Discrete Comput Geom 6:9-31 (1991)

Discrete & Computational
Geometry

Shortest Watchman Routes in Simple Polygons*

Wei-Pang Chin[1] and Simeon Ntafos[2]

[1] AT&T Bell Laboratories, Columbus, Ohio, USA

[2] Computer Science Program, University of Texas at Dallas, Richardson, TX 75083-0688, USA

Abstract. In this paper we present an $O(n^4 \log \log n)$ algorithm to find a shortest watchman route in a simple polygon through a point, s, in its boundary. A watchman route is a route such that each point in the interior of the polygon is visible from at least one point along the route.

1. Introduction

A number of researchers have considered the problem of stationing watchmen in a gallery so that every point in the gallery can be seen by at least one watchman (art-gallery problem). Most of the results on gallery watchmen and related problems can be found in [O]. The watchman-route problem was introduced in [CN]. The problem is to find a route in a polygon with the property that each point in the polygon (interior and boundary) is visible from at least one point along the route. Two points in a polygon are visible to each other if no point on the straight line segment connecting them is exterior to the polygon. The goal is to minimize the length of the route.

Finding a shortest watchman route is known to be NP-hard for polygons with holes and for simple polyhedra [CN]. For simple rectilinear polygons, an $O(n)$ algorithm that constructs a shortest watchman route is given in [CN] (if we allow $O(n \log \log n)$ time for triangulation [TV]). A shortest watchman route follows the boundary of the polygon as it moves in and out of areas that allow the watchman to see around corners. The extent to which a shortest route needs to come into a certain area is captured by the concept of essential lines. These are straight line segments inside the rectilinear polygon such that any watchman route must visit

* S. Ntafos was supported in part by a grant from Texas Instruments, Inc.

them and any route that visits them is a watchman route. After the essential lines are identified, the portions of the polygon that lie outside them can be removed (since a shortest watchman route never needs to enter them). The resulting polygon is triangulated and "unrolled" using the essential lines as mirrors so that the problem of finding a shortest watchman route becomes that of finding a shortest path from a point to an image of itself inside a simple polygon. Existing $O(n)$ algorithms [GH*] are used to solve the shortest-path problem and a shortest watchman route is obtained by "folding" back the shortest path. The approach is illustrated in Fig. 1. Fig. 1(a) shows a simple rectilinear polygon and the set of essential lines for it. Fig. 1(b) shows the reduced polygon when areas that a shortest watchman route does not need to enter are removed. In Fig. 1(c) we have the result of the "roll-out" process (the idea is to straighten a ray that reflects on a mirror into a ray that passes through the mirror, where the ray is the watchman route and the mirrors are the essential lines; to achieve this, triangles d, c, and the degenerate triangle e_3 reflect on e_2, e_3 reflects on itself, then e_3 and the triangles c, b, a, reflect on the degenerate essential line e_4 and finally triangles a, b, c, d reflect on e_1). Fig. 1(c) also shows a shortest path from a point, s, that must be on the shortest watchman route to its image, s'. Finally, in Fig. 1(d) the shortest watchman route (obtained by folding the shortest path at its intersections with the essential lines) is shown.

In this paper we present a polynomial-time algorithm for the watchman-route problem in simple polygons. We assume that a "starting" point, s, on the boundary

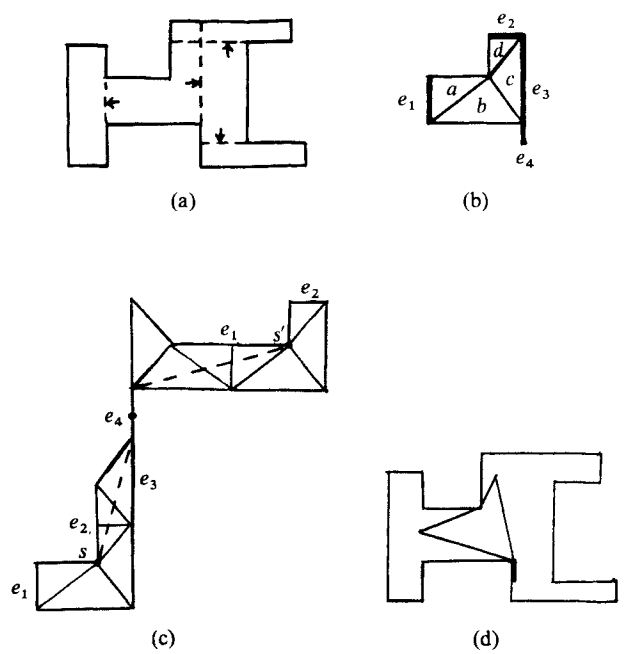

Fig. 1. Finding shortest watchman routes in simple rectilinear polygons.

of the polygon is specified for the route, i.e., the route starts at s and has to return to s. Usually, s can be selected so that it lies on a shortest watchman route. Also, in most applications s is already specified and the route has to be designed through it. The problem without a fixed starting point remains open. In the next section we give an overview of the algorithm. In Section 3 we prove that the shortest watchman route in simple polygons is unique. Finally, in Section 4, we give an algorithm to construct the shortest watchman route and analyze its complexity.

2. Overview of the Algorithm

Consider an n-sided simple polygon P with a point s on its boundary. The polygon can be described by a sequence of vertices $v_0, v_1, \ldots, v_{n-1}$ or a sequence of edges $E_0, E_1, \ldots, E_{n-1}$ indexed in the order that they appear in a clockwise scan of the boundary of P starting at s (E_i connects v_i with v_{i+1} and $v_0 = v_n = s$). We assign an orientation to the edges of the polygon as implied by a clockwise scan, i.e., edge E_i is oriented from v_i to v_{i+1}.

Definition 1. Let v_i be a reflex vertex in P. The *cuts* C_{i-1} and C_i in P are the longest straight line segments that contain E_{i-1} and E_i, respectively, and do not intersect the exterior of P.

For the next three definitions, the orientation we assign to a cut C_i is the same as the orientation of the edge E_i associated with it (the orientations are redefined later). Each cut is described as an ordered pair $C_i = \langle s_i, t_i \rangle$, where s_i and t_i are the starting and ending points of C_i, respectively. Cut C_i separates (cuts) the boundary of P into two disjoint chains, one from s_i to t_i and one from t_i to s_i (the directions are those implied by a clockwise scan). We denote them as LC_i, RC_i, respectively.

Definition 2. Cut C_i is a *visibility cut* in P and s is in LC_i.

Without loss of generality, we assume that $C_i \neq C_j$ if $i \neq j$ for all i, j. The importance of visibility cuts stems from the fact that a watchman must visit at least one point on C_i in order to see the whole edge E_i. There are at most $O(n)$ visibility cuts inside P. Some of them are not important in determining the shape of a shortest watchman route and can be disregarded.

Definition 3. Visibility cut C_i *dominates* visibility cut C_j if RC_j contains RC_i.

The notion of domination is also used in [S]. Clearly, if C_i dominates C_j, any route that visits C_i will automatically visit C_j, i.e., C_j can be disregarded. The cuts that are not dominated may be important in determining the shape of the shortest watchman route.

Definition 4. A visibility cut C_i is an *essential cut* if, and only if, there is no C_j, $i \neq j$, such that RC_j is properly contained in RC_i.

The set of essential cuts can be identified in $O(n)$ by applying the above definition in a scan of the boundary. Once we have the set of essential cuts for P, it is convenient, for the discussion in the remainder of this paper, to index them in the order in which endpoints first appear in a clockwise scan of the boundary. We still describe each essential cut as an ordered pair $C_i = \langle s_i, t_i \rangle$ but now s_i is the endpoint that is visited first in a clockwise scan of the boundary. The orientation of the cut is taken to be from s_i to t_i, e.g., when we say that something is to the left (right) of C_i, we do so with reference to somebody moving along C_i from s_i to t_i. Also, note that the index of the cut and the index of the edge that gave rise to it, will not be the same in general (a correspondence can be easily kept).

