Watchman Routes in the Presence of a Pair of Convex Polygons

Laxmi P Gewali[1]
Department of Computer Science
University of Nevada, Las Vegas
Las Vegas, NEVADA 89154

Simeon Ntafos[2]
Computer Science Program
University of Texas at Dallas
Richardson, TEXAS 75083

Abstract: Given a set of polygonal obstacles in the plane, the shortest watchman route problem asks for a closed route from which each point in the exterior of the polygons is visible to some point along the route. This problem is known to be NP-hard and the development of an efficient approximation algorithm is still open. We present an $O(n^2)$ time algorithm for computing the shortest watchman route for a pair of convex polygons, where n is the total number of edges in the polygons. We also show that the algorithm can be easily extended, without increasing its complexity, to compute the shortest watchman route when the polygons are enclosed by a third convex polygon.

1 Introduction

Problems dealing with the visibility properties of polygons have been considered by many researchers in recent years [2, 14, 17]. Computing collision-free paths in the presence of obstacles is a well known problem in computational geometry having applications in robotics and computer graphics. Visibility properties and shortest path properties are related to each other. Visibility graphs have been used to compute shortest paths in the presence of obstacles [10], and all-pair shortest paths inside a simple polygon have been used to compute visibility graphs. Collision-free paths having visibility properties have also been considered [3, 5, 8, 12]. The watchman route problem asks for a route from a point s to itself such that each point in the given space is visible to some point along the route [5]. An optimum watchman route is a watchman route of minimum length. The optimum watchman route problem is known to be NP-hard for polygons with holes and can be solved in linear time for simple orthogonal polygons [4]. An $O(n^4)$ algorithm for computing the optimum watchman route for simple polygons is reported in [5]. The watchman route problem in the exterior of a single polygon has been explored. When the polygon is monotone, convex, or rectilinear a linear time algorithm for computing the optimum external watchman route is given in [13] which also contains an $O(n^4)$ algorithm for the case of a simple polygon. The $O(n^4)$ watchman route algorithms were improved to $O(n^2)$ in [17, 18]. Recently, an approximation algorithm for computing a watchman route inside an orthogonal polygon with holes has been reported in [11]; this algorithm computes a watchman route whose length is within $O(\log n)$ of the minimum possible length and the time complexity of the algorithm is $O(n^9)$. The development of an efficient approximation algorithm is still an open problem.

---

[1] e-mail: laxmi@cs.unlv.edu. Research supported in part by a grant from US department of energy (NVOO).

[2] email: ntafos@utdallas.edu

In this paper we consider the problem of computing the shortest watchman route in the presence of a pair of convex polygons. This problem was first considered in [7] which contains an $O(n^3)$ time algorithm for it. We present an improved algorithm requiring $O(n^2)$ time. We also show that the algorithm can be easily extended to compute the shortest watchman route when the convex polygons are enclosed by a third convex polygon.

2 Preliminaries

Consider convex polygons P and Q whose vertices in clockwise order are $p_1, p_2, ..., p_k$ and $q_1, q_2, ..., q_r$, respectively. Let $n = k + r$. The notation $< q_1, q_2, ..., q_n >$ is used to denote the *polygonal chain* connecting vertices in the order $q_1, q_2, q_3, ..., q_n$. A *watchman route* for P and Q is a closed path lying outside of P and Q and such that each point in the exterior of P and Q is visible to some point along the route. The optimum watchman route is a watchman route of shortest length. A *common supporting segment* of P and Q is a segment that is tangent to both of them. Without loss of generality, we assume that polygon P is to the right of polygon Q. There are exactly four common supporting segments for a pair of convex polygons; two of them are bridge segments (*top bridge* s_t and *bottom bridge* s_b) and the other two are cross segments (*left cross* s_l and *right cross* s_r). The contact vertices of the common supporting segments s_t, s_b, s_l, s_r on polygon Q are denoted by q_t, q_b, q_l, q_r, respectively; and the corresponding contact vertices on polygon P are denoted by $p_{t'}, p_{b'}, p_{l'}, p_{r'}$, respectively (Figure 1). Bridge segments partition the boundary of each polygon into a *left-chain* and a *right-chain*; for example, $< q_t, q_l, q_r, q_b >$ is the right-chain of the boundary of Q.

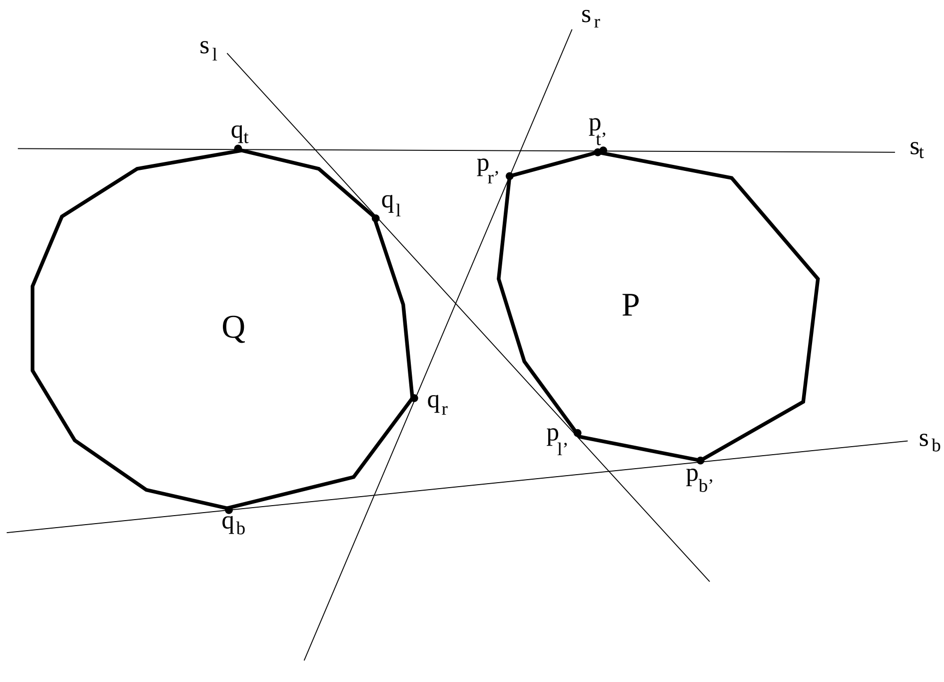


Figure 1:

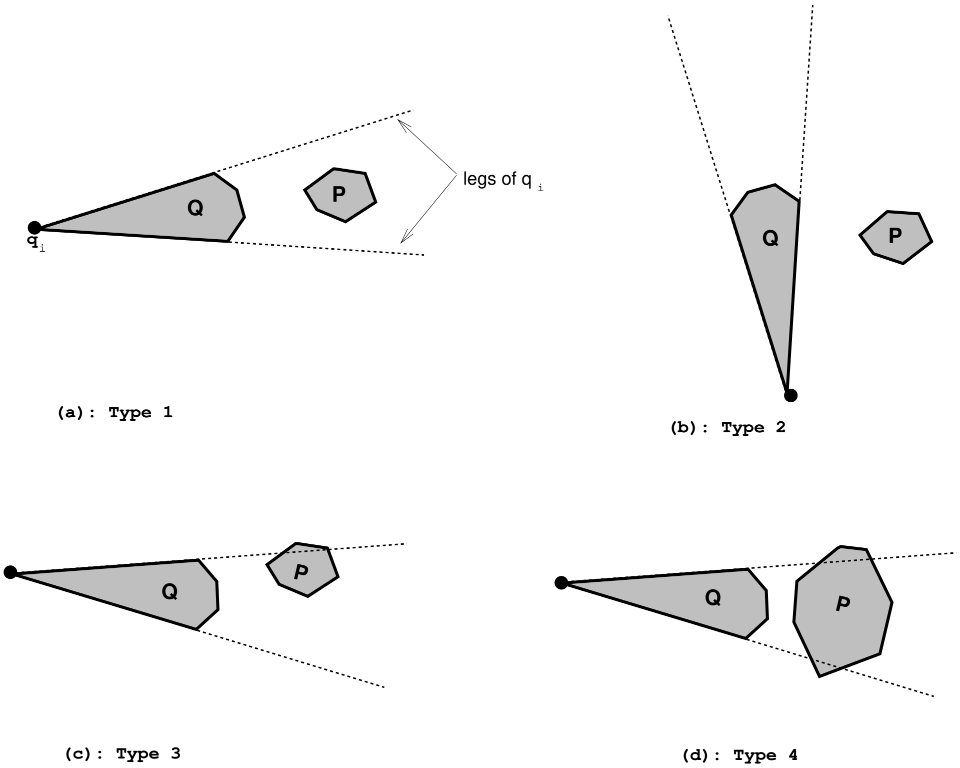


Figure 2: Four types of vertices

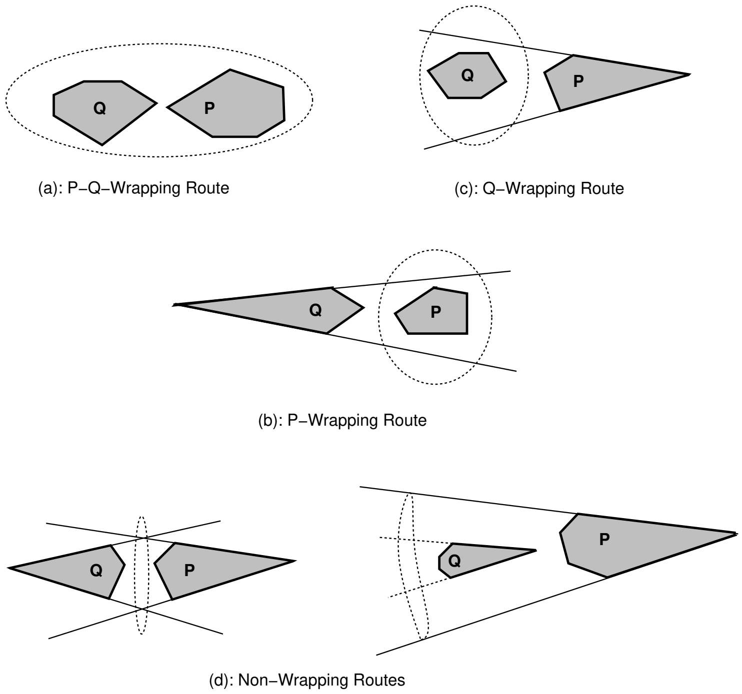


Figure 3: Four Kinds of Watchman Routes

Consider three consecutive vertices q_{i-1}, q_i, and q_{i+1} of Q. The half-lines obtained by extending edges $\overline{q_i, q_{i-1}}$ and $\overline{q_i, q_{i+1}}$ (with end points q_{i+1} and q_{i-1}) are called the *legs* of q_i. The area enclosed by the legs of q_i is the *wedge* induced by q_i. The vertices of P and Q can be distinguished into four categories: A vertex q_i of Q is a *type 1* (respectively, *type 2*) vertex if polygon P is completely inside (respectively, outside) the wedge formed by the legs of q_i. If only one leg (respectively, both legs) intersect P then it is a *type 3* (respectively, *type 4*) vertex (Figure 2).

