Some Aperture-Angle Optimization Problems*

(1) *Prosenjit Bose*
Department of Computer Science
University of British Columbia
Vancouver, Canada

(2) *Ferran Hurtado-Diaz*
Departamento de Matemáticas 2
Universitat Politècnica de Catalunya
Barcelona, Spain

(3) *Elsa Omaña-Pulido*
Departmento de Matemáticas
Universidad Autónoma de Mexico
Mexico City, Mexico

(4) *Jack Snoeyink*
Department of Computer Science
University of British Columbia
Vancouver, Canada

(5) *Godfried T. Toussaint*
School of Computer Science
McGill University
Montreal, Canada

ABSTRACT

Let P and Q be two disjoint convex polygons in the plane with m and n vertices, respectively. Given a point x in P, the *aperture angle* of x with respect to Q is defined as the angle of the cone that: (1) contains Q, (2) has apex at x, and (3) has its two rays emanating from x tangent to Q. We present algorithms with complexities $O(n \log m)$, $O(n + n \log (m/n))$ and $O(n + m)$ for computing the maximum aperture angle with respect to Q when x is allowed to vary in P. To compute the minimum aperture angle we modify the latter algorithm obtaining an $O(n + m)$ algorithm. Finally, we establish an $\Omega(n + n \log (m/n))$ time lower bound for the maximization problem and an $\Omega(m + n)$ bound for the minimization problem thereby proving the optimality of our algorithms.

Keywords: aperture-angle, convexity, unimodality, discrete optimization, algorithms, complexity, computational geometry, robotics, visibility.

1. Introduction

Visibility plays an important role in the manufacturing industry in such problems as accessibility analysis in machining [ABP93], [Wo94], [TWG92], [CW92] and visual inspection [SR90] as well as computer graphics, robotics, computer vision, operations research and several other disciplines of computing science and computer engineering [O'R87], [Sh92]. The traditional model

* Research of the first, third and fifth authors was supported by grants NSERC no. OGP0009293 and FCAR no. 93-ER-0291. Research of the second author was supported by Acción CICYT no. 0747-C02-02-E. Research of the fourth author was supported by NSERC Grant no. OGP0121831 and a fellowship from the B.C. Advanced Systems Institute.

of visibility investigated in computational geometry allows for a guard or camera to "see in all directions," i.e., the *aperture angle* is idealized to be 360 degrees. More recently, computational geometry research has begun investigating more realistic models of visibility where the aperture angle (or *field-of-view* angle as it is called in robotics [CDGP], [Co88]) is restricted to be some angle θ less than 360 degrees. For example, given a convex polygon and a camera with aperture angle θ situated outside the polygon, Teichman [Te89] computes a description of all the points in space where a camera may be placed in such a way that the polygon lies completely in the field of vision of a camera with aperture angle θ. A member x of a set of points S is said to be θ-*visible* if a camera with aperture angle θ can be placed on x in such a way that no other member of S lies in the camera's field of vision. Avis, et al. [ABD93] obtained optimal algorithms for finding all the θ-visible points in such a set. Devroye and Toussaint [DT93] investigate the cardinality of the θ-visible points among a set of special points which are the intersections of a set of random lines. Finally, in another variant of the problem Bose, et al., [BGL93] have shown that n cameras, each with specified aperture angle not exceeding 180 degrees, can be placed at n fixed locations in the plane to see the entire plane if and only if the aperture angles sum to at least 360 degrees.

The simplest of these types of problems is often found as a exercise in calculus texts and called the "picture-on-the-wall" problem (see for example [Sc60], p. 427, problem # 20). In this problem a picture hangs on the wall in a museum above the level of an observer's eye. How far from the wall should the observer stand to maximize the angle at the observer's eye determined by the top and bottom of the picture? While this problem is easily solved with calculus, an elegant solution that does not use calculus has been known for some time [Ni81]. This same solution holds for the more general problem where the picture may not be orthogonal to the floor [VG80].

In this paper we consider a generalization of the "picture-on-the-wall" problem, namely, the problem of computing the aperture angle of a camera that is allowed to travel in a convex region in the plane and is required to maintain some other convex region within its field of view at all times. More specifically, let P and Q be two disjoint convex polygons in the plane with n and m vertices, respectively. Given a point x in P, the *aperture angle* of x with respect to Q is defined as the angle of the cone that: (1) contains Q, (2) has apex at x, and (3) has its two rays emanating from x tangent to Q. We present an $O(n + m)$ time algorithm for computing the *minimum* aperture angle with respect to Q when x is allowed to vary in P. We also present algorithms with complexities $O(n \log m)$, $O(n + n \log (m/n))$ and $O(n + m)$ for computing the maximum aperture angle with respect to Q when x is allowed to vary in P. Finally, we establish an $\Omega(n + n \log (m/n))$ time lower bound for the maximization problem and an $\Omega(m + n)$ bound for the minimization problem thereby proving the optimality of our algorithms.

2. Geometric Preliminaries

In this section, we develop some geometric tools and solve several special cases of the general problems that will be used subsequently to solve the general problems. The model of computation used for the algorithms is the extended real RAM (for details refer to [PS88]). We begin with a few basic observations from Euclidean geometry. Let a, b and x be points on a circle C. Let y be a point in the open halfplane (defined by the line through a and b) that contains x. Let $ang(abc)$ denote the angle at b in triangle abc.

Observation 2.1: If y lies in the exterior of circle C then $ang(ayb) < ang(axb)$ (refer to Fig. 1 (a))

Observation 2.2: If y lies on the circle C then $ang(ayb) = ang(axb)$ (refer to Fig. 1 (b))

Observation 2.3: If y lies in the interior of circle C then $ang(ayb) > ang(axb)$ (refer to Fig. 1 (c))

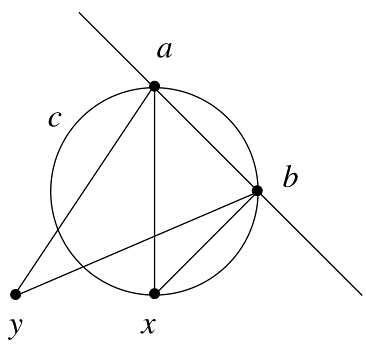

Fig. 1 (a)

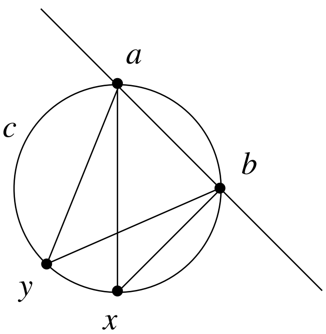

Fig. 1 (b)

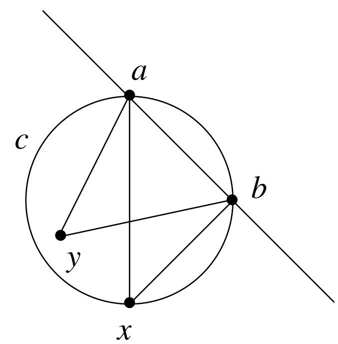

Fig. 1 (c)

The first simplification of the general problems will be referred to as the *Line-to-Segment Problem*, where the convex polygon Q (the object that must be kept in the field of view) is replaced by a segment ab and the convex polygon P (the region where the camera is allowed to roam) is replaced by a line L. Note that this is precisely the "picture-on-the-wall" problem for which a solution is known [Ni81], [VG80]. These authors however only give characterizations of the solution. On the other hand, motivated by the desire to obtain efficient algorithms, we will also characterize the aperture-angle function itself.

The Line-to-Segment Problem

Problem: Given a segment ab and a line L that does not intersect ab, find a point $x \in L$ such that the angle axb is a maximum.

Without loss of generality assume the line L is the x-axis. When the segment ab is parallel to the line L, the solution point x must lie at the perpendicular projection of the midpoint of ab on L. Thus, we can turn our attention to the case where ab is not parallel. Let t be the intersection point of the line containing the segment ab and the line L. Observe that the minimization problem is trivial since the aperture angle of t with respect to ab is zero. The point where the aperture angle is a maximum, however, lies in either of the open sets $(-\infty, t)$ or (t, ∞). Let $\theta(x)$ denote the aperture-angle function (i.e. the aperture angle from a point x on L, the real line, with respect to a given line segment ab, as x varies from $-\infty$ to $+\infty$).

Lemma 2.1: If x is constrained to the interval $(-\infty, t]$, then the function $\theta(x)$ reaches its maximum at the point $y \in (-\infty, t]$ where the circle through a, b and y is tangent to L. Furthermore, $\theta(x)$ is upwards unimodal in $(-\infty, t]$.

Proof: Let C be the circle through a and b that is tangent to L at a point $y \in (-\infty, t]$. For all points $x \in (-\infty, t]$ with $x \neq y$, $\theta(y) > \theta(x)$ by Observation 2.1. Thus, we have established that y yields a maximum. We will now show that the function $\theta(x)$ is upwards unimodal. We consider two cases depending on whether or not the center of C lies on the same side of the line through ab as y.

Case 1: The center of C lies on the same side of ab as y. Let $x_1, x_2 \in (-\infty, t]$ with the property that $x_1 < x_2 < y$ and refer to Fig. 2. Since the circle C is tangent to L at y, when C is enlarged continuously with the constraint that it pass through a and b, the growing circle first intersects x_2 and subsequently x_1. Therefore the circle through a, b and x_2 is smaller than the circle through a, b and x_1. But since the chord ab is the same length in both circles, the angle it induces is smaller in the larger circle. Therefore $\theta(x_1) < \theta(x_2)$.

Case 2: The center of C lies on the side of ab not containing y. A similar argument holds where the circle first shrinks continuously until ab defines its diameter, after which it grows continuously. It follows that $\theta(x)$ is strictly increasing in $(-\infty, y]$. Similar arguments show that $\theta(x)$ is strictly decreasing in $[y, t]$. Therefore $\theta(x)$ is upwards unimodal in the interval $(-\infty, t]$. Q.E.D.