We then partition the set of essential cuts into a number of corners.

Definition 5. A *corner* is an ordered set of essential cuts $C_i, C_{i+1}, \ldots, C_j$ such that:

(1) each C_k intersects C_{k-1}, $i < k \leq j$,
(2) C_i does not intersect C_{i-1}, and
(3) C_j does not intersect C_{j+1}.

Definition 6. An *e-segment* is any line segment along an essential cut C_i that starts at s_i or at an intersection p_{ij} (of C_i with some C_j), ends at t_i or at an intersection p_{ik} (of C_i with some C_k), and does not contain any intersections with other essential cuts in its interior.

The set of essential cuts in P can be easily partitioned into a number of disjoint corners in $O(n)$ time. In a corner with k essential cuts, there are at most $k - 1$ intersections along each cut and thus there are at most $O(k^2)$ e-segments in a corner consisting of k essential cuts. Each intersection corresponds to a switch in dominance between the intersecting essential cuts. For example, consider C_i and C_j in Fig. 2. On one side of the intersection p_{ij}, cut C_j dominates C_i (in the sense that any route that visits C_j will also visit C_i) while the opposite is true on the other side of the intersection. In Fig. 2 there are four e-segments along C_k. The problem faced by our watchman is to select $O(k)$ out of the $O(k^2)$ possible e-segments so that

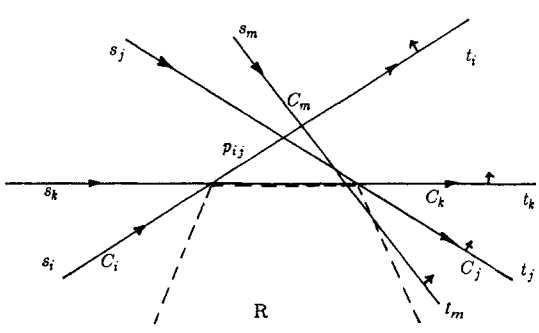

Fig. 2. Types of contacts between a watchman route and essential cuts.

visiting them in some appropriate order will mean that each point in the corner will be seen from the route and the distance traveled will be minimum.

It is helpful to consider a physical analog for the process we use to find a shortest watchman route. Consider a string that is threaded through a set of small rings that are free to slide along the essential cuts (we have at least one ring per cut) and can move through intersections of essential cuts if a certain amount of force is applied. The two ends of the string come together at the point s. Suppose now that we pull the string at s so that it is taut and the threading is such that the string forms a convex chain in each corner. Then the string traces a watchman route since it makes contact with all the essential cuts. This watchman route will be the shortest among all the watchman routes that have bends at the same set of e-segments. Suppose now that we apply additional force at s. In general, a sequence of adjustments will take place as rings move through intersections. Each adjustment will make the string inside the polygon shorter. The process will eventually terminate when no ring moves regardless of how much additional force is applied. At that stage, the string traces a shortest watchman route. The important issues here are to assure that the process is not stuck at some local optimum and to find a short adjustment sequence.

Let us consider some properties that any shortest watchman route must have.

Lemma 1. *There is a shortest watchman route that visits the set of corners in the order in which they appear in a clockwise scan of the boundary of the polygon.*

Proof. Consider any shortest watchman route in P and orient it in a clockwise fashion. Suppose that it visits the corners in an order other than the one in which they appear in a clockwise scan of the boundary. Then there must exist four corners (the point s may be one of them) that are visited as shown in Fig. 3. We note that we can easily replace this route with one that visits the corners in the specified order. The new route has exactly the same visibility properties and is no longer than the original route. □

The above lemma states that a shortest watchman route need not "cross" itself. This property also applies inside a corner, i.e., the shortest watchman route will visit a selected set of segments on the essential cuts so that it does not properly intersect itself (overlapping sections are possible). Let us now consider how a

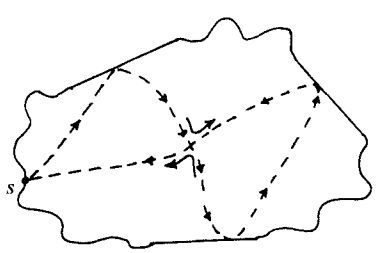

Fig. 3. A shortest watchman route does not need to "cross" itself.

shortest watchman route can come in contact with the essential cuts. We distinguish three types of contacts.

Definition 7. A watchman route make a *reflection contact* with an essential cut if the route and the cut have exactly one point in common. A *perfect reflection* occurs when the angles formed by the incoming and outgoing sections of the route with the essential cut are equal. We say that a watchman route makes a *crossing contact* with an essential cut if there are two common points between the route and the essential cut. Finally, they make a *tangential contact* if they share a line segment.

The three contact types are illustrated in Fig. 2 where the route makes reflection contacts with C_i, C_j, a tangential contact with C_k, and a crossing contact with C_m. In a reflection contact the route comes into the essential cut, makes contact at a single point, and then "reflects" on this cut, i.e., it changes direction and moves away from the essential cut. If the point of contact occurs at an internal point of the essential cut, then an optimum route must reflect perfectly on the essential cut. If the point of contact occurs at the intersection of two essential cuts, then reflection need not (and usually will not) be perfect with respect to either cut.

In a crossing contact, the route passes through the essential cut on its way into a corner and passes through the cut a second time on its way out of the corner. When a shortest watchman route makes a crossing contact on a cut C_m, it must make a reflection contact on some other cut C_j and the point of contact on C_j must be to the left of C_m (Fig. 2). A tangential contact occurs when the portion of the route between two successive reflection contacts happens to overlap with an essential cut (C_k in Fig. 2). Tangential contacts are degenerate cases of reflection contacts.

A shortest watchman route can be defined by the set of e-segments (on some subset of the essential cuts) that the route makes reflection contacts with. If we have the "best" set of e-segments, the shortest watchman route can be found using the same approach as that described in [CN] for the watchman-route problem in simple rectilinear polygons. The portions of P that are behind (as viewed from s) each of the selected essential cuts C_i are removed, the interior of the resulting polygon P' is triangulated and P' is rolled-out by treating the selected segment on each cut C_i as a mirror and reflecting corresponding portions of the polygon with respect to these mirrors. This process reduces the problem of finding a shortest route to the problem of finding a shortest path from s to an image s' of itself inside a simple polygon.

Our approach for finding a shortest watchman route is to construct an initial watchman route R^0. Then, by checking local optimality properties at e-segments where reflection contacts are made, the route goes through a sequence of adjustments so that:

(a) the route is made shorter after each adjustment and
(b) a shortest watchman route is obtained when no more adjustments can be made.

We start by specifying how the initial route R^0 is obtained. First, we select a set of *extended line segments* within each corner so that, if a watchman route (e.g., R^0)

makes reflection (or tangential) contacts with these segments, it will visit all essential cuts in the corner. An *extended line segment* is any continuous portion of an essential cut C_i, starting at s_i or some intersection p_{ij} and ending at an intersection p_{ik} or at t_i (i.e., a set of consecutive e-segments).

We select a set of extended line segments (from which R^0 will be constructed) in each corner and keep them in a queue as follows: Let C_i be the first (least index) cut in some corner. We denote C_i as the current cut and s_i as the current initial point. We repeat the following process, which we call Navigation in a Corner:

Navigation in a Corner. Walk from the current initial point along the direction of the current cut C_i until an intersection is encountered.

(1) If the next intersection is p_{ij} (the intersection of C_i and a new cut C_j) we look for intersections of C_j with some other cut C_k that we have not visited yet and such that p_{jk} is to the left of the current cut C_i. If such an intersection exists, we continue to walk along the current cut. If no such intersection exists, we place the current extended segment (from the current initial point to p_{ij}) in the queue, p_{ij} becomes the current initial point, C_j becomes the current cut, and we continue along C_j.

(2) If the next intersection is t_i (at the boundary), we add the extended line segment from the current initial point to t_i to the queue. If the current cut is the last cut in this corner, we are done; otherwise, let C_j (the next cut in this corner) be the current cut, let s_j be the current initial point, and continue along C_j.

