can graze along a supporting chord).*
A SWT can reflect only on extreme chords.
The SWT lies entirely in the intersection R of outer regions of all extreme chords on which it reflects and the reflecting segments are on the boundary of this region.

An **essential chord** is an extreme chord whose outer region contains portions of each of the visibility polygons having one or more extreme chords on their boundaries. Call the edges on the boundary of R containing the reflecting segments as **essential segments**.

Proposition 2.6 *All points of reflection of the SWT are on essential chords. All essential chords of the same kind, as an essential chord s on which the SWT reflects, should intersect s.*

Proposition 2.7 *The l-essential and r-essential segments(containing l-reflecting, and r-reflecting segments, respectively) form two distinct convex chains on the boundary of R.*
The corners of the SWT which are points of reflection on l-reflecting and r-reflecting segments form two continuous convex chains.

3 The Algorithm

3.1 Finding essential segments

The directions of the supporting chords can be assigned by a clockwise scan around the boundary of the polygon. The extreme chords can be determined by checking pairs of supporting chords and eliminating one of them if it is covered by the other. Essential chords can be found by checking for each extreme chord whether its outer region contains portions of visibility polygons corresponding to all other extreme chords. This can be done by checking whether the extreme chord is in the inner region of or intersects both the supporting chains of the visibility polygons corresponding to the extreme chords. If so, then it means that the outer region of the extreme chord contains some portion of the visibility polygon and that the extreme chord is essential.

3.2 Finding Reflecting Segments

Let L and R be the sets of l-essential and r-essential chords. We traverse the SWT in an anticlockwise direction. Let f_l, e_l denote the first and last l-extreme chords on which the SWT reflects, and let f_r and e_r be similarly the first and last r-extreme chords on which the SWT reflects.

3.2.1 Selecting f_l and e_l

Due to intersections with other supporting chords, the chords in L are divided into $O(n^2)$ segments(each supporting chain can intersect a chord in at most one point, by Proposition 2.1). It is required to select the f_l and e_l from these.

Consider the intersections among the l-extreme chords alone. Each l-extreme chord can intersect all the others. We traverse along these chords *in* their assigned direction.The segments of all the chords until the first intersection are candidates for f_l. Subsequently, at each intersection, all except one of the intersecting extreme chords can be ignored (i.e., the rest of the segments in those chords cannot be candidates for f_l), since a tour reflecting first on any segment on these chords will fail to visit the visibility polygon corresponding to atleast one of the extreme chords intersecting at the point. Thus, there can be a maximum of $2m + 1$ segments remaining where m is the number of chords in L, where $m \leq n$. Due to intersections with other supporting chords, each of these segments may be split into $O(n)$ segments (each supporting chain can intersect the segment in at most one point). Each of these is a candidate for f_l and hence, there are $O(n^2)$ candidates for f_l where n is the number of vertices of the polygon .

Similarly, the first segments while travelling along the l-extreme chords *against* their assigned direction are candidates for e_l. At each intersection, the subsequent segments of all except one l-extreme chords intersecting at that point can be ignored since, if the SWT reflects last on one of these segments, the visibility polygons corresponding to the other l-extreme chords at the intersection cannot be visited. There are $O(n^2)$ candidates for e_l by an argument similar to that for l-extreme chords.

The case of f_r and e_r is similar.

3.3 Finding Reflecting Chains

The SWT reflects either on r-essential chords alone, or on l-essential chords alone, or on both r and l essential chords, depending on the input polygon. Let F_l, F_r and E_l, E_r be the extreme chords containing

f_l, f_r and e_l, e_r respectively. If the set $\{f_l, e_l, f_r, e_r\}$ corresponds to a shortest watchman tour, it should be the case that the outer regions of F_l, E_l, F_r and E_r intersect in a non-empty region, since the SWT lies in the outer region of all the extreme chords on which it reflects. Since the outer regions of F_l and E_l intersect, F_l should intersect E_l since none of them covers the other,by Proposition 2.3. Similar is the case with F_r andE_r. The SWT has to reflect on or intersect all extreme chords l intersecting F_l and E_l such that f_l and e_l are in the outer region of l, and l passes through the intersection of the outer regions of F_l and E_l, since otherwise, the watchman tour cannot visit the visibility polygon corresponding to l. Call these chords the *candidate reflecting chords* for (f_l, e_l). Similar chords in the case of F_r and E_r also have to be intersected or reflected upon by the watchman tour. Define the included angle between x and y to be the angle formed by rotating a line segment from x to y in an anti-clockwise direction denoted Includedangle(x, y). Let θ be the included angle between F_l and E_l. Let $\mathbf{C} = < l_1, l_2, l_3, \ldots >$ be the ordered set of reflecting chords, for (f_l, e_l) such that $x_1, x_2, \ldots, x_n$, their corresponding intersections with F_l are in the order in which they appear while travelling along F_l in its assigned direction. Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be the included angles made by the candidate reflecting chords $l_1, l_2, \ldots, l_n$ respectively with F_l .