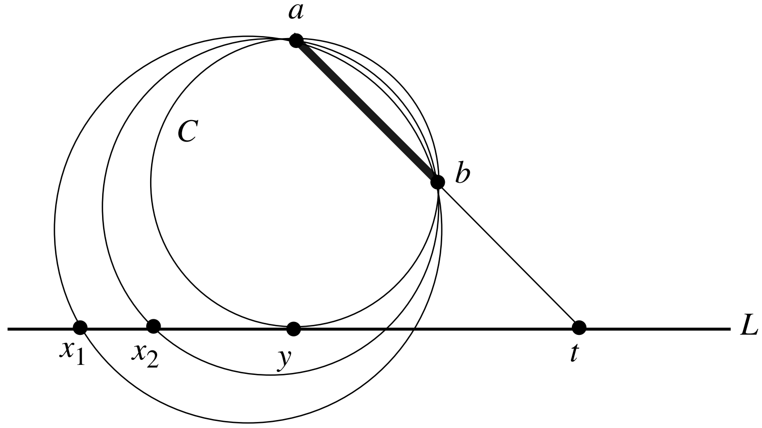

Fig. 2

Lemma 2.1 can also be established for $x \in [t,+\infty)$ in a similar way.

The following corollaries are immediate consequences of lemma 2.1.

Corollary 2.1: Let I be a closed interval contained in L and let y and y' be the points where the two circles through a and b are tangent to L. The maximum aperture angle, over I, with respect to ab is reached at either y or y' or at an end point of I.

Corollary 2.2: Let I be a closed interval contained in L that does not contain point t. Then the minimum aperture angle, over I, with respect to ab is reached at an end point of I.

We now take a step closer to the general problem and consider a simplification we refer to as the *Polygon-to-Segment Problem*, where the object that must be kept in the field of view is still a segment ab but the region where the camera is allowed to roam is a convex polygon P.

The Polygon-to-Segment Problem

Problem: Find a point x in a convex polygon P such that $\theta(x)$ is a maximum with respect to a given segment ab that does not intersect P.

In order to present the solution to this problem, we first define some geometric concepts related to the solution. Unless stated otherwise, we assume throughout the paper that the vertices of the polygon are given in counterclockwise order (refer to Figure 3).

Definition: A line L is a *critical separating line of support* of P and ab if it (1) separates P from ab, and (2) it is tangent to both P and ab.

Let the critical separating lines of support of P and ab be tangent at $\{p_j, a\}$ and $\{p_i, b\}$ respectively (see Fig. 3). If these lines are colinear with edges of P, then let p_j and p_i be the end points of these edges that are furthest from a and b, respectively. These lines partition the boundary of P into two chains. They also partition the plane into four regions (or cones), two of which are empty, one of which contains P and the other ab. Denote the region containing P by R_P. Now, the line segment p_ip_j partitions R_P into a triangle and an unbounded region. The chain $(p_i, p_{i+1},..., p_j)$ contained in the triangle (possibly consisting of a single vertex) is referred to as the *inner boundary of P with respect to ab*, and is denoted by $IB(P)$. The complementary chain is denoted by $IB(P)^c$. Note that p_i and p_j are assumed to be contained in both $IB(P)$ and the complement $IB(P)^c$.

Let $int(P)$, $ext(P)$ and $bd(P)$ denote the interior, exterior and boundary, respectively, of polygon P. If line $L(a, b)$ passing through ab intersects $int(P)$, the chain $IB(P)$ is contained in the triangle (p_i, c, p_j), where p_i and p_j are the two tangent points as defined above and c is the extreme point of the segment ab that is closer to P (using the definition of distance from a point a to a polygon P

as $min\ \{d(a, x) \mid x \in P\}$ and d is the euclidean distance). Thus, $L(a, b)$ divides the convex polygon P into two convex polygons P_1 and P_2, where $L(a, b)$ does not intersect the interior of either and $IB(P)$ is partitioned into $IB(P_1)$ and $IB(P_2)$. Furthermore, the solution to our problem for P will be the maximum of the two solutions obtained for the two problems on P_1 and P_2 separately since on $L(a, b)$ the maximum aperture angle is zero. Therefore, to solve the Polygon-to-Segment problem, we may assume that $L(a, b)$ does not intersect $int(P)$.

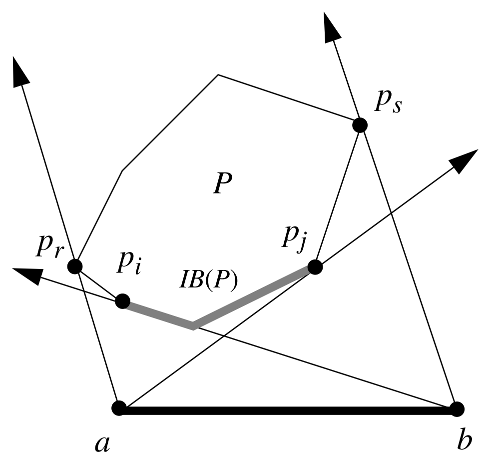

Fig. 3

Lemma 2.2: A point $x \in P$ where the aperture angle reaches the maximum value lies on the chain *IB(P)*.

Proof: (by contradiction) Let x be the point that maximizes the aperture angle and let it not be contained in $IB(P)$. Let the supporting rays from x be denoted by $ray(x, a)$ and $ray(x, b)$, let $cone(x)$ denote the unbounded region of the plane determined by $ray(x, a)$ and $ray(x, b)$ that contains segment ab, and refer to Figure 4. It suffices to demonstrate that $IB(P)$ intersects $int(cone(x))$, for then triangle abx must contain a point y of $IB(P)$ in its interior for which $\theta(y) > \theta(x)$, a contradiction. Therefore assume $IB(P)$ does not intersect $int(cone(x))$. Let c and d be the end points of $IB(P)$ such that d lies on the critical separating tangent through end point b of segment ab and c lies on the critical separating tangent through end point a of segment ab. Let $ray(d, \overline{b})$ denote the ray starting at d in a direction *away* from b and let $ray(c, \overline{a})$ denote the ray starting at c in direction *away* from a. Since $IB(P)$ does not intersect $int(cone(x))$, the $cone(x)$ can intersect at most one of $ray(d, \overline{b})$, $ray(c, \overline{a})$. Without loss of generality, assume $cone(x)$ intersects $ray(d, \overline{b})$. This implies that point b lies in $ext(cone(x))$, a contradiction. Q.E.D.

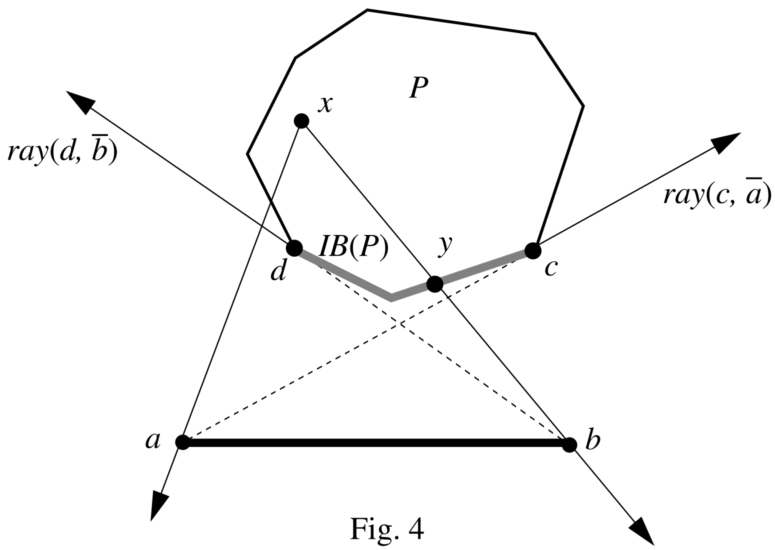

Fig. 4

The initial problem is now reduced to that of finding a point $x \in IB(P)$ such that $\theta(x)$ is a maximum with respect to *ab*. The following result shows that the function $\theta(x)$ has a unique maximum point.

Lemma 2.3: The maximum aperture angle is reached at a unique point $x \in IB(P)$.

Proof: Consider the infinite radius circle through *ab* that does not contain *P*. Consider the continuous transformation of this circle as its center travels along the perpendicular bisector of segment *ab*. By lemma 2.1 the maximum aperture angle is reached at the point where the circle first touches *P*. But a circle tangent to a convex polygon intersects the polygon at a unique point. Q.E.D.

Lemmas 2.2 and 2.3 establish the existence of a unique global maximum over *IB*(*P*). However, this in itself does not preclude the existence of other possible *local* maxima. Fortunately, we are able to show that $\theta(x)$ is an upwards unimodal function over *IB*(*P*), a crucial property that we will exploit subsequently for obtaining efficient algorithms.

Lemma 2.4: The function $\theta(x)$ with respect to the segment *ab* is *upwards unimodal* over *IB*(*P*).

Proof: Let *C* be the circle that contains *a* and *b*, tangent to *P* and let *x* be the point at which tangency occurs. The point *x*, where the maximum aperture angle is reached, can lie in the interior of an edge or on a vertex of the polygon *P*, but by lemma 2.2 it must lie in *IB*(*P*). For every edge $(p_i, p_{i+1}) \in IB(P)$ that does not contain *x* in its interior let $L(p_i, p_{i+1})$ be the line passing through (p_i, p_{i+1}). Notice that the aperture angle defined over edge (p_i, p_{i+1}) with respect to ab is the aperture angle over the interval $I = [p_i, p_{i+1}] \subset L(p_i, p_{i+1})$ with respect to ab. Note also that $L(p_i, p_{i+1})$ does not intersect ab. Assume without loss of generality that segment ab lies above $L(p_i, p_{i+1})$, that the intersection point t between $L(a, b)$ and $L(p_i, p_{i+1})$ is to the left (in the sense of smaller ordinate) of the interval I, and that a lies between b and t. If the intersection point is to the right of I the argument is symmetric. We assume the configuration has been rotated so that no edge of the polygon is vertical. (refer to Figs. 5a and 5b for illustrations).

Since the edge (p_i, p_{i+1}) is in $IB(P)$, the line $L(p_i, p_{i+1})$ intersects the circle C at two points z_1 and z_2, such that $z_1 \neq z_2$ and both points lie outside I. When traversing the circle C in counterclockwise direction from point a, we define the order as a, z_1, z_2, b. Thus, there are two possible arrangements of points over $L(p_i, p_{i+1})$. One of them is $(t, p_{i+1}, p_i, z_1, z_2)$ which occurs if $[p_i, p_{i+1}]$ is more counter-clockwise of x on $IB(P)$ (refer to Fig. 5.a). The other order is $(t, z_1, z_2, p_{i+1}, p_i)$ which occurs when $[p_i, p_{i+1}]$ is more clockwise of x on $IB(P)$ (refer to Fig. 5.b).