The process of navigating a corner is illustrated in Fig. 4, where a corner with six essential cuts is shown. The heavy lines are the extended segments that will be placed in the queue by this process. The navigation process starts at s_1 and we move along C_1. At p_{13}, we determine that C_3 intersects C_2 (a cut that we have not visited as yet) at a point that lies to the left of C_1; thus, we continue moving along

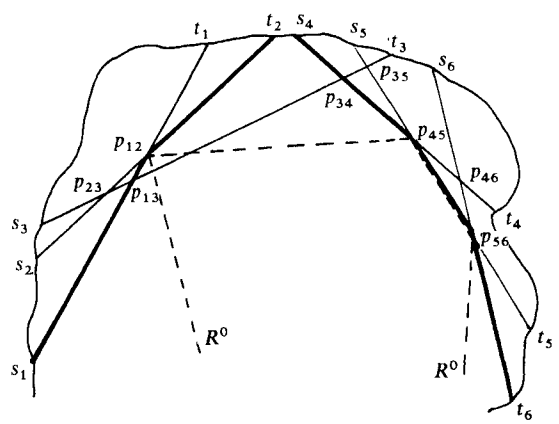

Fig. 4. Navigation in a corner.

C_1 and we reach p_{12}. There, we determine that C_2 does not intersect any cuts we have not visited and such that the intersection is to the left of C_1 (note that C_3 has been visited since we passed through p_{13}); thus, we place the extended line segment from s_1 to p_{12} in the queue, C_2 becomes the current cut and p_{12} becomes the current initial point. We now move along C_2, reach t_2, insert the extended line segment from p_{12} to t_2 in the queue and repeat the process starting at s_4 and moving along C_4.

It is clear that the navigation process visits each essential cut C_i either

(1) along a portion (line segment) of C_i in the direction from s_i to t_i, or
(2) it crosses C_i twice.

The set of extended line segments forms a convex chain in each corner (we make no left turns). A shortest watchman route that reflects on the extended line segments has to visit them in the order in which they were placed in the queue by the navigation process. If it does not, another watchman route can be easily constructed that visits the same set of extended line segments in the specified order and has less or equal length (as in Lemma 1). Note that any shortest watchman route will also consist of convex chains within each corner. Concave sections can occur only where the route comes in contact with the boundary of the polygon. If a concave section was to occur inside a corner, this section could be stretched out to obtain a shorter route.

After the navigation process is completed in all the corners, we find the initial route R^0 by constructing (and then folding back) a shortest path from s to its reflection inside the polygon obtained by unrolling P using the extended line segments as mirrors (as in [CN]). Route R^0 visits the extended line segments in the order that they were recorded by the navigation process. The set of e-segments (one per extended line segment) that R^0 reflects on is defined to be the *active segment set* A^0. In the example of Fig. 4, the active segment set is

$$A^0 = \{\langle p_{13}, p_{12}\rangle, \langle p_{12}, t_2\rangle, \langle p_{34}, p_{45}\rangle, \langle p_{45}, p_{56}\rangle, \langle p_{56}, t_6\rangle\}.$$

Lemma 2. *Route R^0 is a watchman route.*

Proof. By contradiction. If R^0 is not a watchman route, then there exists an essential cut C_j such that R^0 does not contact C_j. From the process of navigation, we know that either

(1) a portion of C_j is in the recorded extended line segment set, or
(2) there is a recorded extended line segment of some C_k that is to the left of C_j.

In case (1), R^0 must make a reflection or tangential contact with C_j because a portion of C_j is used as a mirror in unfolding P; the unfolding process forces the path from s to its image to cross all the mirrors, i.e., the route obtained by folding this path must make reflection contacts with them. In case (2), R^0 must make a reflection contact with C_k to the left of C_j which implies that it must make a crossing contact with C_j (e.g., in Fig. 4, the route reflects on C_2 at a point that is to the left of C_3, i.e., it crosses C_3). Thus, R^0 is a watchman route. □

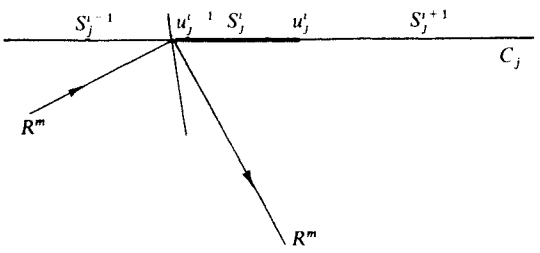

Fig. 5. Route R^m is adjustable at u_j^{i-1}.

The shortest watchman route is obtained by adjusting the current route R^m, where $m \geq 0$. Each adjustment involves a change in the current active segment set and results in a shorter watchman route. Let $S_j^1, S_j^2, \ldots, S_j^{m_j}$ be the segments on essential cut C_j and let u_j^i be the common endpoint of S_j^{i-1} and S_j^i, $1 < i \leq m_j$. Assume that S_j^i is a segment in the active segment set A^m. Consider the example shown in Fig. 5. The watchman route R^m makes a reflection contact with segment S_j^i at its left endpoint u_j^{i-1}. Since the incoming angle of R^m with respect to C_j is less than the outgoing angle of R^m with respect to C_j, the watchman route R^m can be made shorter by moving the contact point with C_j to the left of u_j^{i-1}. Similarly, if R^m made a reflection contact at u_j^i and the incoming angle was greater than the outgoing angle, then R^m could be made shorter by allowing its contact with C_j to move to the right of u_j^i.

Definition 8. A watchman route R^m is *adjustable* at u_j^{i-1} (u_j^i) from S_j^i to S_j^{i-1} (S_j^{i+1}) on some essential cut C_j if, and only if:

(a) R_m has a reflection contact with C_j at u_j^{i-1} (u_j^i),
(b) the incoming angle between R^m and C_j is smaller (larger) than the outgoing angle between R^m and C_j and
(c) the route remains a watchman route if its contact with C_j is shifted to the left of u_j^{i-1} (right of u_j^i).

An adjustment involves a change in the current active segment set. Since an adjustment occurs at the intersection of two essential cuts, one or two segments in the current active segment set will be affected. We distinguish three basic adjustment types as shown in Fig. 6. Each of them has the property that the incoming angle of R^m with C_j is smaller than the outgoing angle of R^m with C_j (for each type, there is a symmetric case where the adjustment occurs at the other endpoint of S_j^i and the incoming angle is larger than the outgoing angle). The bold segments in Fig. 6 stand for segments that belong to the current active segment set A^m. The discontinuous segments represent the segments that will replace them to form the next active segment set A^{m+1}. A possible next route, R^{m+1}, is also shown.

In Fig. 6(a) R^m makes reflection contacts with both C_j and C_k at their intersection. The adjustment involves moving the reflection contact with C_j from S_j^i to S_j^{i-1} (note that we should not adjust along C_k as that would leave C_j without a contact with the new route). The next route, R^{m+1}, will make a reflection contact