We distinguish four kinds of watchman routes (Figure 3). Watchman routes that completely enclose only polygon P (respectively, Q) are called *P-wrapping* (respectively, *Q-wrapping*) and those that completely enclose both P and Q are called *P-Q-Wrapping.* The fourth kind that enclose neither P nor Q are called *Non-wrapping watchman routes.*

Overview of the Algorithm:

Step 1: Preprocess P and Q to determine the common supporting segments of P and Q.

Step 2: Find the convex hull route R_1 of $P \cup Q$.

Step 3: Find the shortest P-wrapping watchman route R_2 and the shortest Q-wrapping watchman route R_3. A straightforward computation of the shortest P-wrapping (or Q-wrapping) watchman route needs $O(n^2)$ time. We reduce the time complexity to $O(n)$ by relating the 2-leg P-wrapping (or Q-wrapping) watchman routes for consecutive vertices in Q (respectively P).

Step 4: Find the shortest non-wrapping watchman route R_4. A direct approach based on generating and examining $O(n^2)$ non-wrapping watchman routes leads to an $O(n^3)$ algorithm. We show how to reduce the time complexity to $O(n^2)$ by identifying sixteen cases such that the computation of the shortest watchman route in each case can be achieved in $O(n^2)$ time.

Step 5: The shortest watchman route is given by the shortest among R_1, R_2, R_3, and R_4.

In [13] it is shown that the optimum (external) watchman route for a single polygon must touch its boundary. This property does not hold for a pair of polygons. The shortest P-Q-wrapping route is given by the convex hull of $P \cup Q$. Given P and Q, the convex hull of P and Q can be computed in $O(n)$ time by using the merge hull technique [15].

Lemma 1 *The shortest P-Q-wrapping watchman route can be computed in $O(n)$ time.*

3 P-Wrapping Watchman Routes

Lemma 2 *[7] The shortest P-Wrapping watchman route must touch the boundary of P at its rightmost vertex p_r.*

Consider the wedge formed by the legs of vertex q_i. A route that connects the two legs internally through the wedge is said to *span* the wedge (Figure 4).

Observation 1: Any route that encloses P and spans the wedge of q_i is a watchman route and we call such a route a *P-wrapping route with respect to* q_i (Figure 5).

A P-wrapping watchman route with respect to q_i can span the wedge of q_i in four distinct ways as shown in Figure 5. This results in three types of P-wrapping routes: (i) a *0-leg* route extends to the exterior of the wedge from both legs, (ii) a *1-leg* route extends to the exterior only from one leg, and (iii) a *2-leg* route does not extend to the exterior.

A vertex of a watchman route is an *extreme-vertex* if it satisfies any one of the following two conditions: (i) it is a vertex where the route makes a U-turn, (ii) it does not lie on the vertices of P or Q. (Note that a vertex v of route is a U-turn if the route turns 180^0 at v.) Figure 5 illustrates extreme-vertices: (i) There are only two extreme-vertices (w and x) in the watchman route of Figure 5c, (ii) Figure 6a has no extreme-vertices, and (iii) each route shown in Figure 5b has just one extreme-vertex (y and z).

Remark 1: If a shortest P-wrapping (or Q-wrapping) watchman route contains an extreme vertex then it must lie on the leg of some vertex.

Observation 2: The shortest 0-leg P-wrapping watchman route (if it exists) is trivially given by the boundary of P.

3.1 1-Leg P-Wrapping Watchman Routes

Observe that the shortest P-wrapping watchman route with respect to a type 2 or a type 3 vertex must be a 1-leg route. Intuitively, a 1-leg P-wrapping watchman route can be obtained by stretching the boundary route of P to meet the leg of q_i not intersected by P. The shortest such route with respect to a given vertex q_i can be found by first constructing the images of P and Q and then computing the shortest path from a carefully selected object vertex to its image vertex as follows.

Let r_1 be the leg of q_i that does not intersect with P. If both legs do not intersect with P then choose the one that is farther from P. Let P' and Q' be the images of P and Q obtained by treating r_1 as a mirror (Figure 6a). By Lemma 2, the shortest 1-leg P-wrapping route with respect to q_i must pass through the rightmost vertex p_r of P.

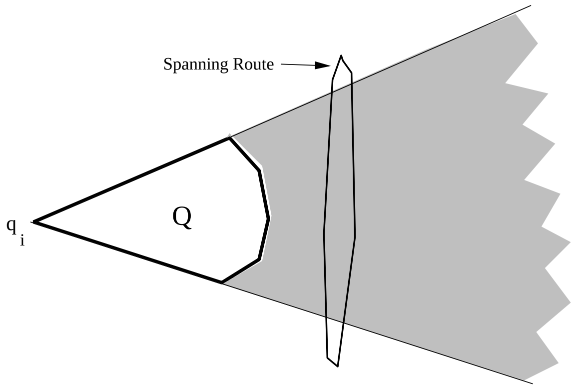


Figure 4: Illustrating a Route Spanning the Wedge of q_i

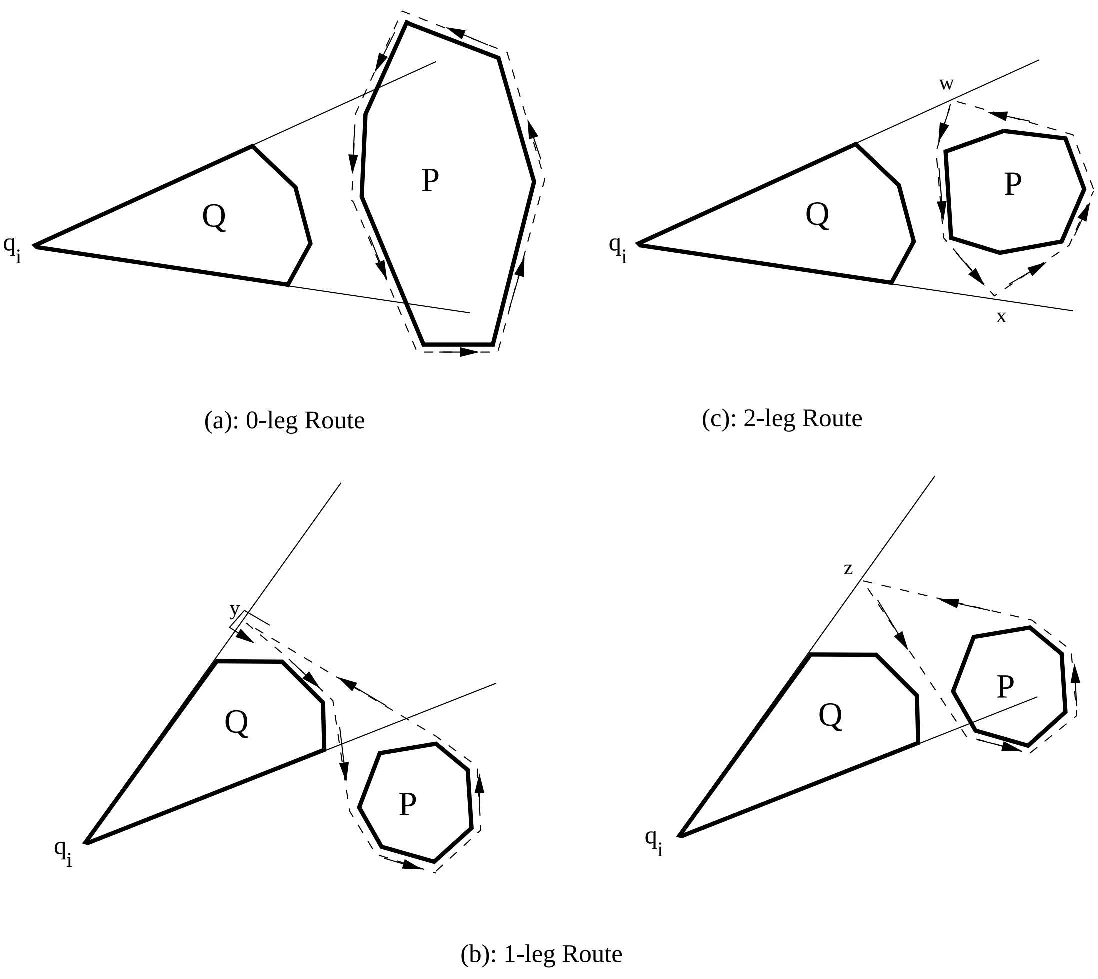
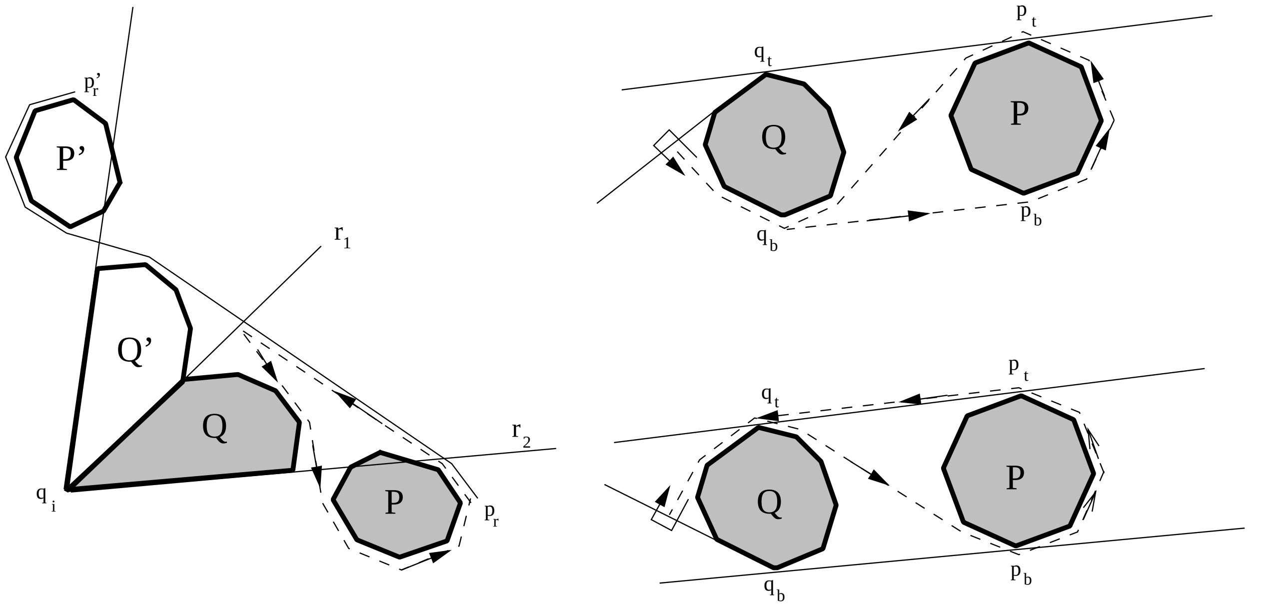