The maximum aperture angle from $L(p_i, p_{i+1})$ with respect to segment ab must occur at a point $y \in [z_1, z_2]$ since any point outside C that is on the interval (t, ∞) of $L(p_i, p_{i+1})$, has a smaller aperture angle than z_1 and z_2, by observation 2.1. Since $[z_1, z_2] \not\subset I$, we have $y \notin I$. Therefore, by lemma 2.1 if the sequence of points on $L(p_i, p_{i+1})$ is $(t, p_{i+1}, p_i, z_1, z_2)$ then the function $\theta(x)$ is strictly increasing over I, and if the sequence is $(t, z_1, z_2, p_{i+1}, p_i)$ then function $\theta(x)$ is strictly decreasing over I. Thus if the maximum aperture angle occurs at a vertex, the lemma holds. If, however, the maximum occurs at a point x in the interior of an edge (p_{k-1}, p_k), we have not yet established that the function is unimodal on that interval. But, in this case the unimodality follows from lemma 2.1. Q.E.D.

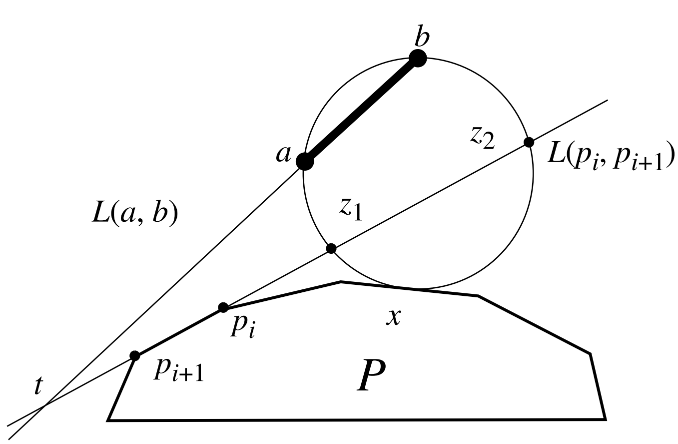

Fig. 5 (a)

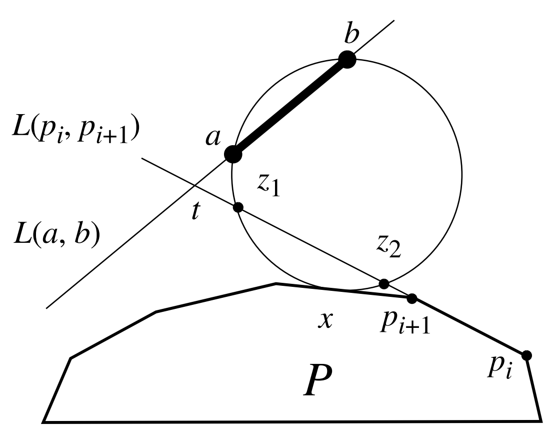

Fig. 5 (b)

An algorithm to find the maximum aperture angle follows directly from the above discussion and is presented below. We assume in this paper that the vertices of the polygons are stored in arrays.

Algorithm Polygon-to-Segment-Max

Input: A convex polygon P with n vertices and a segment ab such that $L(a, b)$ does not intersect P.
Output: A point $x \in P$ for which $\theta(x)$, with respect to ab, is a maximum over P.

Begin

Step 1.- Compute the chain $IB(P)$.

Step 2.- Determine the point x, where the circle C through a and b is tangent to P, by using binary search over $IB(P)$.

Step 3.- Exit with x.

End

Lemma 2.5: Algorithm *Polygon-to-Segment-Max* finds in O(log n) time a point $x \in P$, such that $\theta(x)$ is a maximum with respect to the segment ab.

Proof: Step 1 can be done in O(log n) time using binary search, since the slope of a line segment connecting a point outside a convex polygon to a point that travels along the boundary of the polygon, defines a bimodal function [CD87]. Consider an edge $[p_i, p_{i+1}]$ in $IB(P)$ and the line $L(p_i, p_{i+1})$. Let z be the point on $L(p_i, p_{i+1})$ that realizes the maximum aperture angle for segment ab. From lemma 2.4 it follows that the solution for P lies in $[p_i, p_{i+1}]$ if z lies in $[p_i, p_{i+1}]$, if z lies on $ray(p_{i+1}, p_i)$ beyond p_i then the solution for P lies on the sub-chain of $IB(P)$ clockwise of p_i, and if z lies on $ray(p_i, p_{i+1})$ beyond p_{i+1} then the solution lies on the sub-chain of $IB(P)$ counter-clockwise of p_{i+1}.Therefore we may use binary search on $IB(P)$ to find the solution segment of P where the aperture angle is a maximum. Once this solution segment is identified, the circle through ab and tangent to the solution segment can be found in constant time. Therefore the complexity of step 2 is bounded by O(log n). Q.E.D.

We now turn our attention to the *minimization* version of the Polygon-to-Segment problem. Before presenting a characterization of the solution to this problem, we first define some additional geometric concepts.

Definition: A line L is a *common tangent* of P and ab if: (1) it is tangent to P and ab, and (2) it leaves P and ab in one of the closed halfplanes defined by L.

Let the common tangents of P and ab be tangents at $\{a, p_r\}$ and $\{b, p_s\}$ respectively (see Fig. 6). If the common tangents are colinear with edges of P then let p_r and p_s be the end points of these edges that are furthest from a and b, respectively. We assume that $L(a, b)$ does not intersect $int(P)$, since otherwise the minimum aperture angle is determined by any point the intersection of P and $L(a, b)$. Define the *outer boundary of P with respect to segment ab*, denoted by $OB(P)$, as the intersection of P with the boundary of the convex hull of P union ab. Thus the end points of $OB(P)$ are p_r and p_s. Note that p_r and p_s may coincide.

Lemma 2.6: Any point x in P where the aperture angle reaches the minimum value lies on a vertex of the chain $OB(P)$.

Proof: (by contradiction) Let us assume that x is a point where the minimum is attained and such that it is not contained in $OB(P)$. Let $\theta(x) = ang(a\,x\,b)$ and refer to Fig. 6. Consider the $cone(x)$ that defines the aperture angle $\theta(x)$. The lines $L(x, a)$ and $L(x, b)$ partition the plane into four wedges. Two wedges that share only a point are called *opposite* wedges. The union of two opposite wedges is called a *double wedge*. Let W denote the wedge that does not contain ab but is part of the double wedge that contains ab. By construction, the intersection of $int(W)$ and $OB(P)$ exists. Let y be a point in this intersection and translate $cone(x)$ so that x coincides with y. The new (translated) cone has an angle at y equal to what it had at x and it contains ab in its interior. However, the bounding rays are no longer tangent to ab and can be rotated in the directions of the end points of ab in order to become tangent. Therefore $\theta(y) < \theta(x)$, a contradiction. This establishes that the solution lies on $OB(P)$. That it must also lie on a vertex of $OB(P)$ follows from corollary 2.2. Q.E.D.

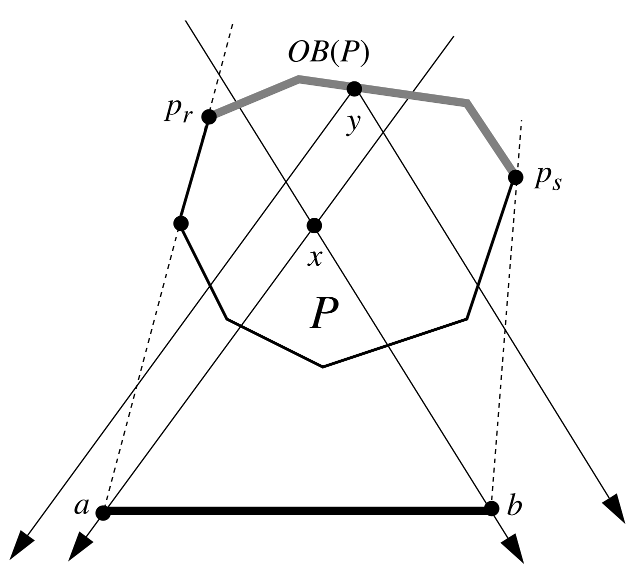

Fig. 6

To find the minimum aperture angle with respect to ab, we evaluate $\theta(x)$ at the end points of every edge in $OB(P)$ and select the global minimum. The algorithm to compute the minimum aperture angle is presented below.

Algorithm Polygon-to-Segment-Min

Input: A segment ab and a convex polygon P that does not intersect ab.
Output: A point $z \in P$ for which $\theta(z)$, with respect to ab, is minimum over P.

Begin

Step 1. Determine the chain $OB(P)$.
Step 2. For every edge e of $OB(P)$ determine the minimum over e.
Step 3. Exit with $z = z_i$ such that $\theta(z_j) \geq \theta(z_i)$ for all $j \neq i$.

End

Lemma 2.7: Algorithm *Polygon-to-Segment-Min* finds in O(n) time a point $z \in P$, such that $\theta(z)$ is minimum with respect to the segment ab.

Proof: Step 1 can be done in O(log n) time using binary search as in lemma 2.5 [CD87]. By corollary 2.2, for every edge e of $OB(P)$ determining the minimum over e takes O(1) time. Thus, the global minimum over P can be found in O(n) time. Q.E.D.

The Line-to-Polygon Problem

We now take a final step towards the general problem and consider a simplification we refer to as the *Line-to-Polygon Problem*, where the object that must be kept in the field of view is a convex polygon Q, but the region where the camera is allowed to roam is a line L.

Problem: Given a convex polygon Q and a line L, find a point $x \in L$ such that the aperture angle $\theta(x)$ is a maximum.