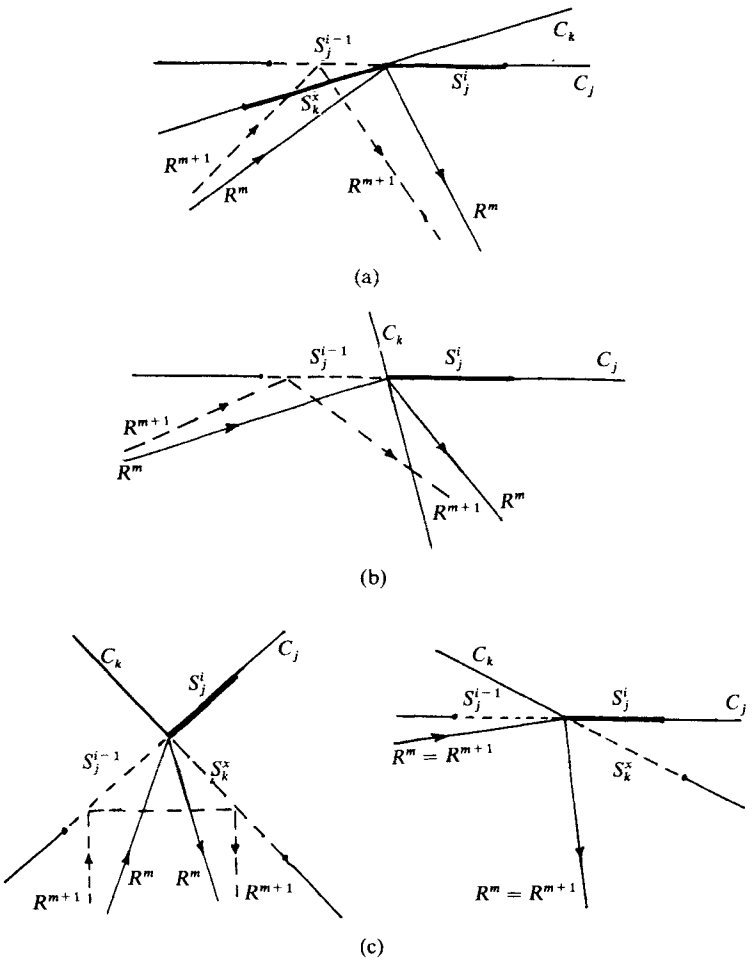

Fig. 6. The three basic adjustment *types*: (a) (− 1)-type adjustment, (b) 0-*type* adjustment, and (c) a (+1)-*type* adjustment and a (+1)-*switch*.

with C_j but a crossing contact with C_k. We call this a (-1)-*type adjustment* as $|A^{m+1}| = |A^m| - 1$ (delete S_k^x and replace S_j^i with S_j^{i-1}). We note that a (-1)-type adjustment also arises if the contact with C_k is tangential.

In Fig. 6(b), R^m makes a reflection contact with C_j and a crossing contact with C_k. Adjusting on the intersection point from S_j^i to S_j^{i-1} does not affect the crossing contact with C_k. We call this a 0-*type adjustment* since $|A^{m+1}| = |A^m|$ (replace S_j^i with S_j^{i-1}). The next route, R^{m-1}, still reflects on C_j and makes a crossing contact with C_k.

In Fig. 6(c) R^m is supposed to make a reflection contact with C_j and a crossing contact with C_k. Since the contact point is at the intersection of the two essential cuts, the intended crossing contact with C_k has degenerated into a reflection contact. To account for this, we replace S_j^i with S_j^{i-1} and insert S_k^x (at the position

following S_j^{i-1}) into A^m to obtain A^{m+1}. Depending on the angles formed by C_j, C_k, and R^m, the next route will either be shorter (reflecting at distinct points on C_j, C_k), or the same as R^m. We refer to the first case as a $(+1)$-*type adjustment*, as $|A^{m+1}| = |A^m| + 1$. If $R^{m+1} = R^m$, the operation serves only to change the active segment set to account for the degeneracy of the crossing contact with C_k and we call it a $(+1)$-*switch*. $(+1)$-switches are easy to perform (the route does not need to be reconstructed) and their total number is at most O(number of adjustments). The following lemma follows directly from the definitions of the various adjustment types.

Lemma 3. *If R^m is an adjustable watchman route, then R^{m+1} is also a watchman route.*

The algorithm proceeds by finding a point at which the current route is adjustable, updating A_m to A_{m+1} and constructing R^{m+1}. This is repeated until no further adjustments are possible. Since each adjustment results in a shorter route and the number of possible active segment sets is at most exponential (in the number of essential cuts), it follows that the process will eventually terminate. Two important questions remain:

(a) Will we end up with a shortest watchman route?
(b) How many adjustments do we need to make?

We treat these issues in the next two sections.

3. Uniqueness of the Shortest Watchman Route

In the previous section we outlined how an initial watchman route can be constructed. Let R be the nonadjustable watchman route obtained from R^0 by continuing to adjust the current route until it is not adjustable any longer. This approach raises the following question: If we start from a different initial route that makes reflection contacts with active segment set B^0, $B^0 \neq A^0$, will the resulting nonadjustable watchman route R' be different from R? In this section we show that the shortest watchman route is unique and that any other watchman route must be adjustable.

We note that the essential cuts that a watchman route has reflection contacts with, form a convex chain in each corner. The watchman route that reflects on a given set of essential cuts is obtained by unrolling the polygon using these essential cuts as mirrors. Consider the example shown in Fig. 7. It shows a watchman route that reflects on four essential cuts in a corner of a polygon and also the rolled-out version of the corner and the route. Note that the essential cuts in Fig. 7(a) become diagonals in the polygon of Fig. 7(b), they cross from one side to the other in a zigzag fashion and the route is now a path (in this case a straight line) that crosses the diagonals corresponding to the essential cuts.

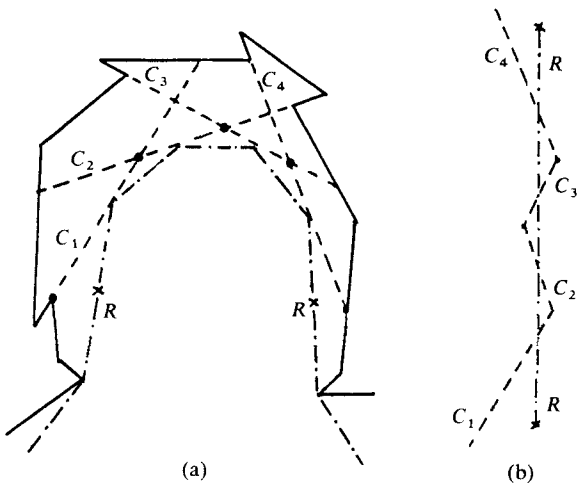

Fig. 7. A watchman route in a corner (a) and its rolled-out version (b).

Definition 9. Let R, Q be two watchman routes. The *angle of divergence*, θ_i, of the two routes at cut C_i is the angle formed by the extensions of the out-going segments of the two routes as they last visit C_i.

Lemma 4. *Perfect reflections preserve the angle of divergence between two watchman routes.*

Proof. Consider the two cuts C_i, C_j and let R, Q be watchman routes that reflect perfectly on C_i and then on C_j (see Fig. 8(a)). We have that $\theta_i = r_i - q_i = 180 - r_j - x = (180 - q_j - x) = q_j - r_j = \theta_j$. □

Lemma 5. *If two nonadjustable routes R, Q, reflect on C_i and then on C_j, then $\theta_i \leq \theta_j$.*

Proof. If the reflections are perfect, the angles of divergence are equal by Lemma 4. If the reflections are not perfect, then they must occur at vertices. We have that $\theta_i = r_i - q_i = q_{j1} - r_{j1}$, where q_{j1}, r_{j1} are the angles formed by the incoming segments of Q, R with C_j. Since the two routes are not adjustable, they are not adjustable toward each other (i.e., in the directions that would bring their contacts with a cut closer to each other). This implies that $q_{j1} \leq q_{j2}$ and $r_{j1} \geq r_{j2}$, where q_{j2}, r_{j2} are now the angles formed by the outgoing segments of Q, R with C_j. Then $\theta_i \leq q_{j2} - r_{j2} = \theta_j$.

Note that it is possible that $\theta_i < \theta_j$. In Fig. 8(b) we have that $q_{j1} < q_{j2}$ and $r_{j1} > r_{j2}$. Then the contacts of R, Q with C_j are trying to slide away from each other along C_j but cannot do so because of other essential cuts (like cuts C', C'' in Fig. 8(b)). If two routes are nonadjustable, they are not adjustable away from (as well as toward) each other. That is, we would have that $q_{j1} \geq q_{j2}$ and $r_{j1} \leq r_{j2}$ in Fig. 8(a).