Figure 5: Three Types of P-wrapping Routes

Observation 3: Consider the images formed by the reflection of Q and P about the leg r_1 of vertex q_i (see Figure 6a). Let $S(p_r, p_r^{'})$ be the shortest path connecting the rightmost vertex p_r of P to its image $p_r^{'}$ such that P, Q, and Q' are to the left and P' is to the right of the path. Then $S(p_r, p_r^{'})$ is the unfolding of the shortest 1-leg P-wrapping watchman route with respect to q_i. In Figure 6a, the route is shown by dashed segments and its unfolding by solid segments.



(a): Illustrating Route and Path

(b): Two Ways of Constructing 1-leg P-wrapping Watchman Route

Figure 6:

Lemma 3 *The shortest 1-leg P-wrapping watchman route with respect to a given type 2 or type 3 vertex can be computed in* $O(n)$ *time.*

Proof: Images of P and Q can be constructed in $O(n)$ time. The shortest path $S(p_r, p_r^{'})$ can be computed by examining common supporting segments between P, Q, Q', and P'. Since common supporting segments between two convex polygons can be computed in $O(\log n)$ time [PS85], $S(p_r, p_r^{'})$ can be computed within $O(n)$. □

To compute the shortest 1-leg P-wrapping route, we could apply Lemma 3 repeatedly to compute 1-leg watchman routes with respect to each type 2 and type 3 vertex of Q and report the one having the least length. This clearly takes $O(n^2)$ time. The following lemma establishes that the time complexity can be reduced to $O(n)$.

Lemma 4 *The shortest 1-leg P-wrapping watchman route can be computed in* $O(n)$ *time.*

Proof: All type 2 and type 3 vertices of Q (if they exist) are in the right chain of Q. Supporting vertices q_t and q_b are either type 2 or type 3. Let l_1 and l_2 be the lengths of the

1-leg P-wrapping watchman route with respect to q_t and q_b, respectively. Let l denote the smaller of l_1 and l_2. The key observation here is that the length of the 1-leg P-wrapping watchman route with respect to any other type 2 or type 3 vertex is longer than l_1 (Figure 6b). Hence, by finding l_1 and l_2 in $O(n)$ time (using Lemma 3), the shorter of the two is the required shortest 1-leg P-wrapping route. $\square$

3.2 2-Leg P-Wrapping Watchman Routes

The shortest P-wrapping watchman route with respect to a type 1 vertex of Q must be a 2-leg route and hence it reflects from both legs. Since the shortest P-wrapping route must be in contact with the rightmost vertex p_r of P (Lemma 2), such a route can be determined by constructing images of P and Q by treating both legs as mirrors and by finding the shortest path connecting p_r and its second image. In general, the shortest 2-leg P-wrapping watchman route with respect to a type 1 vertex reflects from legs l_i and r_i by performing at most four take-offs and at most four landings on P and Q (shown by dark dots in Figure 7a). We call these take-off and landing points the *contact points* of the route with the polygons. Let P', Q', and $r_i^{'}$ be the images of P, Q, and r_i, respectively, by treating l_i as a mirror (Figure 8). Similarly, let P'' and Q'' be the images of P' and Q', respectively, by treating $r_i^{'}$ as a mirror. Let $p_r^{''}$ be the image of p_r in P''.

Observation 4: Let W denote the ordered sequence of polygons $[P, Q, Q', Q'', P'']$. Let $SP(q_i)$ be the shortest path connecting p_r to $p_r^{''}$ that keeps W to the left and P' to the right. Then $SP(q_i)$ is the unfolding of the shortest 2-leg P-wrapping watchman route with respect to q_i.

From now on we simply refer to the shortest path $SP(q_i)$ as the *unfolded path*.

Lemma 5 *The shortest 2-leg P-wrapping watchman route with respect to a type 1 vertex can be computed in $O(n)$ time.*

Proof: Let q_i be a type 1 vertex. We construct images of polygons in $O(n)$ time by treating the legs of q_i as mirrors. We then compute the shortest path X connecting p_r to $p_r^{''}$ (see Figure 8) that keeps the ordered sequence of polygons W to the left. This can be done in $O(n)$ time by computing supporting segments between consecutive polygons in W and stretching the resulting path. We next check X for its intersection with P'. If X intersects P' then we adjust X to support P' to the right by computing supporting segments between Q, P', and P''. This adjustment can be done within $O(n)$ time and the adjusted path is the unfolding of the required route. $\square$

To compute the shortest 2-leg P-wrapping watchman route naively, we could apply Lemma 5 repeatedly to all type 1 vertices of Q and select the one having the minimum length, and this would require $O(n^2)$ time. In the remaining of this section we show how the shortest 2-leg P-wrapping watchman route can be computed in $O(n)$ time.

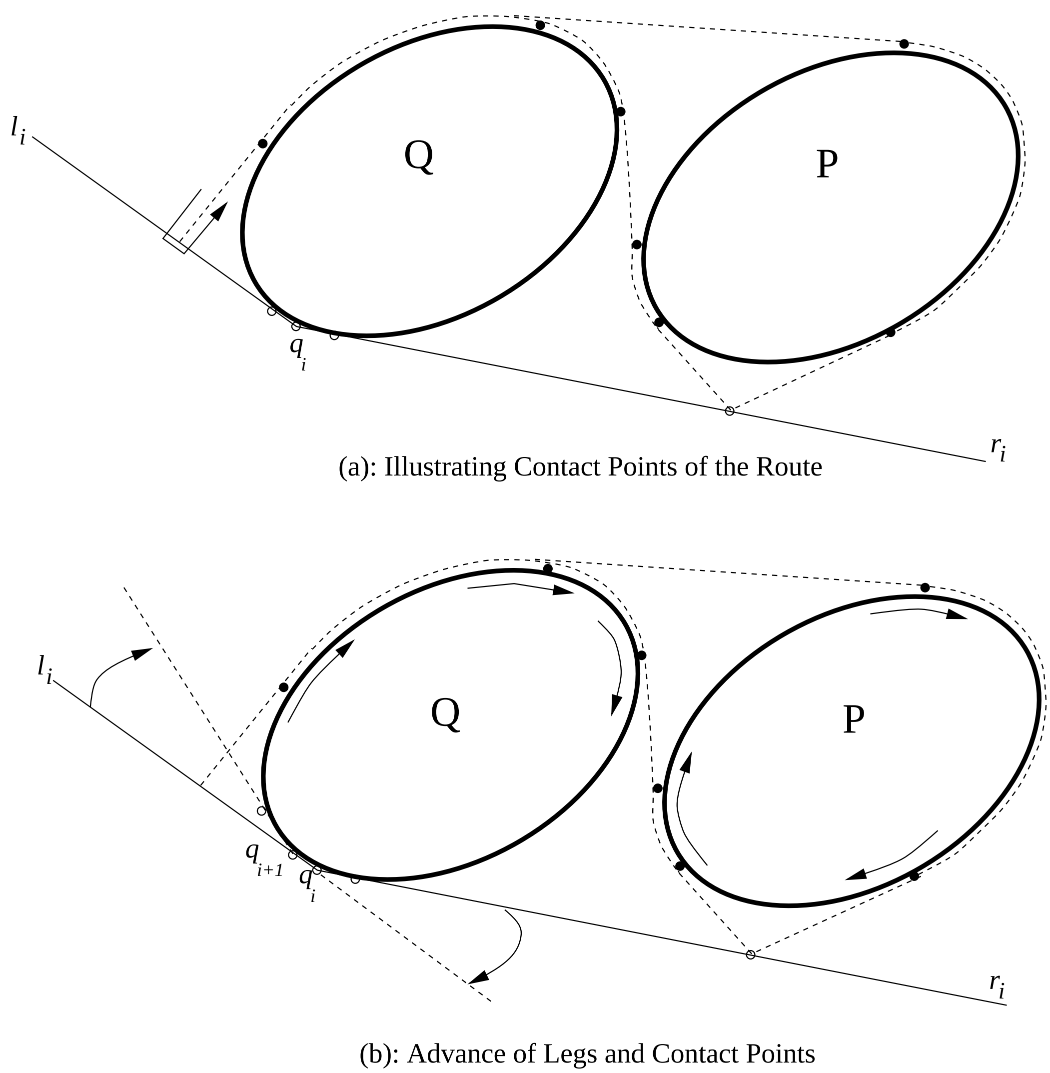


Figure 7:

Observation 5: If we advance the leg vertex clockwise from q_i to q_{i+1} and also maintain the shortest 2-leg P-wrapping watchman route then the contact vertices may advance. The key observation here is that if the leg vertex is moved in the clockwise direction (from q_i to q_{i+1}) and results in a shift of the contact vertices then the shift must be in the clockwise direction. Figure 7b illustrates the direction of the movement of contact vertices when the leg vertex is changed from q_i to q_{i+1}.