To simplify the notation, assume that no edge of Q is parallel to the line L. Also assume that the polygon and the line do not intersect. Without loss of generality assume L is the x-axis, and let q_h be the vertex of Q with the highest y coordinate and q_l be the vertex with the lowest y coordinate. Thus the boundary of Q is decomposed into two chains: a left chain $Q_a = \{q_h, q_{h+1},..., q_l\}$ and a right chain $Q_b = \{q_l, q_{l+1},..., q_h\}$. We partition L by extending every edge of Q_a until it intersects L at a point a_i and every edge of Q_b until it intersects L at a point b_j. Finally we merge the ordered sets $A = \{a_1, a_2,..., a_{l-h}\}$ and $B = \{b_1, b_2,..., b_{n-l}\}$ (subindex addition is done modulo n) to obtain an ordered set $R = \{r_1, r_2,..., r_n\}$. The partition of L consists of the intervals $I_k = [r_k, r_{k+1}]$ $k = 1, 2,..., n$-1 together with two unbounded intervals $I_0 = (-\infty, r_1]$ and $I_n = [r_n, +\infty)$.

The following lemma provides the link between the *Line-to-Segment problem* and the *Line-to-Polygon problem* by reducing the latter to a family of instances of the former.

Lemma 2.8: For every interval $I_k = [r_k, r_{k+1}]$ in the partition, there are two vertices $q_s \in Q_a$ and $q_t \in Q_b$, that determine a diagonal d_k of Q, such that for every point $x \in I_k$ the aperture angle $\theta(x)$ with respect to Q is given by $ang(q_s\, x\, q_t)$.

Proof: Since q_h and q_l are the highest and lowest points of Q, respectively, then we have that for all $x \in I_0 = (-\infty, r_1]$, $\theta(x)$ is given by $ang(q_h\, x\, q_l)$, (see Fig. 7 (a)).

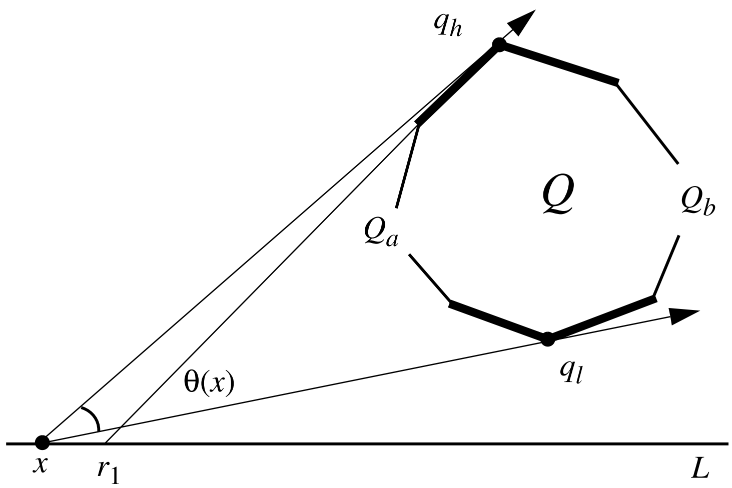

Fig. 7 (a)

Suppose that for interval $I_k = [r_k, r_{k+1}]$, $\theta(x)$ is given by $ang(q_s\, x\, q_t)$ for all $x \in I_k$. Note that if $r_{k+1} = a_s$ (i. e. r_{k+1} is the intersection point of L with the extension of the edge (q_s, q_{s+1}) of Q_a) then for interval I_{k+1}, $\theta(x)$ is given by $ang(q_{s+1}\, x\, q_t)$, for all $x \in I_{k+1}$ (see Fig. 7 (b)).

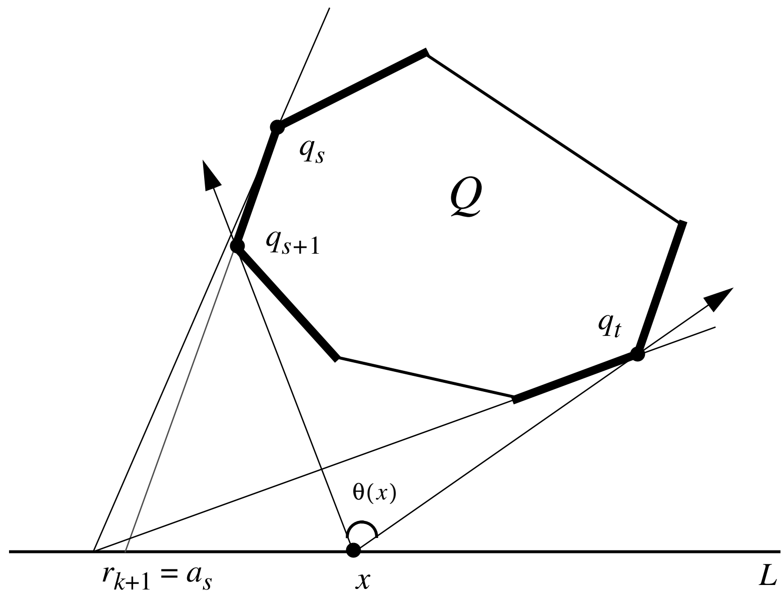

Fig. 7 (b)

If $r_{k+1} = a_t$ (i. e. r_{k+1} is the intersection point of L with the extension of the edge (q_t, q_{t+1}) of Q_b) then for interval I_{k+1}, $\theta(x)$ is given by $ang(q_s\, x\, q_{t+1})$, for all $x \in I_{k+1}$ (see Fig. 7 (c)).

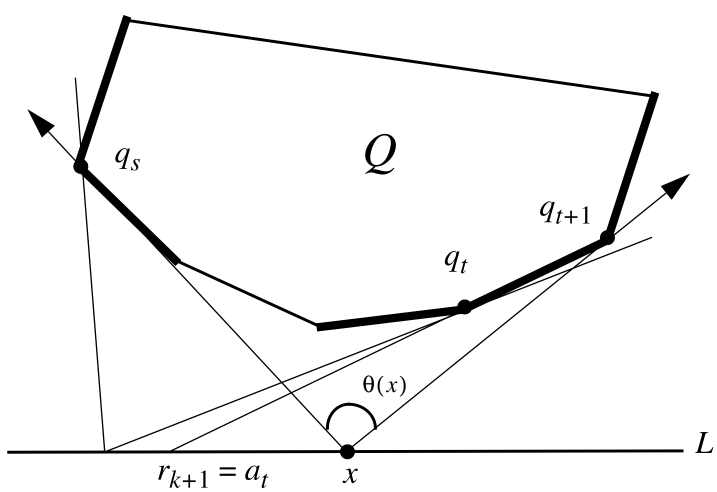

Fig. 7 (c)

Thus by induction the lemma follows. Q.E.D.

As a consequence of lemma 2.8 the aperture angle function $\theta(x)$ with respect to Q is piecewise defined over L. For every interval I_k, the problem is reduced to the Line-to-Segment problem, where the segment d_k is determined by the diagonal of Q spanning the two vertices that define the interval I_k. Therefore, to find the maximum (respectively, minimum), we simply compute candidates for the maximum (respectively, minimum) for every interval and choose, as the global maximum (respectively, minimum), the maximum (respectively, minimum) of all the candidates.

The algorithm to compute the maximum aperture angle is given below. Recall that for every interval $I_k = [r_k, r_{k+1}]$ in the partition described above, there are two vertices $q_s \in Q_a$ and $q_t \in Q_b$ that determine a diagonal d_k of Q, such that for every point $x \in I_k$ the aperture angle $\theta(x)$ with respect to Q is given by $ang(q_s\, x\, q_t)$. To compute the minimum, simply find the minimum in step 3 and change the direction of the inequality in step 4.

Algorithm Line-to-Polygon

Input: A convex polygon Q with n vertices and a line L that does not intersect Q.
Output: A point x in L for which the aperture angle $\theta(x)$, with respect to Q, is maximum.

Begin

Step 1.- Find the partititon of L into intervals $I_0, I_1,..., I_n$.

Step 2.- For every interval I_k find the diagonal d_k such that the aperture angle function with respect to Q and d_k coincide over I_k.

Step 3.- For every interval I_k find $x_k \in I_k$ such that the aperture angle, with respect to d_k is a maximum over I_k.

Step 4.- Exit with x_i is such that $\theta(x_i) \geq \theta(x_j)$ for all $j = 0, 1,..., n$.

End

Theorem 2.1: Algorithm *Line-to-Polygon* finds in $O(n)$ time a point $x \in L$, such that $\theta(x)$ is a maximum with respect to Q.

Proof: Step 1 can be done in $O(n)$ time by first scanning the polygon's edges, extending the edges to rays in the appropriate direction, and intersecting the resulting rays with the line L. This process is then repeated by scanning in the opposite direction. Finally, due to convexity, the two resulting sorted lists of intersection points on L can be merged in $O(n)$ time. By lemma 2.8, step 2 may be performed in $O(n)$ time. To compute each point x_k in step 3 $O(1)$ time suffices by corollary 2.1 and since there are $O(n)$ intervals, step 3 can be done in $O(n)$ time. Thus Algorithm *Line-to-Polygon* takes $O(n)$ time to find a point x in L for which the aperture angle $\theta(x)$, with respect to Q, is a maximum. Q.E.D.

3. The Case of Two Convex Polygons: The Maximization Problem

We now have the tools to solve the general problem where the object that must be kept in the field of view is one convex polygon Q, and the region where the camera is allowed to roam is another convex polygon P. We assume that $P = [p_1, p_2,..., p_m]$ is represented by an array in clockwise order and $Q = [q_1, q_2,..., q_n]$ is represented by an array in counterclockwise order, in order to simplify notation. Let v be the point in P where the viewer (camera) is located. The maximum aperture angle with respect to Q over all locations v in P will be denoted by $\theta_{max}(v)$. Let $OB^c(Q)$ denote the portion of the boundary of Q not containing $OB(Q)$ together with end points q_i and q_j. Note that $OB^c(Q)$ could be the entire boundary of Q (see Fig. 8).

Problem: Given two disjoint convex polygons P and Q in the plane with m and n vertices, respectively, find $\theta_{max}(v)$.

Lemma 3.1: $\theta_{max}(v)$ is realized by a point v on $IB(P)$.

Proof: The proof is similar to that of lemma 2.2.

.