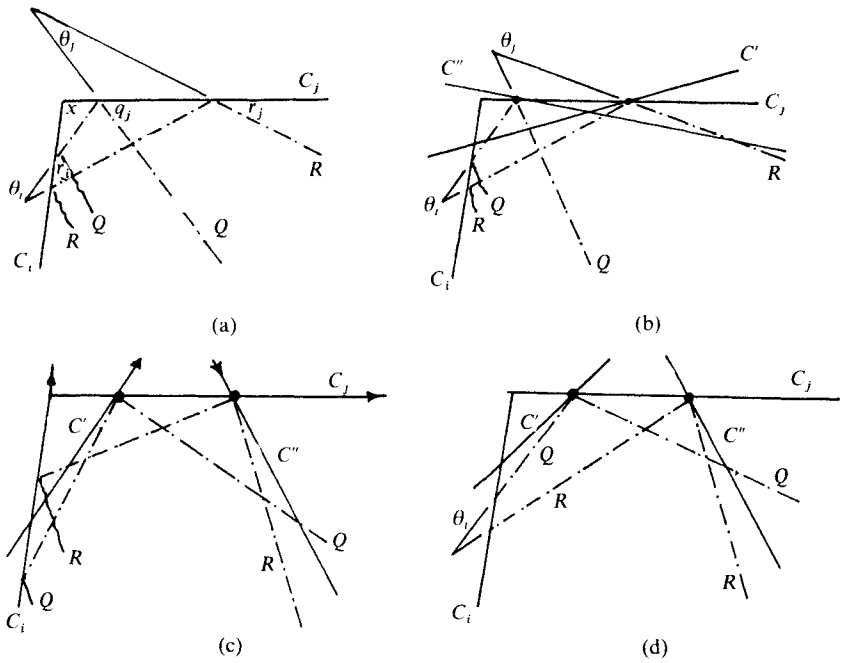

Fig. 8. Angles of divergence between two routes.

Then, if one or both inequalities hold, it would lead to the opposite conclusion, i.e., that $\theta_i > \theta_j$. This does not occur because, in order for the routes to be nonadjustable, we must have essential cuts C', C'' (as shown in Fig. 8(c)) that prevent the contacts along C_j from sliding toward each other. But then route Q must visit C'' which means that the outgoing segments of R, Q from C_j must intersect to the right of C_j (Fig. 8(c)). Similarly, route R needs to visit C' which implies that the outgoing segments of R, Q from C_i intersect to the right of C_i. There must be a first time when the outgoing segments of the two routes intersect to the right of a cut, i.e., we must have the situation shown in Fig. 8(d). But then route R does not visit cut C', which contradicts our premise that R is a watchman route. □

Definition 10. The *characteristic vector*, VR, of a watchman route, R, is a binary vector with one entry for each essential cut such that $VR_i = 1$ if route R reflects on C_i; $VR_i = 0$, if R crosses C_i. (Essential cuts are indexed in the order that they are first visited in a clockwise scan starting at s.)

Lemma 6. *Let R, Q be watchman routes and assume that $VR_i = VQ_i = 0$. If we remove C_i (without changing the e-segments along other cuts), R, Q are still watchman routes in P and they are adjustable if, and only if, they were adjustable before the removal of C_i.*

Proof. The shape of a watchman route is determined by the e-segments with which the route makes reflection contacts. The removal of a cut that is crossed by R, Q, will not disturb the routes. Adjustments take place only along essential cuts that a route reflects on. Then the only concern is that a route that was not adjustable before removing C_i will become adjustable once we disregard it (i.e., the new route will not contact C_i and will not be a watchman route in the original setting). This cannot happen in a 0-*type* adjustment since the route would cross C_i both before and after adjusting. No crossing contact is affected by the other adjustment types (note that crossing contacts that have degenerated into reflection contacts are taken care of by $(+1)$-switches in the route constructed by the algorithm). □

Lemma 6 allows us to disregard essential cuts that are crossed by both routes as far as the immediate adjustability of these routes is concerned. We are now ready to prove that there is only one nonadjustable watchman route.

Theorem 1. *There is a unique nonadjustable watchman route in P.*

Proof. The proof is by contradiction. Let us assume that R, Q are watchman routes in P and both of them are not adjustable. We start with a slightly simplified case that captures all the important issues and generalize it later. In our simplified case, we assume that the two routes R, Q meet the boundary of the polygon only at s. Also, without loss of generality (Lemma 6), we assume that there are no essential cuts that are crossed by both R and Q.

Since R, Q are distinct, they must diverge from each other at some point. The essence of the contradiction that we will develop is that once the two routes start diverging, nonadjustability implies that they will continue to diverge and then they cannot both return to the same point s. We consider a number of cases:

Case 1: $VR = VQ$. Then both routes reflect on the same set of essential cuts. Consider the unrolled version of the polygon (see Fig. 9). Then both routes are now paths from s to its image s' inside a simple polygon. Clearly, there is a unique shortest path from s to s'. Since reflections cannot reduce the angle of divergence (Lemmas 4 and 5), it follows that the two routes cannot both get to s'. In order to

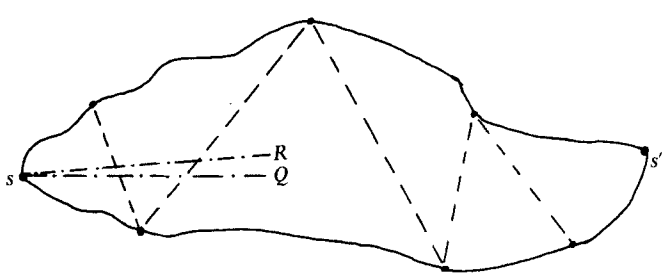

Fig. 9. Routes R, Q reflect on the same set of cuts.

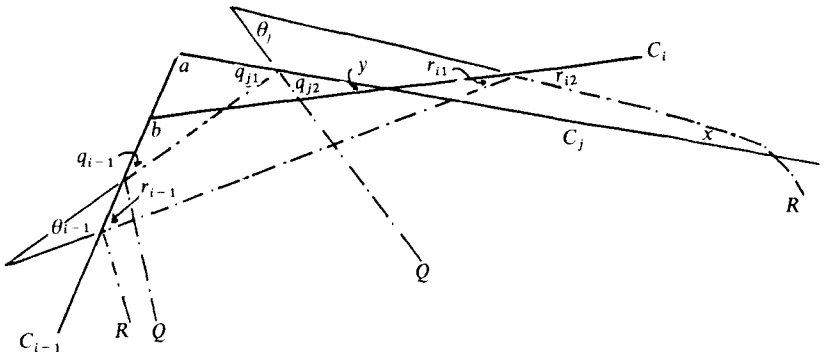

Fig. 10. Case 2.1 of Theorem 1.

do so, one or both of them must be adjustable toward the shortest path (by 0-type adjustments).

Case 2: $VR \neq VQ$. Let the first difference be at cut C_i and (without loss of generality) assume that $VR_i = 1$ (R reflects on C_i), while $VQ_i = 0$ (Q crosses C_i). Then Q must reflect on a cut C_j and its reflection contact with C_j must be to the left of C_i.

Case 2.1: $j = i + 1$ *and* $VR_j = 0$. The situation is illustrated in Fig. 10. We have that $\theta_{i-1} = r_{i-1} - q_{i-1} = 180 - r_{i1} - b - 180 + q_{j1} + a = q_{j1} - r_{i1} + (a - b)$. Since neither route is adjustable toward the other, we have that $q_{j1} \leq q_{j2}$ and $r_{i1} \geq r_{i2}$. Also, from the convexity of the essential cuts in a corner we have that $a < b$. Substituting, we have that $\theta_{i-1} < q_{j2} - r_{i2} = \theta_j + x - y - x < \theta_j$, i.e., $\theta_{i-1} < \theta_j$.