The above observation can be used to develop a faster algorithm for computing the shortest 2-leg P-wrapping watchman route. As mentioned earlier, the contact vertices of the watchman route unfold to the corresponding contact vertices in the unfolded path $SP(q_i)$. The unfolded path is not necessarily in contact with all polygons in the sequence $[P, Q, Q', Q'', P', P'']$. For polygons that are not in contact with the shortest path $SP(q_i)$, we introduce the notion of *closest vertex* (the vertex with least distance to the shortest path). In Figure 8, contact vertices are a, b, c, d, e, f, and g. Polygon Q'' is not in touch with the unfolded path and so we identify the closest vertex on it (vertex h).

Forward Rotation and Backward Rotation: In order to generate $SP(q_{i+1})$ from $SP(q_i)$, we introduce two types of rotations for the legs of a vertex: The *forward rotation* of the left leg of q_i is its gradual rotation in the clockwise direction, about the axis point q_{i+1}, so that it aligns with the edge $\overline{q_{i+1}, q_{i+2}}$; and the *backward rotation* of the right leg of q_i is its rotation in the clockwise direction, about the axis point q_i, to make it align with the edge $\overline{q_{i+1}, q_i}$ (Figure 9). Note that the legs of q_i are transformed to the legs of q_{i+1} when we apply forward rotation to the left leg and backward rotation to the right leg.

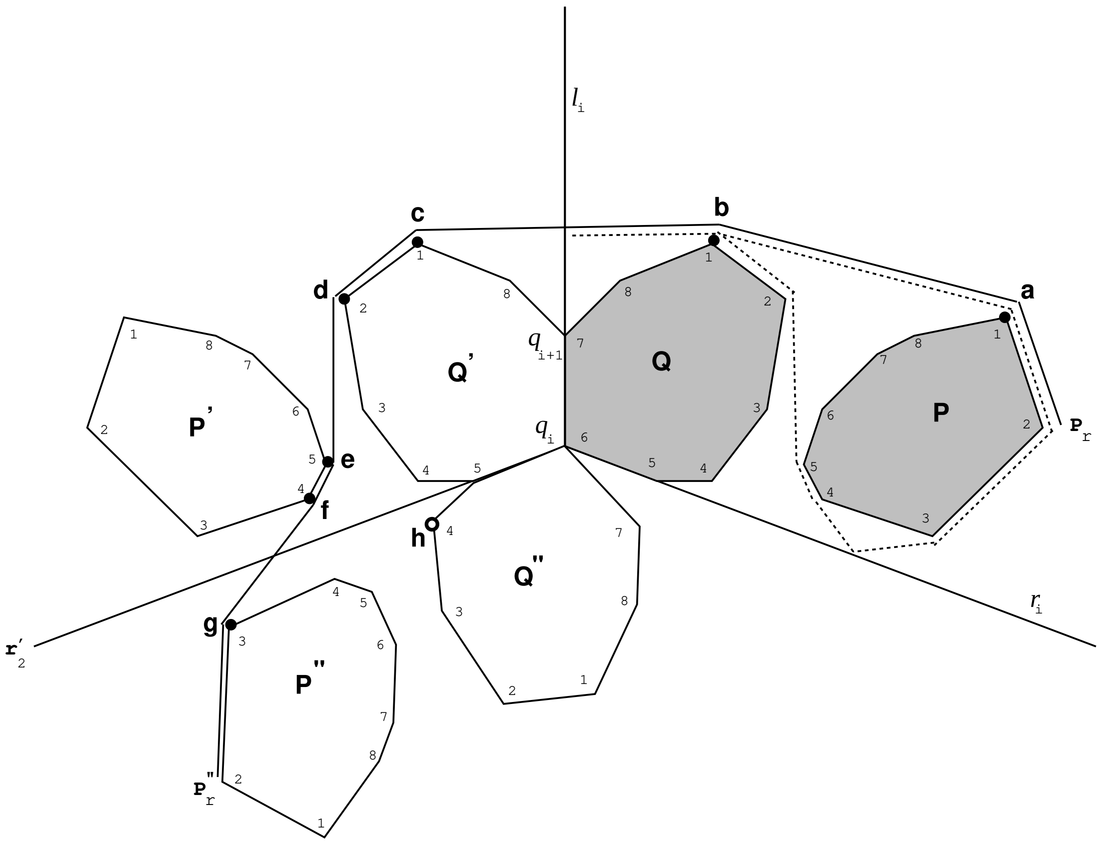


Figure 8: Illustrating the Un-wrapping of the Shortest 2-leg P-Wrapping Route

If we closely observe the change in the unfolded path with a gradual rotation of a leg of a vertex, we find that the change is not uniform. In certain intervals during the rotation, the unfolded path will pass through the same ordered sequence of vertices. But the path changes abruptly when either a vertex disappears from the path and two adjacent segments become collinear, or a new vertex enters into the path and a segment is replaced by two segments. Similarly, the closest vertices of polygons may shift during rotation. The occurrence of a shift of a contact vertex (or a shift of a closest vertex) is referred to as a *shift event*. Certain segments of the unfolded path are more important than others in triggering a shift event. The segments of the unfolded path that intersect with the boundary between the image and the object region are *crossing segments* (segments $\overline{b,c}$ and $\overline{f,g}$ in Figure 8). It turns out (Lemma 6) that the vertices on which crossing segments are incident (we call them *critical vertices*) are the only ones where a shift event can occur. For example, in Figure 8, a sufficiently small rotation of a leg of q_i does not change the relative orientation between segments $\overline{c,d}$ and $\overline{d,e}$; but any slight rotation of a leg of q_i changes the relative orientation between segments $\overline{e,f}$ and $\overline{f,g}$ (and also that between $\overline{a,b}$ and $\overline{b,c}$).

Our approach to compute the shortest 2-leg P-wrapping wrapping route efficiently is to generate $SP(q_{i+1})$ from $SP(q_i)$ by applying appropriate rotations to the legs of q_i (forward rotation to the left leg and backward rotation to the right leg) and by updating the contact vertices after each shift event.

Lemma 6 *Consider the change in the unfolded path (and a possible shift of contact vertices) when a leg of q_i is rotated slightly about q_{i+1}. If a contact vertex shifts then that vertex must be a critical vertex.*

Proof: A shift of the contact vertex occurs when the relative orientation of the segments of the unfolded path incident on a contact vertex changes. Consider now how the relative orientation of two line segments l_1 and l_2 changes when a leg is rotated. If both segments l_1 and l_2 lie inside the object region then their relative orientation does not change with the rotation of the leg mirror. Similarly, if both of them lie inside the same image region then also their relative orientation does not change. Thus, if the relative orientation of l_1 and l_2 changes with the rotation of the mirror then one of them must be a crossing segment and the vertex that shifts is a critical vertex. □

Lemma 7 *Let s be a line segment connecting an object point a with an image point b' in the presence of a line segment mirror l (Figure 10). The angular rotation required by the mirror (about a given axis point o) to result in a rotation of s by angle θ can be determined in $O(1)$ time.*

Proof: A simple geometric construction using the reflection principle can determine the required rotation. Let b' be the image of point b. We draw a ray $\overline{a,r}$ at an angle θ with the line segment $\overline{a,b'}$. We find the point of intersection b'' of the ray $\overline{a,r}$ with the circumference of the circle centered at o and radius $\overline{o,b}$. The required angle for mirror rotation is given by the angle between l and a line l' perpendicular to the line passing through b and b''. This construction can be done in $O(1)$ time. □

The *shift distance* between two paths $SP(q_i)$ and $SP(q_{i+1})$ is the total number of occurrences of shift events when $SP(q_i)$ is transformed to $SP(q_{i+1})$ by forward rotation and backward rotation of legs incident at q_i.

Lemma 8 *Given the shortest path $SP(q_i)$ and its contact and closest vertices, we can determine $SP(q_{i+1})$ in $O(d)$ time, where d is the shift distance between $SP(q_i)$ and $SP(q_{i+1})$.*

Proof: We first apply forward rotation to the left leg of q_i and update contact and closest vertices at each shift event. We next update contact and closest vertices by applying backward rotation to the right leg of q_i. Both rotations are performed in primitive steps so that exactly one shift event is triggered at the end of each primitive step. The unfolded path