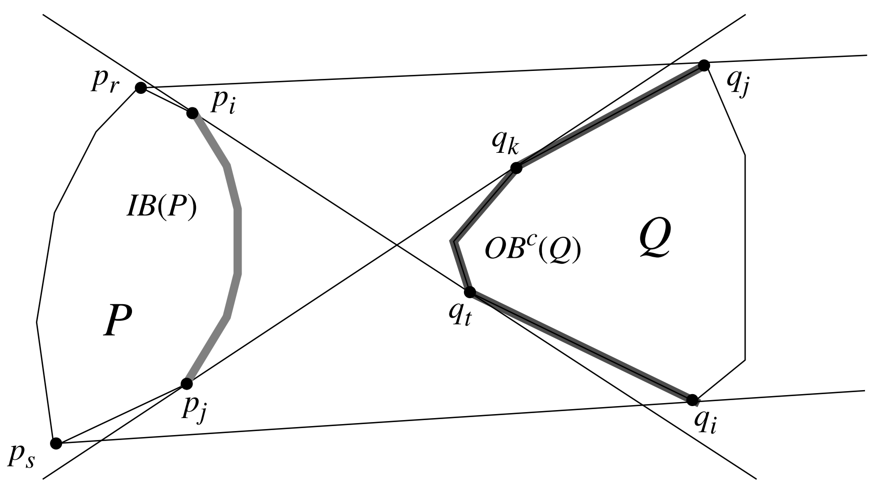

Fig. 8

Given that the maximum aperture angle is reached at a point on *IB(P)*, we define a partition of $IB(P)$, similar to the partition of the line in the *Line-to-Polygon* problem. For every edge e of $OB^c(Q)$ - $IB(Q)$, extend e until it intersects P (refer to Fig. 9). The resulting intersection points determine the desired partition. Notice that the extension of the edges of $IB(Q)$ and $OB(Q)$ do not intersect P and therefore we need only consider the extension of edges in $OB^c(Q)$ - $IB(Q)$. Removing the edges of Q that belong to $IB(Q)$ and $OB(Q)$ divides the boundary of Q into two chains that we denote by Q_a and Q_b. We denote by A the ordered set of intersection points between the extended edges of Q_a and $IB(P)$, and by a_k the intersection of the *k-th* extended edge from Q_a with $IB(P)$. These vertices are illustrated by black circles in Fig. 9. Analogously B is the ordered set of intersection points between the extended edges of Q_b and $IB(P)$, and b_k denotes the intersection of the *k-th* extended edge from Q_b with $IB(P)$. These vertices are illustrated by white circles in Fig. 9. Finally, the original vertices of P are illustrated by grey circles in Fig. 9. Let the partition of the boundary of P obtained by merging the two ordered sets A and B be denoted by $R(P)$ and the resulting merged intersection points by $r_0, r_1, \ldots, r_s$. Every pair of consecutive intersection points r_{k-1} and r_k in the merged set forms a piece of the boundary of P and is denoted by $R_k(P)$. Note that these resulting polygonal chains are convex with respect to Q. Furthermore, for every such convex chain the aperture angle is defined by one and the same diagonal of Q. More precisely, with arguments similar to those of lemma 2.8 we can establish the following result.

Lemma 3.2: For every polygonal chain $R_k(P) \subseteq IB(P)$ in the partition of $bd(P)$, there are two vertices $q_k \in Q_a$ and $q_t \in Q_b$ such that for every point $x \in R_k(P)$, the aperture angle $\theta(x)$ with respect to Q is given by $ang(q_s\, x\, q_t)$.

Therefore lemma 2.5 is applicable to each chain $R_k(P)$, where the diagonal plays the role that the segment *ab* plays in lemma 2.5.

Theorem 3.3: $\theta_{max}(v)$ can be computed in $O(n + m)$ time.

Proof: Let $EIB(P)$ denote the extended inner boundary of P with respect to Q, obtained by inserting dummy vertices in $IB(P)$ where the extended edges of Q intersect $IB(P)$. The polygonal chain $EIB(P)$ is convex with respect to Q and contains $O(n + m)$ edges. For each such edge we find the vertices of Q that admit tangent rays to Q from any point on the edge. These vertices yield a candidate diagonal of Q for each such edge in question. We then compute a candidate maximal aperture angle with respect to Q for that edge by computing the maximum aperture angle for the candidate diagonal. Finally, we select the candidate with a maximum value as $\theta_{max}(v)$. The correctness of this procedure follows from corollary 2.1 and lemmas 3.1 and 3.2.

Consider now the complexity. Using the rotating calipers [To83], we may find the common and separating tangent points of support between P and Q in $O(n + m)$ time. Alternately, we may use the algorithm of Rohnert [Ro86] and accomplish the same task in $O(\log n + \log m)$ time if desired. Therefore the chains $IB(P)$ and the Q_a and Q_b sub-chains of $OB^c(Q)$ may be found within the same time complexity. The first intersection point a_1 that the first extended segment of Q_a makes with $IB(P)$ may be found in $O(\log m)$ time using the algorithm of Chazelle and Dobkin [CD87]. Due to convexity each subsequent intersection point $a_2, a_3 \ldots$ can

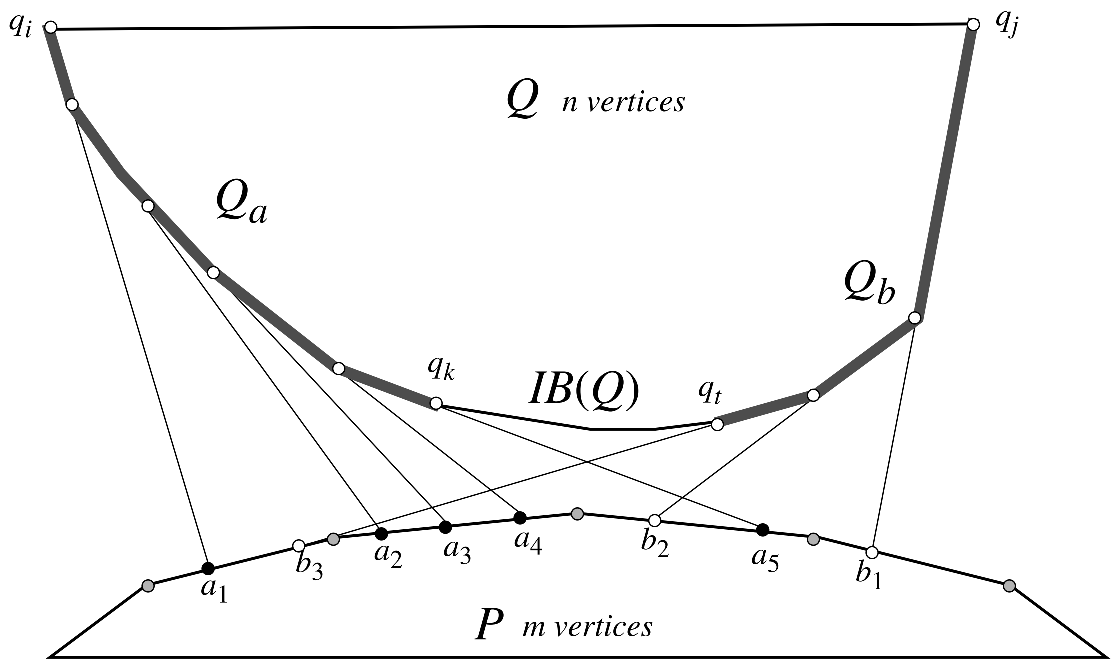

Fig. 9 Illustrating the partition of the boundary of P into regions (edges) where the aperture angle is determined by a single diagonal of Q.

be found by advancing either one edge on $IB(P)$ or one extended edge of Q_a, whichever comes first. Therefore with this alternating traversal of the edges of $IB(P)$ and Q_a the set A of all the intersection points generated by Q_a can be found and inserted in $IB(P)$ in $O(n + m)$ time. Subsequently, in the same way the set B of all the intersection points generated by Q_b on $IB(P)$ can be found and inserted in $O(n + m)$ time. Therefore the extended chain $EIB(P)$ can be found in $O(n + m)$ time. Furthermore, as we advance along edges of P to find the next intersection point of an extended edge of Q_a (similarly for Q_b) we insert pointers from these edges of $EIB(P)$ to their tangent vertices of Q. Therefore, for each edge of $EIB(P)$ we can subsequently find the candidate diagonal of Q that determines its aperture angle in constant time per candidate. Finally for each such diagonal-edge pair candidate we may compute a candidate maximal aperture angle also in constant time per candidate. Therefore the overall procedure takes $O(n + m)$ time. Q.E.D.

In the above $O(n + m)$ time procedure the chain $EIB(P)$ is obtained by merging the two ordered sets A and B (that jointly form $R(P)$) with the chain $IB(P)$ in $O(n + m)$ time, and subsequently computing $O(n + m)$ candidates for $\theta_{max}(v)$, each in constant time. We may obtain a different upper bound on the problem by computing only $O(n)$ candidates for $\theta_{max}(v)$, each in time $O(\log m)$, as we now show.

Theorem 3.4: $\theta_{max}(v)$ can be computed in O(n log m) time.

Proof: Consider the partition $R(P)$ obtained by merging the two ordered sets A and B. Every pair of consecutive intersection points r_{k-1} and r_k in $R(P)$ forms a convex polygonal chain $R_k(P)$ which is a subset of $IB(P)$. For each such chain its maximum aperture angle is determined by a single diagonal of Q. Therefore we may use binary search to find a candidate aperture angle for $R_k(P)$ for each k. The correctness of this procedure follows from corollary 2.1 and lemmas 2.1-2.4.

Consider now the complexity. We may use the algorithm of Chazelle and Dobkin [CD87] to determine all the intersection points (the a_k's and b_k's) that form the sets A and B, respectively. Since there are at most n intersections and each one is found in O(log m) time, the sets A and B are found in O(n log m) time. We now show how to merge A and B in O(n) time.

Let the edges of $IB(P)$ be numbered 1, 2,...,k in clockwise order. When computing each a_k and b_k, we associate with the intersection point a pointer to the label of the edge of $IB(P)$ on which the intersection point occurs. For example, if a_1 occurs on edge 5 then we store edge 5 with a_1 and so on. Now we can merge sets A and B in O(n) time since the sorted order of the intersection points (the a_k's and b_k's) is known and the labels of the edges on which these intersections occur is known. Thus, we avoid looking at the whole chain $IB(P)$ and only concentrate on the edges which contain intersection points.

Finally, computing a candidate aperture angle for $R_k(P)$ for each k takes O(log m) time for the binary search. Since there are at most O(n) candidates to be computed, finding the maximum takes O(n log m) time. Q.E.D.

The above algorithm can in fact be improved to O(n + n log (m/n)) time. The improvement is based on a method of finding the intersection points (the a_k's and b_k's) that form the sets A and B in a more efficient manner. We outline this method below.