Proposition 3.1 *(a) The SWT should reflect on* $\mathbf{R}=< y_1, y_2, \ldots, y_k >$ *the maximum sized subset of* $< l_1, l_2, \ldots, l_n >$ *such that:*

1. $< y_1, y_2, \ldots, y_k >$ *appears in the same relative order in* $\mathbf{R}$ *as in* $\mathbf{C}$,
2. *Includedangle*$(F_l, y_l) < \ldots$ *Includedangle*$(F_l, y_k) <$ *Includedangle*(F_l, E_l).
3. *There does not exist a chord* $\mathrm{y_m} \in \mathbf{C}$ *and* $\mathrm{y_m} \notin \mathbf{R}$ *such that* $y_n < y_m$ *and Includedangle*$(F_l, y_m) \leq$ *Includedangle*(F_l, y_n) *for any* $y_n \in \mathbf{R}$.

In other words, for any $y_j \in \mathbf{R}$*, there does not exist* $y_k \in \mathbf{R}$, $\mathrm{k} < \mathrm{j}$ *and Includedangle*$(F_l, y_k) >$ *Includedangle*(F_l, y_j).

(b) The set $\mathbf{R}$ *is unique and can be constructed by moving down* F_l *along its direction and, when an intersection* x_i *with an* $l_i \in \mathbf{C}$ *is encountered, removing all candidate reflecting chords* l_k *intersecting* F_l *before* x_i *such that Includedangle*$(F_l, y_k) >$ *Includedangle*(F_l, y_i).

(c) All chords in $\mathbf{C}$*, but not in* $\mathbf{R}$ *are intersected by any path reflecting on the chords of* $\mathbf{R}$.

3.4 Checking Validity of Reflecting Chains

A set of reflecting segments is valid if there exists a tour reflecting on these segments and visiting the visibility polygons corresponding to all the extreme chords.

Proposition 3.2 *A set* S *of reflecting segments is* valid
only if each of the reflecting segment is in the outer region of the extreme chord corresponding to each of the other reflecting segments, and, if so,
if and only if for each vertex v such that atleast one of its supporting chords is an extreme segment containing no reflecting segment, there exist atleast one of the reflecting segments of s *to the inner regions of each of the supporting chains of* v .

3.5 Finding the shortest length tour reflecting on prescribed segments

The reflecting chain(s), along with the boundary of the polygon form a polygon which consists of a chain of one or more simple polygons, with each simple polygon connected to the next by a single point. The reflecting segments are on the boundary of the terminal polygons. If the chain has more than one simple polygon, the *SWT* has to pass through the points connecting the polygons , and it is sufficient to find the minimum length tour reflecting on the prescribed segments of the terminal polygons, passing through the interconnecting points to the next polygon in the chain, and finally connect the tours of the terminal polygons by the shortest tour through the interconnecting points of the chain. We use an O(nloglogn) algorithm of [CN2] which first triangulates the polygon, and considering the reflecting segments as 'mirrors', 'unfolds' the polygon and finds the shortest path connecting one of the reflecting segments to its 'reflected image'.

4 Proof of Correctness

Proposition 4.1 *The algorithm - shortest watchman tour computes in polynomial time, the SWT in a given weak visibility polygon .*

Proof

The reflecting segments form one or two continuous, convex chains according to Proposition 2.7. The method in Section 3.2 selects all the possible first and last segments for these chains. Proposition 3.1 proves the method for finding the unique convex chains between the selected corner segments of the same type. By Proposition 2.3, it is sufficient for the watchman tour to visit the visibility polygons corresponding to the extreme chords. Hence, by Proposition 3.2 the algorithm computes all possible watchman tours for the polygon, and hence, the Shortest Watchman Tour.

Let n be the number of convex vertices. Finding extreme chords can be done in $O(n^2)$ time (each pair of supporting chords may have to be considered to check whether one of them is covered by the other and there are $O(n^2)$ of them). Since there are $O(n^2)$ candidates for f_l, e_l, f_r and e_r, there are $O(n^4)$ possible (f_l, e_l) and (f_r, e_r) pairs, and $O(n^8)$ quadruples (f_l, e_l, f_r, e_r). Checking for valid chains take $O(n^2)$ time and finding the minimum length touching the prescribed reflecting segments take O(nloglogn) time. Thus, the overall complexity of the algorithm is $O(n^{10})$.

References

[1] [SP] Sudebkumar Prasant Pal, ' Weak Visibility and related problems on simple polygons ', Ph.D. thesis, 1990, Indian Institute of Science, Bangalore, India.

[2] [CN1] W.P.Chin and S.Ntafos, ' Shortest Watchman Routes in simple polygons ', *Tech. Report.*

[3] [CN2] W.P.Chin and S.Ntafos, ' Optimum Watchman routes ', *Info. Proc. Lett. 28* , 1988, 39-44.

[4] [HIT] Hirata, Inagaki, and Tan , ' An incremental algorithm for constructing shortest watchman tours ' *Int. Symp. on Algorithms ISA 91 , LNCS 557* , 1991.