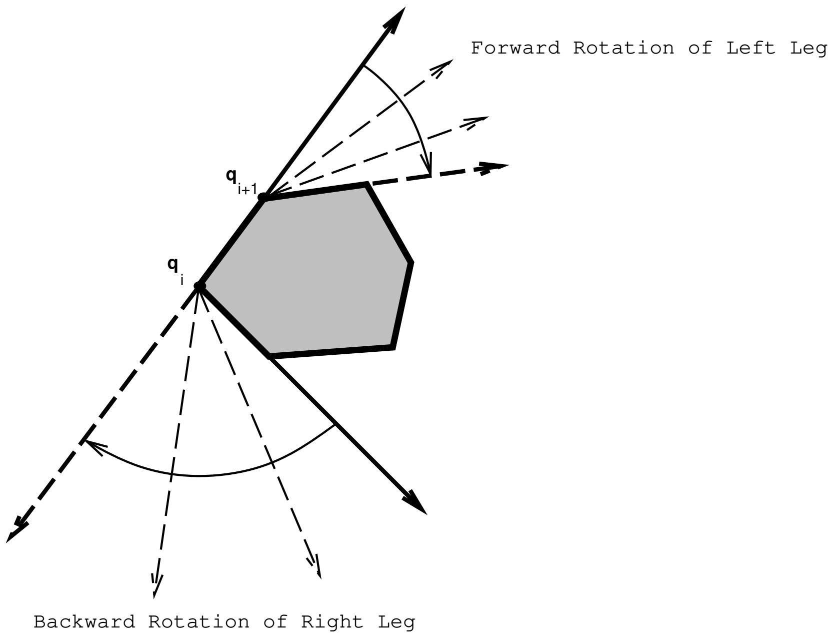


Figure 9: Illustrating the rotation of legs

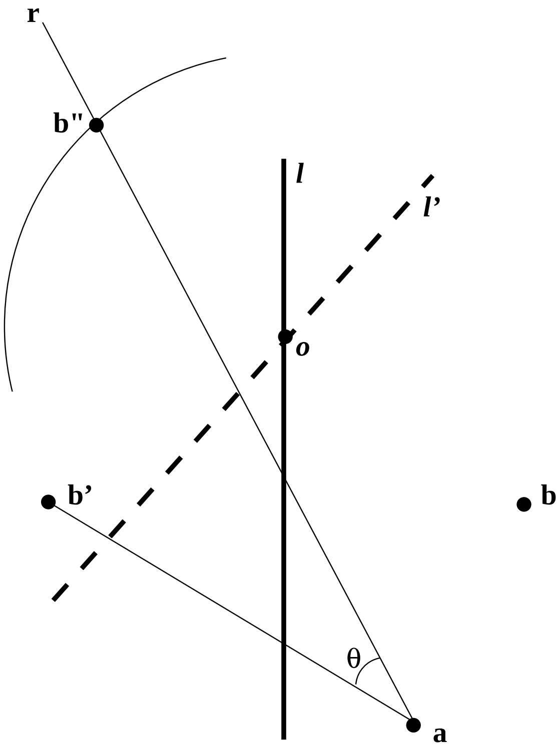


Figure 10: Proof of Lemma 7

defined by the updated contact vertices after applying the above two rotations is the required unfolded path $SP(q_{i+1})$. In order to update the contact vertex (or the closest vertex) at each shift event efficiently we need to predict the next shift event quickly, which can be done as follows. By Lemma 6, the next shift event must occur at one of the four critical vertices. By checking the angles between the crossing segment l and two edges of the polygon incident on a critical vertex x we can determine in constant time the least angular rotation $\phi(l)$ required for l so that the contact vertex shifts from x. The rotation required by the leg (mirror) to result in a given rotation of a crossing segment can be determined in constant time by using Lemma 7. Since there are only four critical vertices, the smallest rotation θ_1 required for the mirror to produce a shift of a contact vertex can be determined in $O(1)$ time. Similarly, the smallest rotation θ_2 required for the mirror to result in a shift of the closest vertex can be determined in $O(1)$ time. The smallest rotation of the mirror to result in a shift event (either a shift of a closest vertex or a shift of a contact vertex) is given by the smaller of θ_1 and θ_2. Thus the vertex at which the next shift event occurs can be predicted in $O(1)$ time.

After each primitive rotation, the images of the polygons change and so we need to recompute them. However, we need not recompute them entirely. The new coordinates can be expressed in term of old coordinates and the magnitude of the angle of rotation. At each shift event the advance of the contact vertex (or the closest vertex) occurs in the same direction. The total cost for generating the final unfolded path $SP(q_{i+1})$ is the sum of the costs for processing each shift events. Since constant work is done per shift event the total time is $O(d)$. □

Lemma 9 *The shortest 2-leg P-wrapping watchman route can be computed in $O(n)$ time.*

Proof: We can compute the unfolded path with respect to a starting type 1 vertex q_i in $O(n)$ time by using Lemma 5. We can determine the unfolded paths with respect to the remaining type 1 vertices by advancing the leg vertex in the clockwise direction and by applying Lemma 8 after each advance. It may be noted that shift events need not occur always when $SP(q_{i+1})$ is computed from $SP(q_i)$. The cost of computing a shortest path is charged either to the leg vertex or to the vertices where shift events occurred. If the number of shift events is $d(d > 0)$ then the cost of computing $SP(q_{i+1})$ is $O(d)$ and this cost is charged to the vertices where shift events occurred. On the other hand, if no shift event occurs then the cost of computing $SP(q_{i+1})$ is $O(1)$ and this cost is charged to the leg vertex. Since the advance of legs, contact vertices, and closest vertices proceed in the same direction, the total cost for computing all unfolded paths is bounded by a constant multiple of the total number of vertices in the polygons, i.e., $O(n)$. The route obtained by folding the shortest of the above paths is reported as the shortest 2-leg P-wrapping route. □

Lemma 10 *The shortest P-wrapping watchman route can be computed in $O(n)$ time.*

Proof: Follows from Observation 2, Lemma 4, and Lemma 9.

4 Non-Wrapping Watchman Routes

In this section we consider the computation of the shortest non-wrapping watchman route. Intuitively, a non-wrapping watchman route is a route that loops around neither Q nor P

or both. Some examples of non-wrapping watchman routes are shown in Figure 11. Except for the route shown in Figure 11a, all other routes are also the overall shortest watchman routes. The precise definition of a non-wrapping watchman route can be stated in term of *exposed* vertices as follows. A vertex v of P or Q is said to be *exposed* with respect to a given route r if we can draw a ray starting from v that does not intersect with r. A watchman route r for P and Q is *non-wrapping* if there exist at least one exposed vertex in each of P, Q. In Figure 11a, all vertices except p_x are exposed; all vertices in Figure 11e are exposed.

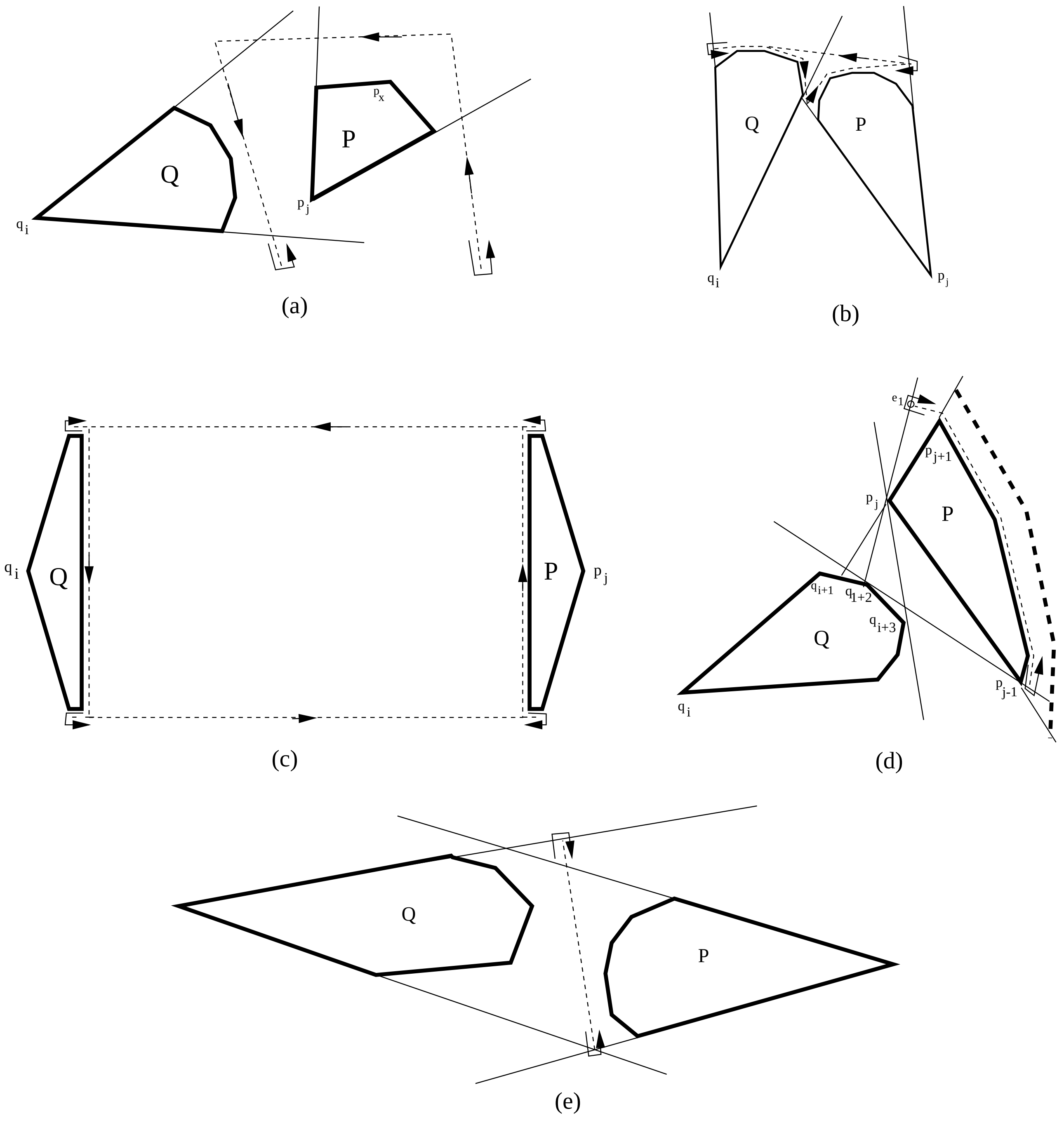


Figure 11: Examples of Non-Wrapping Watchman Routes

A non-wrapping watchman route can be defined in term of a pair of wedges as follows. A pair of wedges (formed by adjacent edges of a polygon) are said to be *matched* if one of them is formed by adjacent edges in P while the other is formed by adjacent edges in Q.

Observation 6: Any non-wrapping watchman route must span a pair of matching wedges.

The correctness of Observation 6 follows from the fact that if a non-wrapping route does not span any matching wedges then we can find a boundary point not visible from the route. However, the converse of Observation 6 is not true: A non-wrapping route that spans a pair of matching wedges need not be a watchman route. The thick dashed route shown in Figure 11d is a non-wrapping route which also spans the wedges of q_i and p_j; but it is not a watchman route. (edge $\overline{q_{i+1}, q_{i+2}}$ is not completely visible from the route.)