Theorem 3.5: $\theta_{max}(v)$ can be computed in O(n + n log(m/n)) time.

Proof: We first show how to find all intersection points a_i from the set A in O(n + n log(m/n)) time. The points from set B can be found in the same way. We number the edges of $IB(P)$ = 1, 2,..., k in clockwise order.

In O(log m) time using the algorithm of Chazelle and Dobkin [CD87] we find the edge j containing a_1. Since the chain A is convex, the a_i's occur in sorted order on the chain $IB(P)$. We find the a_i's in order of their occurrence.

Given that a_i occurs on edge j, we show how to find a_{i+1}. Let w_{i+1} be the number of edges to the right of the edge j at which a_{i+1} occurs, i.e. a_{i+1} is on edge $j + w_{i+1}$. The strategy used to find a_{i+1} is quite simple. We first check to see if a_{i+1} occurs on edge j, then we check edge j+1, edge j+2, edge j+4 and so on. In short, we verify edges $j+2^t$ (t = 0,...) until we find the first edge $j+2^s$ which either contains, or is to the right of, a_{i+1}. This implies that a_{i+1} occurs on one of the edges in the chain from $j+2^{s-1}$ to $j+2^s$. If $s > 1$ then we apply binary search on this chain to find edge $j+w_{i+1}$.

Let us analyze the complexity of the search procedure. In the first step, we find the edge $j+2^s$. If a_{i+1} occurs on edges j, j+1, or j+2, we expend a constant amount of time to find it. If it occurs beyond edge j+2, then we expend O(log s) time. Therefore, this step takes time max{O(1), O(log s)}.

If $s > 1$, then in the second step, we apply a binary search on the chain from $j+2^{s-1}$ to $j+2^s$. The binary search takes time O(log s). Note that for $s > 1$, we have that $w_{i+1} \le 2^s \le 2w_{i+1}$. Therefore, the total time used to find a_{i+1} is max{O(1), O(log s)} which in turn equals max{O(1), O(log w_{i+1})}.

We now analyze the time $T_A(n,m)$ taken to compute all the intersection points a_i from the set A. First, finding a_1 takes O(log m) time. To find every subsequent a_i takes time equal to max{O(1), O(log w_i)}.

Note that

$$\sum_{i=2}^{n} w_i \le m \qquad (1)$$

Therefore, the total time $T_A(n,m)$ equals:

$$O(\log m + \sum_{i=2}^{n} max\{1, \log w_i\})$$

$$\le O(\log m + \sum_{w_i < 2,\ i = 2 \dots n} 1 + \sum_{w_i \ge 2,\ i = 2 \dots n} \log w_i)$$

$$\le O(\log m + n + \sum_{w_i \ge 2,\ i = 2 \dots n} \log w_i)$$

Using Jensen's inequality and equation (1) above we have:

$$T_A(n, m) \le O(\log m + n + n\log(m/n))$$

$$= O(n + n\log(m/n))$$

Thus the total time taken to compute all the a_i's is O(n + n log(m/n)). In the same way, all the intersection points b_j from the set B can be computed in O(n + n log(m/n)) time. Finally, the two sets can be merged in O(n) time as shown in the proof of theorem 3.4.

Merging the two ordered sets creates a partition $R(P)$ of chain $IB(P)$. Every pair of consecutive intersection points r_{k-1} and r_k in $R(P)$ forms a convex polygonal chain $R_k(P)$ which is a subset of $IB(P)$. If the chain has less then 3 edges, the solution can be found in constant time. Otherwise a binary search can be used to find a candidate aperture angle for each $R_k(P)$. Finally, the maximum of all these candidates is chosen as the maximum aperture angle. The correctness of this procedure follows from corollary 2.1 and lemmas 2.1-2.4.

We now analyze the complexity of computing the maximum aperture angle. Let c_k represent the number of edges in chain $R_k(P)$.

Note that:

$$\sum_k c_k \le m \qquad (2)$$

Furthermore, the total time taken to find the maximum aperture angle equals:

$$O\left(n + n\log(m/n) + \sum_k max\{1, \log c_k\}\right)$$

which, by Jensen's inequality and equation (2) is no greater than:

$$O(n + n\log(m/n))$$

Q.E.D.

4. The Case of Two Convex Polygons: The Minimization Problem

We assume as before that $P = [p_1, p_2,..., p_m]$ is represented by an array in clockwise order and $Q = [q_1, q_2,..., q_n]$ is represented by an array in counterclockwise order. Let v be the point in P where the viewer (camera) is located. The minimum aperture angle with respect to Q over all locations v in P is denoted by $\theta_{min}(v)$.

Problem: Given two disjoint convex polygons P and Q in the plane with m and n vertices, respectively, find $\theta_{min}(v)$.

Before we characterize the solution points in P for $\theta_{min}(v)$ we recall the characterization for the *Polygon-to-Segment* minimization problem presented in lemma 2.6. In that problem, because the solution is trivially zero when the line through the segment that is viewed intersects the polygon P, it was assumed that the line does not intersect P. Because of this assumption the points in P where the aperture angle reaches a minimum lie on vertices of $OB(P)$. On the other hand, in the general problem considered here this characterization is no longer valid. It suffices to consider a configuration such as that illustrated in Fig. 10 where P is thin and wide with $OB(P)$ a single segment $[p_r, p_s]$ and Q is thin and tall "pointing" towards the central region of $[p_r, p_s]$. In such an example $\theta_{min}(v)$ is realized by a point v in the interior of $[p_r, p_s]$ and not by either p_r or p_s. Nevertheless, we now show that in general the solution can only occur at a finite number of locations in $OB(P)$, and that these may be searched efficiently.

Lemma 4.1: $\theta_{min}(v)$ is realized by a point on $OB(P)$ that is either a vertex of $OB(P)$ or an intersection point of $OB(P)$ with a line that is colinear with an edge of Q.

Proof: The two separating tangents of P and Q partition the plane into four wedges. Let $W(P)$ denote the wedge that contains P. Therefore the solution must lie in $W(P)$. Now partition $W(P)$ into a convex subdivision as follows. For each vertex q_i in Q_a (except the last vertex of Q_a) construct the infinite half ray in the direction of q_{i+1} and denote it by $ray(q_i, q_{i+1})$. Similarly, for each vertex q_j in chain Q_b (except the last vertex of Q_b) construct $ray(q_{j+1}, q_j)$. Finally, construct rays from the first and last vertices of Q_a and Q_b along the common and separating tangents of P and Q and in the direction of P. This arrangement of rays induces a subdivision of $W(P)$ and hence of P. Denote the resulting subdivision of P by $F(P)$. Each face f_i of $F(P)$ is a convex polygon with the property that the aperture angle of any point v in f_i is determined by one and the same diagonal of Q, say d_i. Therefore, for each face f_i of $F(P)$ we have an instance of the *Polygon-to-Segment* problem and by lemma 2.6 the solution to subproblem f_i is determined by $OB(f_i)$ with respect to d_i. If the solution to a subproblem f_i does not lie in $OB(P)$ then the same argument used in the proof of lemma 2.6 shows that a smaller aperture angle exists in $OB(P)$.

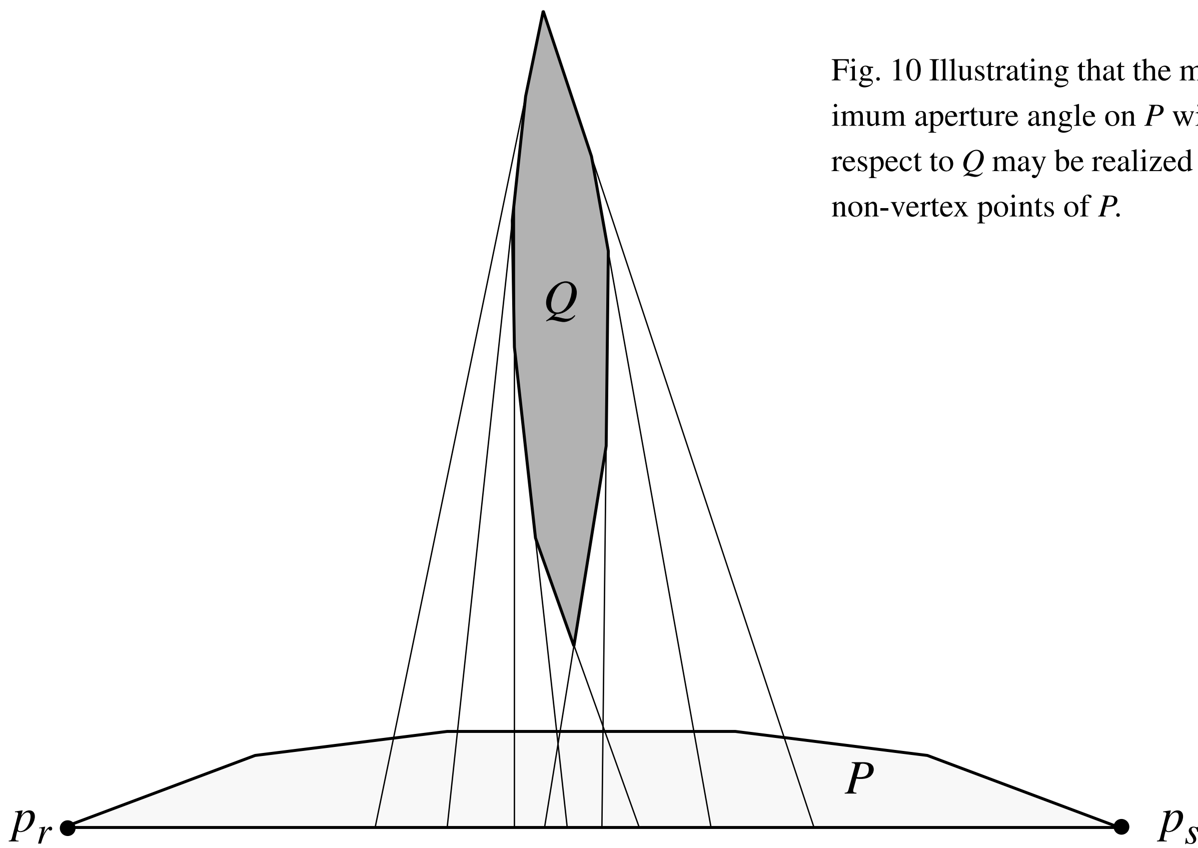

Fig. 10 Illustrating that the minimum aperture angle on P with respect to Q may be realized by non-vertex points of P.