Case 2.2: $j = i + 1$ *and* $VR_j = 1$. The situation is illustrated in Fig. 11(a) and (b). The difference in the two parts of the figure is the relative position of the two routes as they come into C_{i-1}. For the situation in Fig. 11(a) we have that $\theta_{i-1} = r_{i-1} - q_{i-1} = 180 - x - z - 180 + x + y = y - z = q_{j1} - r_{j1} - z \leq q_{j2} - r_{j2} = \theta_j$. In Fig. 11(b) we have that $\theta_{i-1} = q_{i-1} - r_{i-1} = 180 - x - a - 180 + r_{i1} + a = r_{i1} - x \leq r_{i2} - x = 180 - r_{j1} - b - x = q_{1j} - r_{1j} - 2x \leq q_{2j} - r_{2j} = \theta_j$. Thus, in both cases, we have that $\theta_{i-1} < \theta_j$.

Case 2.3: $j > i + 1$. This means that there is a (long) sequence of differences in the characteristic vectors before both routes reflect on the same cut. We show that $\theta_{i-1} < \theta_j$ by extending the arguments used in Cases 2.1 and 2.2. First assume that VR has the pattern (...11..100..0...) and the corresponding pattern in VQ is (...00..011..1...). This is similar to Case 2.1 with the difference that we have sequences of reflections (1's) rather than a single reflection. Consider Fig. 10 and let the cut C_i in Fig. 10 correspond to the last reflection in the sequence in VR. Also, let the cut C_j in Fig. 10 correspond to the first reflection in the sequence in VQ. If the remaining reflections did not exist, we have, from the arguments in Case 2.1, that the angle of

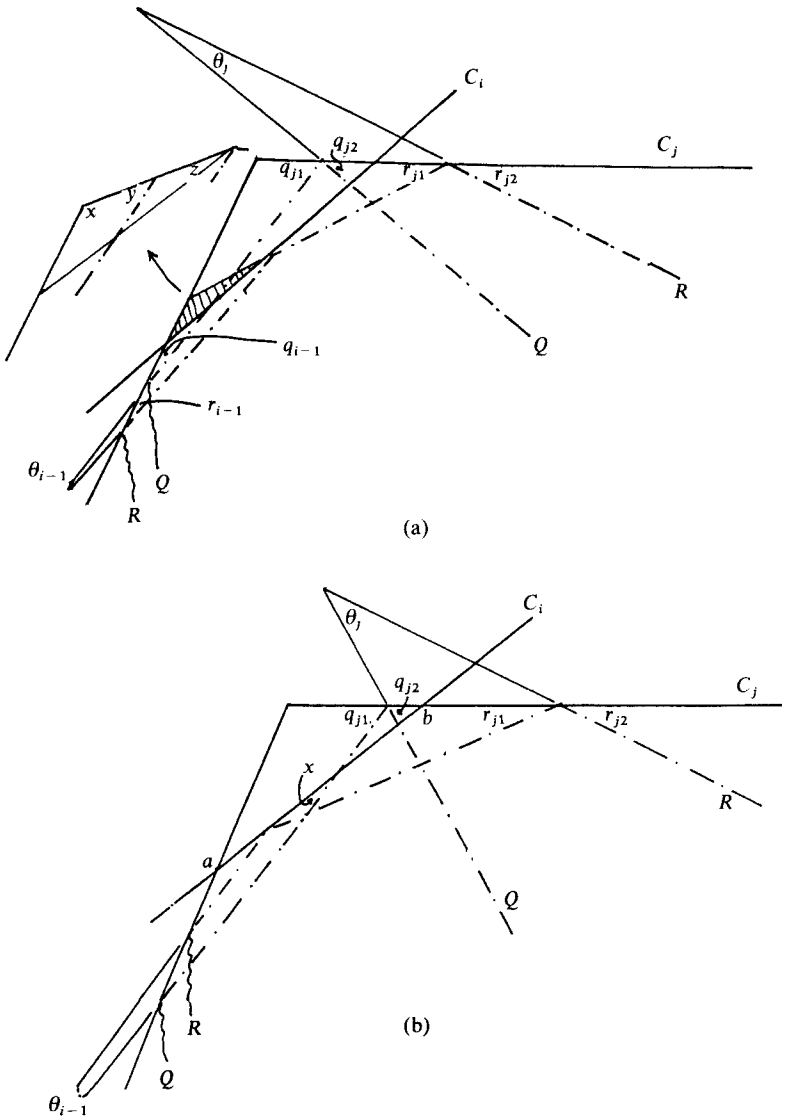

Fig. 11. Case 2.2 of Theorem 1.

divergence between the two routes increases. Consider now the effect of the remaining reflections on the angle of divergence. The additional reflections of R imply that the angle r_{i1} is smaller than in Fig. 10 and the segment of R that forms θ_j will be rotated clockwise. The additional reflections of Q imply that the segment that forms θ_j will be rotated counterclockwise. Thus, the net effect of the additional reflections is to increase θ_j even more than in Fig. 10 and it follows that $\theta_{i-1} < \theta_j$. Similarly, if VR has the pattern (...11..11...) and the corresponding pattern in VQ is (...00..01...), we have a situation similar to that of Case 2.2. Again, the effect of the

additional reflections of R is to rotate its outgoing segment clockwise and the angle of divergence increases further.

Continuing with the proof of Case 2, we have established that a difference in the characteristic vectors increases the angle of divergence between the two routes. Then the two routes cannot meet at s (note that s cannot be one of the intersection points of the two routes in the figures because, at those intersection points, one of the routes has not visited all the cuts the other has visited, i.e., if s is such an intersection, one of the routes is not a watchman route).

To complete the proof, we need to remove the assumption that the two routes contact the boundary of the polygon only at s. Clearly, a shortest watchman route may have to contact the boundary in many places. Let x be the first point at which routes R, Q separate and let R, Q next contact the boundary at points y, z, respectively. If $y = z$, we produce a contradiction exactly along the lines used above, i.e., once the routes start diverging at x, they cannot come together at point y unless at least one of them is adjustable. If $y \neq z$, consider the last straight line segments along the two routes as they move out of a corner to reach y, z. Then, either these two segments intersect before they get to y, z, or their extensions past y, z intersect (perhaps outside P). (The two segments cannot be parallel because that would imply a zero angle of divergence.) In either case, we can use this intersection point to show (using the same approach as above) that once the two routes start to diverge, they cannot come together at the intersection point unless at least one of them is adjustable. □

Corollary 1. *A watchman route R is a shortest watchman route if, and only if, R is not adjustable.*

Proof. First, assume that R is a shortest watchman route but it is adjustable. Then R can be made shorter by performing the appropriate adjustment, a contradiction. Conversely, assume that R is not adjustable but there is another watchman route Q that is shorter than R. Then Q, or some route derived from and shorter than Q, is not adjustable. This would give us two nonadjustable watchman routes contradicting Theorem 1. □

Corollary 2. *The shortest watchman route through s in a simple polygon is unique.*

4. Algorithm and Complexity

In Section 2 we described an approach for constructing the shortest watchman route in a simple polygon. We construct an initial watchman route and adjust it repeatedly until no further adjustments are possible. From the results of the previous section, it then follows that the resulting route is the shortest watchman route. The algorithm is shown below:

Algorithm: WATCHMAN-ROUTE.

1. Find all essential cuts and partition them into disjoint corners.
2. Do *Navigation* in each corner.

3. Obtain the initial route R^0 and the initial active segment set A^0. Let $i, j = 0$.
4. *While* R^j is adjustable D_o
   (a) Pick the first segment that is adjustable in A^i and construct the next active segment set, A^{i+1}.
   (b) Use A^{i+1} to construct R^{i+1} as follows:
      (1) remove the portions of the polygon that lie outside the segments in A^{i+1};
      (2) triangulate the resulting polygon;
      (3) unroll the polygon using the extended segments containing the segments in A^{i+1} as mirrors;
      (4) find a shortest path from s to its image s' in the unrolled polygon;
      (5) fold the shortest path to obtain the route R^{i+1}.
   (c) Let $j = i + 1$.
5. Report R^j as the shortest watchman route.