In Section 3 it was remarked (Remark 1) that all "extreme-vertices" (defined in Section 3) of the shortest P-wrapping watchman route must lie on the legs of vertices of Q. This property does not hold for shortest non-wrapping watchman routes. For example, the shortest watchman route shown in Figure 11d is non-wrapping and one of its extreme vertices (e_1) does not lie on any leg.

To characterize the extreme-vertices of non-wrapping watchman routes we need to incorporate the notion of "bridge-lines". A visibility edge connecting a vertex of P with a vertex of Q is a *bridge-edge* and a *bridge-line* is formed by extending a bridge-edge as far as possible without intersecting P or Q. It turns out that the extreme-vertices of the shortest non-wrapping watchman route must lie on legs or on bridge-lines. This is stated in the following lemma.

Lemma 11 *The extreme-vertices of any shortest non-wrapping watchman route must lie on legs or on bridge-lines.*

In the shortest non-wrapping watchman routes shown in Figures 11b, 11c, and 11e, all extreme-vertices lie on legs; but in the one shown in Figure 11d, one of the extreme-vertices (e_1) lies on a bridge-line.

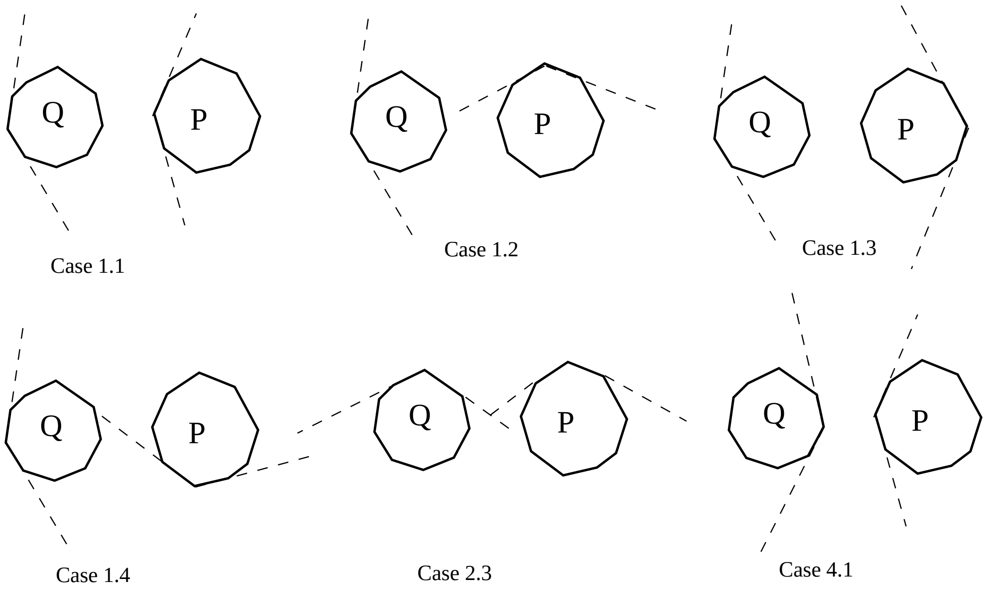
Let $U(q_i, p_j)$ denote the shortest non-wrapping watchman route that spans the wedges of q_i and p_j. Observation 6 and Lemma 11 suggest the following direct approach for computing the shortest non-wrapping watchman route: compute $U(q_i, p_j)$ for all pairs of matching-wedges and the shortest among them is the shortest non-wrapping watchman route. Since $O(n)$ time may be required to compute each $U(q_i, p_j)$, and since there may be $O(n^2)$ pairs of matching wedges, the direct approach above leads to an $O(n^3)$ time algorithm. To develop a faster algorithm we start with a closer examination of the structure of shortest non-wrapping watchman routes.

The set of vertices of polygon Q can be partitioned into four subsets, VQ_1, VQ_2, VQ_3, and VQ_4, which we call *induced-subsets* of vertices of Q. VQ_2 and VQ_4 consist of all vertices (except q_t and q_b) in the left-chain and the right-chain of Q, respectively. VQ_1 and VQ_3 contain just one vertex each, q_t and q_b, respectively. Similarly, the vertices of P are partitioned into the corresponding four induced-subsets VP_1, VP_2, VP_3, and VP_4.

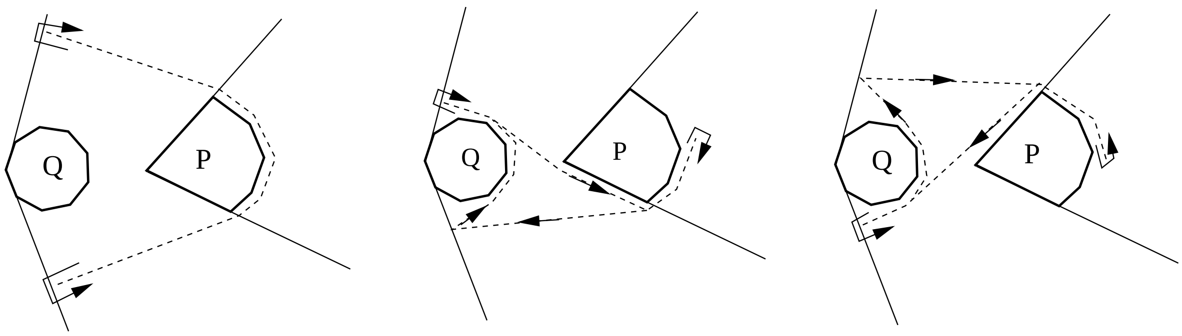
An inspection of the routes in Figure 11 suggests that the structure of the shortest non-wrapping watchman route $U(q_i, p_j)$ depends on the relative orientation of the corresponding pair of matching-wedges. Based on their relative orientation, matching-wedges can be classified into sixteen cases as illustrated in Figure 12.

Case 1.1	Case 1.2	Case 1.3	Case 1.4	···	Case 3.1	···	Case 4.4
q in VQ_1 p in VP_1	q in VQ_1 p in VP_2	q in VQ_1 p in VP_3	q in VQ_1 p in VP_4	···	q in VP_3 p in VP_1	···	q in VP_4 p in VP_4

(a): Tabulation of Sixteen Cases



(b): Illustrating Relative Orientation of Matching-Wedges



(c): Three ways of Constructing Non-Wrapping Routes for Case 1.1

Figure 12: Illustrating Sixteen Types of Non-Wrapping Watchman Routes

Each case consist of the matching wedges from the same pair of induced subsets of vertices. The cases are tabulated in Figure 12a and the matching-wedges for six selected cases are shown in Figure 12b.

Lemma 12 *The shortest non-wrapping watchman route $U(q_i, p_j)$ with respect to a given pair of matching-wedges can be constructed in $O(n)$ time.*

Proof: The construction of $U(q_i, p_j)$ depends on the case to which the relative orientation of the corresponding matching-wedges belongs. Although the basic principles for constructing the routes is the same (the principle of reflection) the details for each need to be handled appropriately. We describe the construction for two cases (Case 1.1 and 1.3); construction detail for other cases follow similarly.

In general, watchman routes for Case 1.1 can have three types of layouts (layout 1, layout 2, and layout 3) as shown in Figure 12c. Also, the shortest watchman route having any one of the three layouts must be in contact with at least one of the vertices of P, and a contact vertex can be located easily. By knowing a contact vertex (say v), we can determine the shortest watchman route having a particular layout by finding the shortest path connecting v to its image formed by reflections through the legs of q_i and p_j as done in Section 3.2. By repeating the process for each of the three layouts, the shortest among the three layouts is the required $U(q_i, p_j)$.

Observe that for all the cases, except for the Case 1.3, $U(q_i, p_j)$ must be in contact with one of the vertices of P or Q. Hence, the approach used for Case 1.1, works for all cases other than Case 1.3.

The shortest watchman route for Case 1.3 may not be in contact with P or Q. If $U(q_i, p_j)$ need not be in contact with polygons then finding a point through which the route must pass is non-trivial. We need to find the shortest path connecting a point of a leg to its image formed by reflections. Shortest path connecting a line to its image in the presence of polygons can be computed in $O(n)$ time [3]. Using this method [3], $U(q_i, p_j)$'s that need not be in contact with polygons can be computed in $O(n)$ time. □

To compute the shortest non-wrapping watchman route we could compute $U(q_i, p_j)$'s for all combinations of q_i and p_j and select the shortest one, but the time complexity of the resulting algorithm would be rather expensive: There can be $O(n^2)$ pairs of matching-wedges in some cases. (For example, in Case 1.1, there can be $O(n)$ wedges in left-chains of both P and Q.) Hence if we repeat the construction of Lemma 12 for each case then the total time to compute the shortest non-wrapping route becomes $O(n^3)$. However, we can reduce the total computation time to $O(n^2)$ if we obtain sequences of $U(q_i, p_j)$'s by computing an initial one and updating it (as in Lemma 8). This is stated in Lemma 13. Consider the notation $U(p, q)$. We extend this notation to denote set of routes as follows. $U(q, VP_i)$ is used to denote the set $\{U(q, p) | p \in VP_i\}$. Similarly, $U(VQ_i, VP_j)$ is used to denote the set $\{U(q, p) | q \in VQ_i, p \in VP_j\}$.

Lemma 13 *The shortest non-wrapping watchman route can be computed in $O(n^2)$ time.*

Proof: We process all pairs of matching-wedges in each of the sixteen cases to compute the corresponding shortest watchman routes. Consider two watchman routes $U(q_i, p_j)$ and $U(q_i, p_{j+1})$ corresponding to the same case (Case 1.1, for example). These routes have some similarity between them. Given $U(p_i, q_j)$, we can find $U(p_i, q_{j+1})$ by using the method of "advancing contact vertices and closest vertices" that was used for computing the shortest P-wrapping route (Lemma 8) in time proportional to the shift distance between $U(p_i, q_j)$ and $U(p_i, q_{j+1})$, which implies that all routes in the set $U(q_i, VP_3)$ can be determined in $O(n)$ time. Therefore, all routes in the set $U(VQ_1, VP_1)$ can be determined in $O(n^2)$ time. By repeating the process for all sixteen cases the shortest non-wrapping watchman route can be found in $O(n^2)$ time. □

Theorem 1 *The shortest watchman route for a pair of convex polygons can be computed in $O(n^2)$ time.*

Proof: Follows from Lemma 1, Lemma 10, and Lemma 13.