Therefore $\theta_{min}(v)$ is realized by a vertex of a face f_i of $F(P)$ that lies on $OB(P)$. But these vertices are precisely either the vertices of P or the intersection points that the rays extended from Q_a and Q_b make with $OB(P)$. Q.E.D.

Theorem 4.2: $\theta_{min}(v)$ can be computed in $O(n + m)$ time.

Proof: As in the proof of theorem 3.3, we compute an extended outer chain $EOB(P)$ by inserting dummy vertices in $OB(P)$ where the rays of the extended edges from Q_a and Q_b meet $OB(P)$. For each edge in $EOB(P)$ the aperture angle is determined by a single diagonal of Q. From corollary 2.2 it follows that a candidate solution is determined for each edge of $EOB(P)$ by one of its end points. The correctness of this procedure is immediate from lemma 4.1. The computational tools are the same as those used in the proof of theorem 3.3 and $O(n + m)$ time suffices. Q.E.D.

5. Lower Bounds

In the previous two sections we described algorithms for computing $\theta_{max}(v)$ and $\theta_{min}(v)$. We presented three algorithms for computing $\theta_{max}(v)$. Their running time complexities are $O(n + m)$, $O(n \log m)$ and $O(n + n \log(m/n))$. We also gave an algorithm for computing $\theta_{min}(v)$ in $O(n + m)$ time.

In this section we show that the complexity of computing $\theta_{min}(v)$ is $\Omega(\max(m, n))$. We also show a time complexity of $\Omega(\max(n, n \log(m/n)))$ for computing $\theta_{max}(v)$. This proves the optimality of the algorithms to compute $\theta_{max}(v)$ and $\theta_{min}(v)$. We begin by describing a construction that proves $\Omega(m)$ is a lower bound for computing $\theta_{min}(v)$. Then we describe another construction that shows $\Omega(n)$ is a bound for $\theta_{min}(v)$ and which also affords a simple modification of it to establish the same bound for $\theta_{max}(v)$. Finally, when m is $\omega(n)$, we establish an $\Omega(n \log(m/n))$ lower bound. Our lower bounds rely on the fact that the polygons are given in the form of linear arrays, a very natural representation.

Theorem 5.1: The complexity of computing $\theta_{min}(v)$ is $\Omega(m)$.

Proof: We create two convex polygons, the vertices of which lie on the unit circle C centered at the origin. For P we choose vertices on the lower semi-circle of C, whereas Q's vertices are located in the upper semi-circle (see Fig. 11). These polygons have the property that every

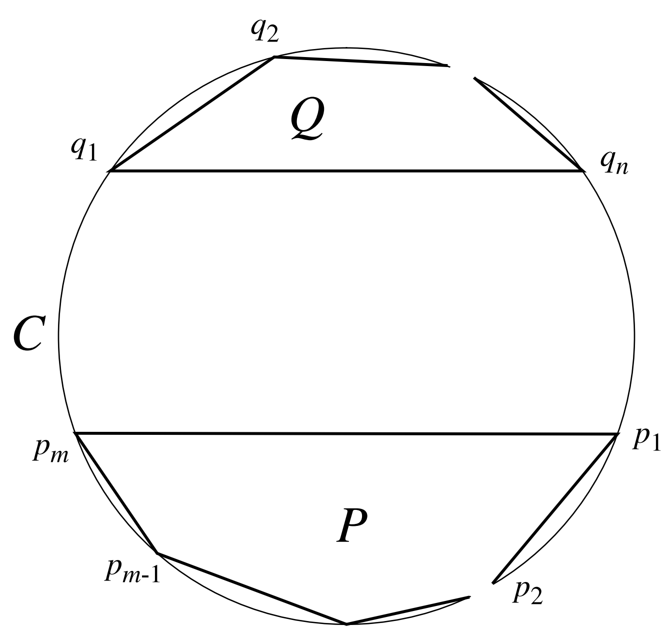

Fig. 11 Illustrating the $\Omega(m)$ lower bound on $\theta_{min}(v)$.

edge of each of the two polygons can be extended by an arbitrarily large distance without intersecting the interior of any other edge in either polygon. Therefore polygon Q has the appearance of a line segment to a viewer in P. In particular, Q behaves as if it were the edge $[q_n, q_1]$. Therefore, by lemma 2.6, $\theta_{min}(v)$ must be realized by a vertex of P. Furthermore, note that since P's vertices are on the circle C and edge $[q_n, q_1]$ is a chord of the same circle, it follows that the aperture angle at each vertex of P is equal. If any algorithm does not inspect a vertex p_i, then an adversary can move it outward and make the smallest angle occur at p_i. Q.E.D.

We turn now to the construction for the $\Omega(n)$ bounds. First we construct a polygon Q of n vertices in the first quadrant of $\boldsymbol{R}^2$ in such a way that the aperture angle function $\theta(x)$ contains $\Omega(n)$ local maxima. The general idea may be likened to cutting a convex polygon from a piece of paper with an office paper cutter. With such a cutter one may slide the paper against a supporting border in a direction orthogonal to the cutting blade, then lower the blade at the desired position. We will fix the paper and move the cutter. In particular, we will rotate the cutter frame and translate the blade before each cut. Assume that our original piece of paper consists of the first quadrant of $\boldsymbol{R}^2$ and refer to Fig. 12. Our paper cutter is anchored at the point (0,1) about which it is allowed to rotate. Once a position of the cutter is fixed, the infinite blade may be translated as far from (0,1) as desired. Successively we pick points on the x-axis increasingly far from (0,1), lower the blade and discard the paper below the cut. We also make one cut at $y = 1$. This leaves a convex but still unbounded polygon (the shaded region in Fig. 12). To fix this we make one final cut along a line through (0,1) and of sufficiently large but finite slope.

Let us consider the above idea in more detail. For simplicity assume that Q has $n + 1$ vertices labelled $q_0, q_1, \ldots, q_n$ in counter-clockwise order. We begin by locating the last and first vertices of Q at $q_n = (0,1)$ and $q_0 = (2,1)$, respectively, and constructing the circle C_0 of unit radius centered at (1,1) (see Fig. 13). Let $x = T_0$ be the point at which C_0 is tangent to the x-axis. Let r_0 denote the ray starting at T_0 in the direction of q_0. The next edge of Q, namely $[q_0, q_1]$, is chosen to lie on r_0. To know where on r_0 to locate q_1, pick any point on the x-axis some finite distance to the right of T_0 and call it T_1. The line segment $[q_n, T_1]$ must intersect C_0 at a point z_1 in the interior of the arc of C_0 (measured in a counter-clockwise direction) given by $arc(T_0, q_0)$, with the property that z_1 is smaller than T_1. Next construct the ray r_1 starting at T_1 in an upwards direction parallel to $[z_1, q_0]$. Since the line through $[z_1, q_0]$ intersects r_0 at q_0, and z_1 lies to the left of T_1, r_1 must intersect r_0 at some point to the right and above q_0. We locate vertex q_1 at this intersection point and call $[q_n, q_1]$ the diagonal d_1 of Q. To finish the procedure that is to be iterated we construct a circle C_1 with diameter d_1 that passes through the four points $\{q_n, T_0, T_1, q_1\}$. That such a circle exists follows

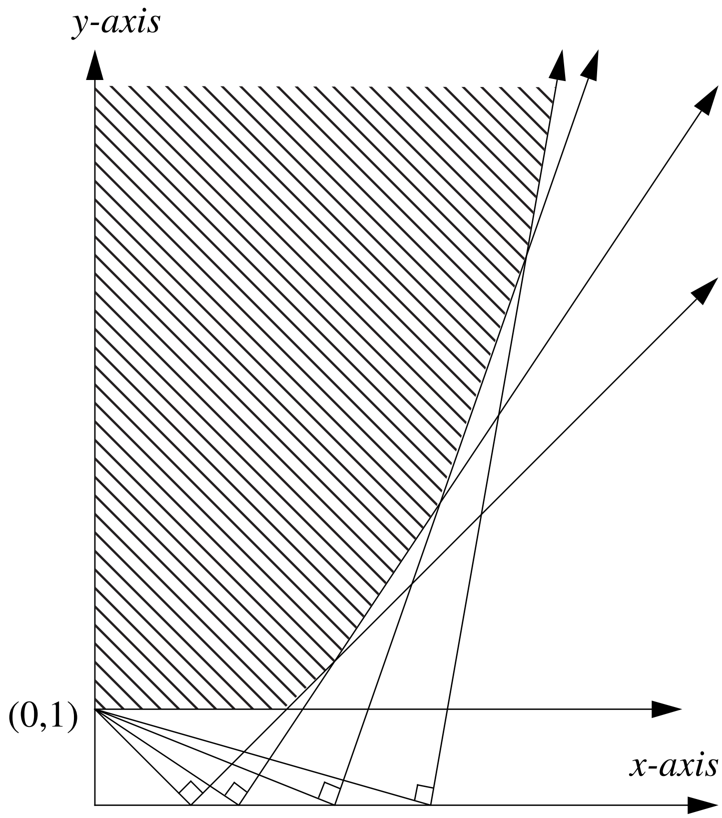

Fig. 12 Illustrating the construction of polygon Q.

from the fact that both angles $ang(q_n, T_0, q_1)$ and $ang(q_n, T_1, q_1)$ are $\pi/2$. At the next iteration q_2 is located on r_1 and above q_1 thus preserving convexity. When the $(n + 1)st$ vertex is located it is connected to q_n thus completing Q.

To summarize the inductive step assume we are given the above construction at step k. In other words, we constructed the tangent point T_k, the ray r_k, the vertex q_k and the circle C_k with diameter $d_k = [q_n, q_k]$ and we want to insert vertex q_{k+1}. Accordingly, we pick a point T_{k+1} to the right of T_k on the x-axis. We find the intersection point z_{k+1} of $[q_n, T_k]$ with the circle C_k. That such an intersection point exists with the required property that z_{k+1} be left of T_{k+1} follows from the fact that circle C_k intersects the x-axis at both T_k and T_{k-1} and therefore the $arc(T_k, q_k)$ lies above the x-axis. Next we construct ray r_{k+1} emanating at T_{k+1} and parallel to $[z_k, q_k]$ which creates the desired vertex q_{k+1} at its intersection with r_k at a point above and to the right of q_k and above ray r_{k-1}, thus maintaining the convexity of Q.