We start the complexity analysis of the algorithm by examining the navigation process (step 2). We have:

Lemma 7. *The "navigation in a corner" process can be performed throughout the polygon in* $O(n^2 \log n)$ *time and* $O(n^2)$ *space.*

Proof. To access the intersection points easily, prior to the navigation, we sort the intersection points in the direction from s_i to t_i on every essential cut C_i. There can be $O(n)$ intersection points on each of the $O(n)$ essential cuts. So, sorting takes $O(n^2 \log n)$ time and $O(n^2)$ space. For the navigation itself we use a vector to mark the essential cuts that have been visited and we use the sorted lists to move and search along the essential cuts. The process takes $O(n^2)$ time and space. Therefore, the overall time complexity is $O(n^2 \log n)$ and the space complexity is $O(n^2)$. □

Most of the work in the algorithm is done in step 4. In step 4(a) we choose the first among the many possible candidates that may be adjustable. We refer to this selection rule as *adjust at the first choice.* Note that the segments in each active segment set are ordered according to the index of the essential cut they are part of. In turn, the essential cuts are indexed in the order that they are first visited in a clockwise scan of the boundary of the polygon starting at s. Suppose that s_{jx} is the first segment in A^i such that R^i is adjustable at the left (right) endpoint of s_{jx} along C_j. The algorithm will next perform a (possibly empty) sequence of adjustments on cuts with index less than j before it again adjusts on a cut with index j or higher. We have:

Lemma 8. *Let* s_{jx} *be the first segment in the active segment set* A^i *such that* R^i *is adjustable at* s_{jx}. *Let* R^{i+1} *(the result of the adjustment) contact* C_j *to the left (right) of the left (right) endpoint of* s_{jx}. *Let* R^{i+d} *be the watchman route at the next time that the first adjustment is along* C_k, $k \geq j$. *Then the adjustments that take* R^i *to* R^{i+d} *(all on cuts with index less than* j*) will not cause* R^{i+d} *to contact the left (right) endpoint of* s_{jx} *(i.e., there is no oscillation on* C_j*).*

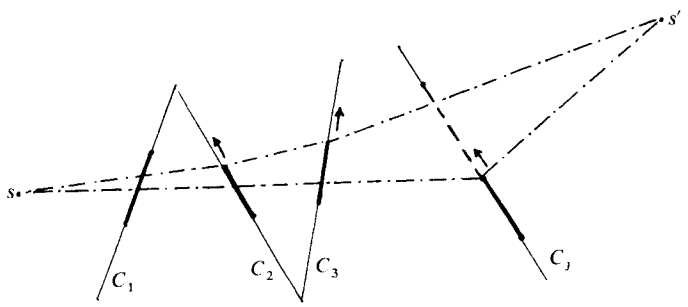

Fig. 12. Adjustments on previous cuts cannot cause oscillation along C_j.

Proof. Without loss of generality, assume that the contact point moves to the left of the left endpoint of s_{jx} when R^i is adjusted. Consider the unrolled version of the route up to C_j (Fig. 12). Clearly, the effect of moving the contact point on C_j to the left (with respect to the route) will be to (perhaps) make the route adjustable on previous cuts but we note that the adjustments made on these previous cuts will all be in the same direction, i.e., the contact point with C_j may move further left but it can never move to the right because of them. □

Theorem 2. *The time complexity of algorithm WATCHMAN ROUTE with the "adjust at the first choice" selection rule is* $O(n^4 \log \log n)$.

Proof. Lemma 8 establishes that the route cannot oscillate back and forth on C_j while adjustments resulting from an adjustment along C_j are performed on previous cuts. However, it is possible that the contact point with C_j will oscillate back and forth as adjustments on cuts with index higher than C_j are made. The worst case occurs when the unrolled path past C_j zigzags at every cut and each adjustment causes it to swing back and forth all the way across C_j each time. Since there can be at most $O(n)$ essential cuts with index higher than that of C_j, and at most $O(n)$ segments along C_j, it follows that the total number of adjustments on C_j is at most $O(n^2)$. Then the total number of adjustments performed by the algorithm is at most $O(n^3)$. For each adjustment, we need to construct the corresponding route which takes $O(n \log \log n)$. Thus, the time complexity for all the adjustments is $O(n^4 \log \log n)$. This dominates the time complexities of the other steps in the algorithm. We also note that we need $O(n^2)$ space to store the $O(n^2)$ intersection points. □

Figure 13 shows an example of the computation of the shortest watchman route. In Fig. 13(a) the extended line segments that are identified by the navigation process are shown as heavy lines. Fig. 13(b) shows the unrolled polygon with respect to these segments and the shortest path from s to s'. Folding this path produces R^0 (Fig. 13(c)). The first active segment set is

$$A^0 = \{\langle s_1, p_{12}\rangle, \langle p_{12}, p_{23}\rangle, \langle p_{23}, p_{34}\rangle, \langle p_{34}, t_4\rangle, \langle s_5, t_5\rangle\}.$$

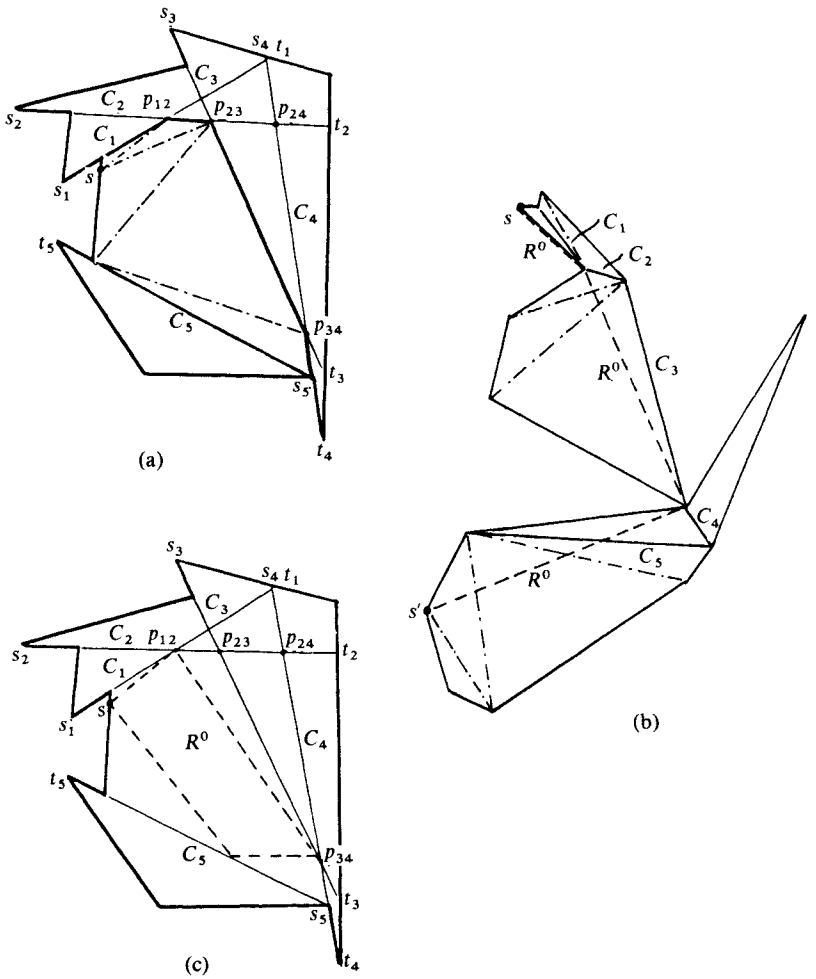

Fig. 13. The construction of (a)-(c) R^0, (d) R^1, (e) R^2, (f) R^3, and (g) the shortest watchman route R^4.