5 Enclosed Polygons

In this section we consider the generalization of the results of the previous section when the polygons are enclosed inside a convex polygon. Again, four kinds of watchman routes can be distinguished: (i) P-wrapping watchman routes, (ii) Q-wrapping watchman routes, (iii) P-Q-Wrapping watchman routes, and (iv) Non-wrapping watchman routes. The approach is to compute shortest watchman route of each kind and report the shortest among them as the overall shortest watchman route. We start with a few definitions. The *left hit point* LH_i for q_i is the point of intersection of the boundary of R with the left leg l_i. The *left foot point* LF_i for q_i is the point of intersection of l_i with the line that supports Q and is perpendicular to l_i. The *right hit point* RH_i, and the *right foot point* RF_i are similarly defined (Figure 13). The presence of the enclosing polygon R makes the computation of the shortest watchman route a little more involved due to the fact that the route may now touch the boundary of R. The points on the boundary of the enclosing polygon where the shortest watchman route can possibly come in contact are indicated in the following lemma.

Lemma 14 *If the shortest watchman route meets the boundary of the enclosing polygon R then it must do so at a hit point of a vertex of P, Q.*

Proof: Shortest wrapping watchman routes can not contact the boundary of R because of convexity. Other types of shortest watchman routes contact one or more legs and hit points are the intersections of legs with the boundary of R; again convexity precludes a shortest watchman route from contacting R at a point that is not a hit point and also contacting one or more legs (the route could be made shorter by "stretching" it); thus, shortest watchman routes can contact the boundary of R only at hit points. □

We first consider the computation of the shortest watchman route when R encloses only Q (Figure 13). In this case, only two kinds of route can be distinguished: (i) wrapping routes and (ii) 2-leg routes. The shortest wrapping route is trivially given by the boundary of Q. The shortest 2-leg route is found by computing 2-leg routes induced by each vertex of Q and by selecting the shortest among them as the required route. Observe that a 2-leg watchman route has two extreme vertices from which it makes U-turns to complete the route.

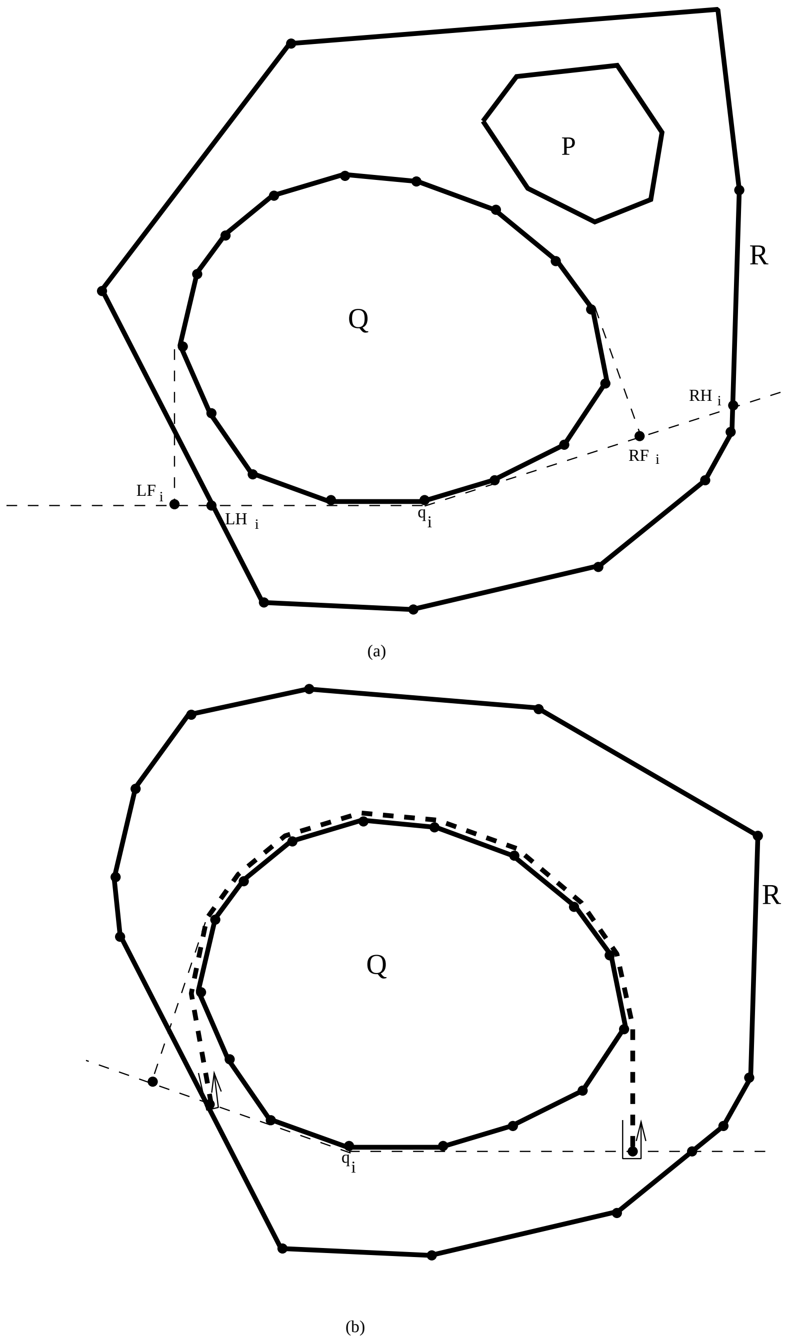


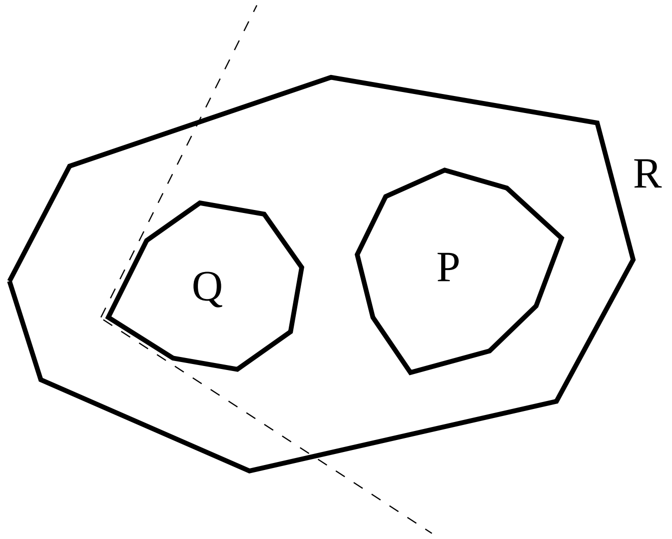
Figure 13: Watchman route for a convex annular region

A 2-leg watchman route with respect to a vertex q_i is a route having extreme vertices on the legs l_i and r_i of vertex q_i and spanning the wedge defined by the legs. The shortest 2-leg route induced by a vertex q_i can be constructed in a straightforward way by knowing its extreme points (the left extreme vertex e_l and the right extreme vertex e_r): We construct line segments supporting Q from each of the extreme vertices. The route formed by putting together supporting segments and the boundary of the polygon between the contact vertices of the supporting segments is the required 2-leg route. The extreme vertices for the shortest 2-leg route induced by a given vertex can be captured in term of the intersection of legs, boundary of R, and the supporting segments perpendicular to legs: If the left foot point LF_i is inside R then it is taken as the left extreme point e_l; otherwise, the left hit point LH_i becomes the left extreme point. The right extreme vertex e_r is identified analogously. In Figure 13b, the left extreme vertex e_l is given by the left hit point LH_i; and the right extreme point is given by the right foot point RF_i. The shortest 2-leg route induced by q_i is shown by thick dashed segments (Figure 13b).

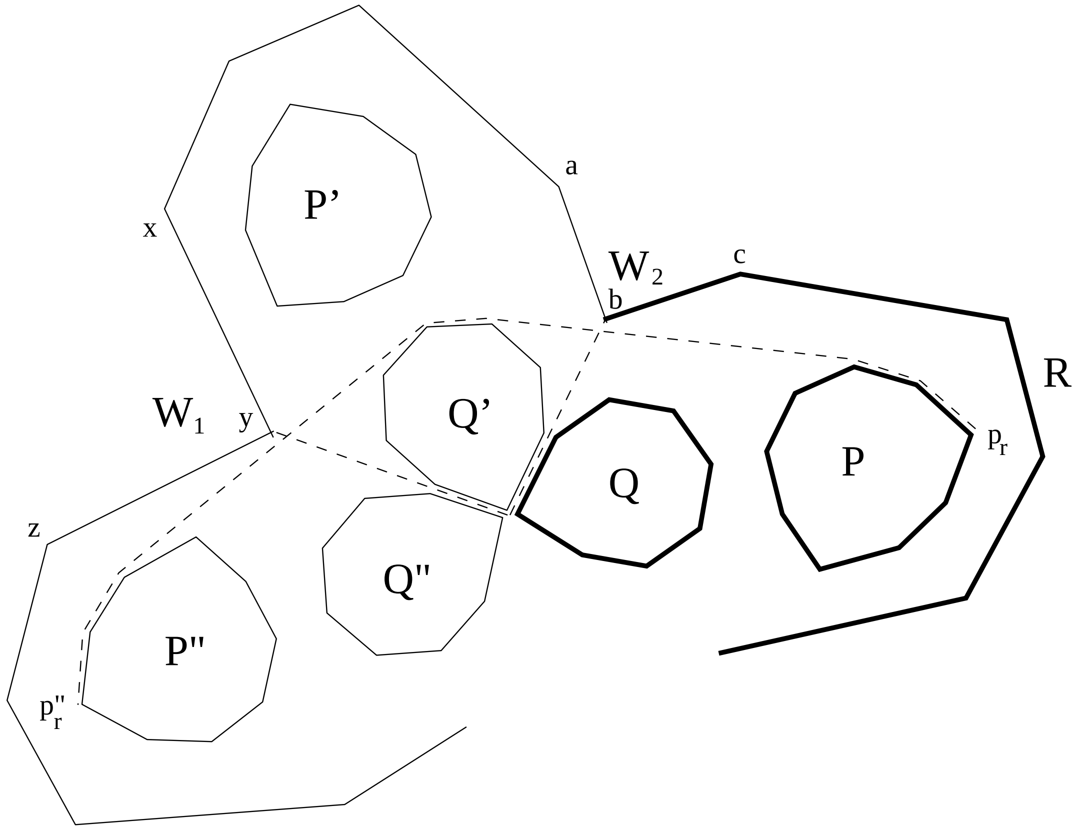
Lemma 15 *The shortest watchman route for the annular region formed by enclosing a convex polygon Q by a convex polygon R can be computed in $O(n)$ time.*

Proof: The support points and the extreme points of the shortest 2-leg route with respect to q_i can be determined in $O(\log n)$ time: Since Q is convex, the segments supporting Q and perpendicular to a leg can be computed in $O(\log n)$ time by using binary search, and hence the left foot point LF_i and the right foot point RF_i can be determined within the same time. Similarly, the left hit point LH_i and the right hit point RH_i can be determined in $O(\log n)$ time. Since R is convex, we also can check whether or not a foot point falls inside R in $O(\log n)$ time. Knowing the contact points we can read-off the shortest 2-leg route induced by q_i in $O(n)$ time by following the boundary of Q between support vertices. After knowing the shortest 2-leg route with respect to q_i we determine the shortest 2-leg route induced by other vertices by walking around the boundary of Q. When a leg vertex is advanced from q_i to q_{i+1} in the clockwise direction, the hit points in R and the support vertices in Q also advance in the clockwise direction. Hence new hit points (respectively, new contact vertices) can be determined by following the boundary of R (respectively, Q), in the clockwise direction, and by checking angles between legs (respectively, candidate supporting segments) and edges of R (respectively, Q). The time needed to compute the shortest 2-leg route induced by q_{i+1} is $O(d)$, where d is the total number of advances of support vertices and hit points. By continuing the advance, the shortest 2-leg routes with respect to each of the vertices of Q are determined. We can charge the cost for computing 2-leg routes to the vertices of the polygons. Since the advance of the hit-points and support points proceed in the same direction, only constant cost is charged to each vertex, which implies that the shortest 2-leg route can be computed in $O(n)$ time. This route is compared with the convex hull route and the shorter one is reported as the shortest watchman route. $\square$

Consider next the computation of the shortest watchman route when R encloses both P and Q. The shortest PQ-wrapping watchman route is given by the boundary of the convex hull of P and Q, which can be computed in $O(n)$ time by using merge-hull techniques. For the computation of the shortest Q-wrapping route (or the shortest P-wrapping route) we again have to look for the possibility of the watchman route meeting the boundary of R. Here also, the shortest P-wrapping route with respect to a vertex of Q can be either 0-legged, 1-legged, or 2-leged, depending on how the legs of q_i intersect with P (analogous to Figure 5). We describe the computation for the shortest 2-leg P-wrapping watchman route (computation for 0-leg and 1-leg routes are simpler). As in Section 3, we consider the images of polygons by treating legs of q_i as mirrors (first reflecting from l_i, then from r_i. An illustration is given in Figure 14. The unfolded route (the shortest path connecting p_r to $p_r^{''}$) and leg segments are shown dashed. In the unfolded diagram, inward wedges (W_1 and W_2 in Figure 14b) are formed at the points where the legs intersect the boundary of R.



(a)



(b)

Figure 14: Illustrating the construction of shortest watchman route for a pair of enclosed convex polygons

By Lemma 14, the points where the shortest P-wrapping watchman route can possibly meet the boundary of R are the tips of the wedges. Thus, the shortest 2-leg P-wrapping watchman route with respect to q_i can be constructed by finding the shortest path connecting p_r to p_r'' in the presence of eight obstacles: six polygons P, P', P'', Q, Q', Q'' and two wedges W_1 and W_2. The approach based on the "maintenance of the contact vertices and closest vertices" while we advance the leg vertex along the boundary of Q (as outlined in Lemma 8 and Lemma 9) is rehashed to compute the shortest 2-legged P-wrapping watchman route. The only difference is that the tips of wedges W_1 and W_2 can be potential contact points; this does not effect the time complexity and hence we have the following lemma.

Lemma 16 *The shortest Q-wrapping (or P-wrapping) watchman route for the region formed by enclosing two convex polygons P and Q by a convex polygon R can be computed in $O(n)$ time.*

For computing the shortest non-wrapping watchman route we extend the approach described in Section 4. There will again be sixteen cases to handle. In the unfolded diagram (corresponding to reflection from up to four legs) of each case there can be up to four wedges in addition to polygons and their images. The wedges also can be potential contact points for the route; but this does not increase the time complexity for computing the shortest non-wrapping route. Hence, the shortest non-wrapping watchman route can be computed in $O(n^2)$ time.

Theorem 2: *The shortest watchman route for the region formed by enclosing a pair of convex polygons by a convex polygon can be computed in $O(n^2)$ time.*

6 Discussion

Computing the shortest watchman routes in the presence of polygonal obstacles is intractable [4]. Recently, an approximation algorithm of time complexity $O(n^9)$ for computing the shortest watchman routes when the polygons are restricted to be orthogonal has been reported [11]. This clearly highlights the need for faster approximation algorithms. We presented an $O(n^2)$ time algorithm to compute the shortest watchman route in the presence of a pair of convex polygons. The bottle-neck of the algorithm is the computation of non-wrapping routes and faster ways to compute them will also reduce the overall complexity of the algorithm. We conjecture that an $O(n)$ algorithm is possible if a consistent way is found to allow legs from both P and Q to rotate while keeping the contact/closest points advancing in the same direction. The approach based on considering all possible types of routes can be extended to find shortest watchman routes in the presence of any fixed number k of convex polygons in time polynomial in n but exponential in k.

Another interesting problem is to find shortest watchman routes when the enclosing polygon is simple. Watchman route algorithms for the interior of a polygon are based on the notion of "essential lines" [5], i.e., extensions of polygonal edges that need to be visited by any watchman route in order to view areas around corners. When obstacles are introduced inside the polygon, essential lines are not well defined because the work of viewing a certain area may be split between two (or more) parts of the route.

References

[1] A. Aggarwal, The Art Gallery Theorem: Its Variations, and Algorithmic Aspects, PhD Thesis, John Hopkins University, 1984.

[2] J. Bhadury, V. Chandru, A. Maheshwari, and R. Chandrasekaran, "Art Gallery Problems for Convex Nested Polygons," to appear in *INFORMS Journal of Computing*, 1996.

[3] J. Czyzowicz, P. Egyed, H. Everett, D. Rappaport, , T. Shermer, G. Toussaint, and J. Urrutia, "The Aquarium Keeper's Problem," *The Second SIAM Symposium on Data Structure and Algorithms*, 1991.

[4] W. P. Chin and S. Ntafos, "Optimum Watchman Routes," *Information Processing Letters*, Vol. 28, pp. 39-44, May 1988.

[5] W. P. Chin and S. Ntafos, "Watchman Routes in Simple Polygons," *Discrete and Computational Geometry*, Vol. 6, No. 1, pp. 9-31, January 1991.

[6] S. Carlsson, B. J. Nilsson, and Simeon Ntafos, "Optimum Guard Covers and m-Watchman Routes for Restricted Polygons", *Lecture Notes in Computer Science*, 519, 1991.

[7] L. Gewali and R. Lombardo, "Watchman Routes for a Pair of Convex Polygons", Lecture Notes in Pure and Applied Mathematics, No. 141, 1993.

[8] L. Gewali and I. Stojmenovic, *"Computing External Watchman Routes on PRAM, BSR, and Interconnection Models of Parallel Computation"*, *Parallel Processing Letters*, Vol. 4, No. 1, 1994, pp. 83-93.

[9] D. T. Lee and A. K. Lin, "Computational Complexity of Art Gallery Problems", *IEEE Transactions on Information Theory*, IT-32, pp. 276-282, 1986.

[10] T. Lozano-Perez and M. Wesley, "An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles", *Communications od the ACM*, 22, 1979.

[11] Cristian S. Mata and Joseph S. B. Mitchell, "Approximation Algorithms for Geometric Tours and Network Design Problems", *11th Annual Symposium on Computational Geometry*, 1995.

[12] J. S. B. Mitchell and E. W. Wynters, "Watchman Routes for Multiple Guards", *Proceedings of the 3rd Canadian Conference on Computational Geometry*, 1991.

[13] S. Ntafos and L. Gewali, "External Watchman Routes," *The Visual Computer*, Special Issue on Computational Geometry, Vol. 10, No. 8, pp. 474-483, 1994.

[14] J. O'Rourke, "Art Gallery Theorems and Algorithms," *The International Series of Monographs on Computer Science*, Oxford University Press, 1987.

[15] F. P. Preparata and M. I. Shamos. "Computational Geometry - An Introduction," Springer Verlag, 1985.

[16] T.C. Shermer, *Recent Results in Art Galleries*, Proceedings of the IEEE, Sept. (1992), pp. 1384-1399.

[17] X.-H. Tan, T. Hirata and Y. Inagaki, "An incremental Algorithm for Constructing Shortest Watchman Routes", Proc. ISA-91, pp. 163-175, 1991.

[18] X.-H. Tan, T. Hirata and Y. Inagaki, "Constructing Shortest Watchman Routes by Divide and Conquer," Proc. ISAAC-93, pp. 68-77, 1993.

[19] G. T. Toussaint (Editor), "Computational Morphology," North Holland, 1988.