The first adjustment is a (-1)-type adjustment along C_2 resulting in

$$A^1 = \{\langle s_2, p_{12}\rangle, \langle p_{23}, p_{34}\rangle, \langle p_{34}, t_4\rangle, \langle s_5, t_5\rangle\}$$

from which we construct R^1 (Fig. 13(d)). Next comes another (-1)-type adjustment, this one along C_4, resulting in

$$A^2 = \{\langle s_2, p_{12}\rangle, \langle p_{24}, p_{34}\rangle, \langle s_5, t_5\rangle\}$$

and producing the route R^2 (Fig. 13(e)). The large shift of the reflection contact along C_4 gives rise to a $(+1)$-type adjustment along C_2. The active segment set becomes

$$A^3 = \{\langle s_1, p_{12}\rangle, \langle p_{12}, p_{23}\rangle, \langle p_{24}, p_{34}\rangle, \langle s_5, t_5\rangle\}$$

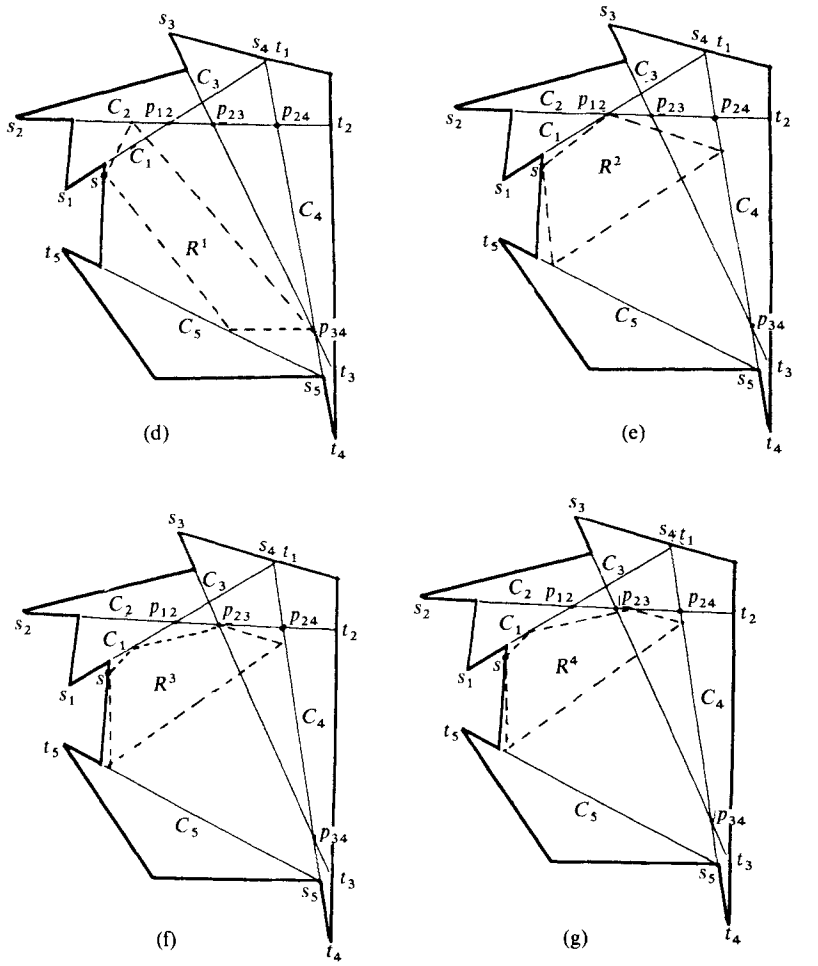

Fig. 13 (*continued*).

and the watchman route constructed from it is R^3 (Fig. 13(f)). Next, we have a 0-type adjustment at p_{23} which creates the active segment set

$$A^4 = \{\langle s_1, p_{12}\rangle, \langle p_{23}, p_{24}\rangle, \langle p_{24}, p_{34}\rangle, \langle s_5, t_5\rangle\}.$$

The route R^4 in Fig. 13(g) is constructed from A^4. This route reflects perfectly on C_1, C_2, C_4, and C_5. Since no further adjustments are possible, R^4 is the shortest watchman route.

5. Concluding Remarks

We have presented a polynomial-time algorithm for finding a shortest watchman route in a simple polygon. The complexity of the algorithm is $O(n^4 \log \log n)$

compared with the $O(n \log \log n)$ algorithm for the case of simple rectilinear polygons in [CN] and the intractability of the watchman-route problem for polygons with holes and simple polyhedra [CN].

We conjecture that the algorithm WATCHMAN ROUTE can be modified to improve its complexity ($O(n^2 \log \log n)$ seems possible). From the proof of Theorem 2, we have that the "adjust at the first choice" selection rule may lead to repetitions of the same adjustments along an essential cut C_j as the route swings back and forth along C_j due to large adjustments in cuts with higher indices. It may be possible to improve the performance of the algorithm, if we try to reduce this "swing" effect. A way to do this is to replace the "adjust at the first choice" selection rule in step 4(a) with an "adjust at Maximum Tension" rule.

Assume that a watchman route R is adjustable on cut C_i. We define the *sliding tension* of the route at C_i as the length of the vector sum of the projections on C_j of two unit vectors with origin at the contact point and directed toward s along the incoming and outgoing portions of R. This definition suggests the following alternative rule for selecting a candidate for adjustment: *Adjust at MAX Tension*: Select the segment at which the route is adjustable and the sliding tension is maximum. The new selection rule does not eliminate repeated adjustments completely but appears to reduce them significantly.

Another possible improvement to the algorithm is in the handling of 0-adjustments. Rather than treating each adjustment in a sequence of 0-adjustments as independent and reconstructing the route each time, we could group them together and adjust only once. The fact that the set of essential cuts on which reflections occur does not change due to a 0-adjustment makes this possible, However, it is not always clear how many of these adjustments can be safely grouped together at each extended line segment. We conjecture that combining the max tension rule with the grouping of certain 0-adjustments can reduce the number of adjustments performed by the algorithm from $O(n^3)$ to $O(n)$.

An $O(n \log \log n)$ algorithm for the watchman-route problem in simple rectilinear polygons is given in [CN]. An interesting related problem is to find other classes of polygons for which faster algorithms can be developed. The algorithm in [CN] does not assume a fixed starting point for the route, i.e., it finds the shortest route overall. For simple polygons, we have assumed that a starting point is specified. This makes it possible to show that the shortest watchman route is unique. In most cases, the starting point can be selected to be on the shortest watchman route, i.e., the algorithm can find the shortest watchman route overall. For example, in any polygon containing two corners that are not visible to each other, a fixed point that must be in any shortest watchman route can be easily identified. We conjecture that the shortest watchman route remains unique even when there is no fixed starting point except for very special cases where there is an infinite number of shortest routes of equal length (e.g., consider four essential cuts that form a square).

If a starting point that must lie on a shortest watchman route cannot be easily found, the algorithm can be used to obtain an approximate shortest route as follows. We select a starting point and find the shortest route through it. Then select another starting point along this route (preferably at a place least affected by

the previous choice) and reconstruct the shortest watchman route through the new starting point. If this process is repeated a few times, the resulting route will be very close to the overall optimum watchman route.

Acknowledgment

We thank one of the referees for many helpful suggestions.

References

[CN] W. P. Chin and S. Ntafos, Optimum watchman routes, *Inform. Process. Lett.* **28** (1988), 39–44; preliminary version in *Proceedings of the 2nd ACM Symposium on Computational Geometry*, pp. 24–33, 1986.

[GH*] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan, Linear time algorithms for visibility and shortest path problems inside simple polygons, *Proceedings of the ACM Symposium on Computational Geometry*, pp. 1–13, 1986.

[O] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, Oxford, 1987.

[S] S. Suri, Minimum Link Paths in Polygons and Related Problems, Ph.D. Dissertation, John Hopkins University, 1987.

[TV] R. E. Tarjan and C. Van Wyk, An O(n log log n) algorithm for triangulating a simple polygon, *SIAM J. Comput.*, **17** (1988), 143–178.

Received March 10, 1988, and in revised form February 10, 1989, and June 27, 1989.