We will now use Q to establish our first lower bounds.

Theorem 5.2: The complexity of computing $\theta_{min}(v)$ is $\Omega(n)$.

Proof: We construct P to lie within the strip determined by $0 \leq y \leq 1/2$ such that one of its edges

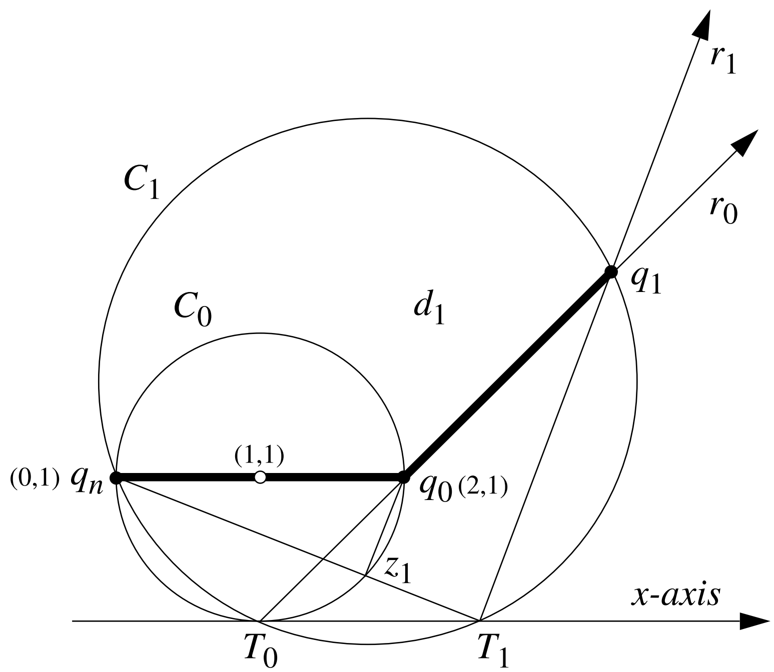

Fig. 13 Illustrating the $\Omega(n)$ bounds on $\theta_{max}(v)$ and $\theta_{min}(v)$.

belonging to $OB(P)$ is flush with the x-axis and contains all the tangent points T_0, T_1, T_2,..., $T_{n\text{-}1}$ obtained from the previous construction of Q. Now consider an interval $[T_{i\text{-}1}, T_i]$ on this edge of P. Since $\theta(x)$ is upwards unimodal in this interval it follows from lemma 2.6 that its minimum value is determined by one of its end points. Therefore $\theta_{min}(v)$ is determined by one of the tangent points T_i. Recall that $\theta(T_i) = \pi/2$ for $i = 0, 1, 2,..., n\text{-}1$. Therefore, if any algorithm does not inspect a diagonal d_i, then an adversary can modify this diagonal so that there exists a point on the x-axis that yields a global minimum less than $\pi/2$. This modification may be accomplished by picking an arbitrary edge $[q_{i\text{-}1}, q_i]$ of Q and increasing its slope by a suitably small but positive amount without changing the position of $q_{i\text{-}1}$, thus creating a small open interval on the x-axis which lies in between and outside both circles $C_{i\text{-}1}$ and C_i in which $\theta(x) < \pi/2$ and where the global minimum is located. Q.E.D.

Theorem 5.3: The complexity of computing $\theta_{max}(v)$ is $\Omega(n)$.

Proof: We construct polygon P such that no part of it lies above the x-axis and such that one of its edges belonging to $IB(P)$ lies flush with the x-axis and contains all the tangent points T_0, T_1, T_2,..., $T_{n\text{-}1}$ obtained from the original construction of Q. Recall that the aperture angles at all tangent points T_i are each $\pi/2$. Now consider the function $\theta(x)$ in the range of some interval $[T_{i\text{-}1}, T_i]$. Since throughout this interval, Q behaves as the diagonal d_i and $[T_{i\text{-}1}, T_i]$ is also a chord of C_i that is not intersected by d_i, it follows from lemma 2.1 that $\theta(x)$ is upwards unimodal in this range and therefore contains a local maximum with a value greater than $\pi/2$. The exact value of the local maximum in the interval $[T_{i-1}, T_i]$ is determined by the distance between T_{i-1} and T_i which is also the relative length of the chord $[T_{i-1}, T_i]$ of the circle C_i. We can select every T_i after T_0 so that the local maximum for every interval is $\pi/2 + \varepsilon$, where ε is a fixed small positive number. If any algorithm does not inspect diagonal d_i then an adversary can move vertex q_i further out along r_{i-1} and make a local maximum angle greater than $\pi/2 + \varepsilon$ (i.e., the global maximum) for some x in interval $[T_{i-1}, T_i]$. Q.E.D.

Note that for the $\Omega(n)$ lower bound of the maximum aperture angle problem no assumptions are made on the size of polygon P. When m is $O(n)$ this proves that our algorithms with complexities $O(n + n \log(m/n))$ and $O(n + m)$ are optimal. However when m is $\omega(n)$ this lower bound no longer proves optimality. We will use a similar but more complicated construction that proves an $\Omega(n + n \log(m/n))$ lower bound for the $\theta_{max}(v)$ problem when m is $\omega(n)$.

First we choose a suitable pair curves on which the m vertices of P and n vertices of Q will lie. Then we pick the vertices of Q so that there are n local maxima of angle at most $\pi/2$. Finally an adversary reveals the vertices of P in response to queries in such a way that $\log(m/n)$ queries must be asked to determine the true angle of each local maximum.

The vertices of P are chosen on the hyperbola $y = -1/x$ for $3/2 \leq x \leq n + 1/2$ and polygon Q has vertices $(0,0)$, $q_i = ((3i^4-1) / 2i^3, (i^4-3) / 2i)$ for integers $2 \leq i \leq n$, and $(0, n^3)$. These q_i are chosen so that the following properties hold:

(1) the chain from q_2 to q_n is convex,

(2) the slope of the line containing q_i and q_{i+1} is positive for $2 \leq i \leq n$,

(3) the circle with diameter from the origin to q_i is tangent to $y = -1/x$ at a point on $x = i$,

(4) the ray from the point $(i, -1/i)$ through point q_i intersects Q only at point q_i.

To verify (1) and (2), note that the slope of the line containing q_i and q_{i+1} is greater than i^2 but less than $2i^2$. Since this slope is positive and increasing with i, (1) and (2) hold. To verify equation (3), observe that the equation of the line that bisects the origin and the point $(i, -1/i)$ is given by $(y + 1/2i) / (x - i/2) = i^2$. The line normal to $y = -1/x$ at $x = i$ is $(y + 1/i) / (x - i) = -i^2$. These two lines intersect at the circle center $((3i^4-1) / 4i^3, (i^4-3) / 4i)$ and the point q_i is double this vector. As a result, if $(i, -1/i)$ is a vertex of P, then the aperture angle, which is defined by the origin and q_i, has a local maximum of $\pi/2$ at that vertex. Otherwise, aperture angles using q_i are less than $\pi/2$, in accordance with observation 2.1. Finally, (4) holds since the chain from q_2 to q_n is convex and the slope of the line containing $(i, -1/i)$ and q_i is less than the slope of the line containing q_i and $q_{i+1.}$

With this construction we can now prove the following lower bound by an adversary argument.

Theorem 5.4: The complexity of computing $\theta_{max}(v)$ is $\Omega(n \log(m/n))$ when m is $\omega(n)$.

Proof: Initially, the algorithm knows the polygon Q, as described above, and knows that the vertices of P have x coordinates $3/2 \le x_{2m/n} < x_{2m/n+1} <...< x_m \le n+1/2$ and lie on the curve $y = -1/x$ (The rather strange looking subscripts are chosen so as to make later index calculations easier.) The algorithm discovers the exact point $(x_i\ -1/x_i)$ by a query to an adversary. Since knowing the x coordinates is sufficient, we focus on these. We will show that $\Omega(n \log(m/n))$ queries are necessary. The previous section showed that $O(n \log(m/n))$ were sufficient.

The adversary begins by fixing every (m/n)th point on the curve: $x_{(m/n)j} = j- 1/2$, for integers $2 \le j \le n$. The chain between two consecutive "fixed" points is said to be *floating* since the x-coordinates of the points in that chain are not yet fixed. Initially, there are n-1 floating chains and each chain contains m/n-1 points whose x-coordinates are not fixed. We will say that a floating chain is *fixed* when all the points in that chain are *fixed*. We will show that the adversary can ensure that $\Omega(\log(m/n))$ queries are asked before all points in a floating chain are fixed. Furthermore, only one point will have an integral x-coordinate and thus achieve the maximum aperture angle by the construction of Q. The adversary will ensure that this point is fixed only after n-2 chains are fixed. Therefore, $\Omega(n \log(m/n))$ queries must be made before the x-coordinate of the point with the maximum aperture angle is fixed.

On each floating chain, space is reserved for the $((m/n)j + k)$th point, for $1 \le k < m/n$, at both $j\text{-}1/2 + k/(4m/n)$ and $j+ 1/4 + k/(4m/n)$. The adversary responds to a query for x_i as follows: if x_i is fixed, it reports $(x_i, -1/x_i)$. Otherwise x_i is in a floating chain and $i = ((m/n)j + k)$ for some integer $1 \le k < m/n$. If this is the last floating chain, then the adversary fixes $x_i = j$ and fixes all other floating points in that chain to their reserved spots before or after x_i as needed to preserve their order. If x_i is not in the last floating chain, then the adversary fixes either all points of the floating chain before and including x_i to their reserved spots less than j or all points including x_i and after to their spots greater than j, which ever causes fewer points to be fixed. These actions force all but one floating chain to be fixed before the algorithm discovers which point is fixed to $(j,-1/j)$ for some integer j and, thereby, finds out which aperture angle achieves the maximum value of $\pi/2$. Since the adversary ensures that $\Omega(\log(m/n))$ steps are required to fix each chain, this gives the bound of $\Omega(n \log(m/n))$ queries. Q.E.D.

Notice that all computations can be performed in rational arithmetic with numerators and denominators bounded by small polynomials in n and m. Thus, the adversary can operate within the standard unit-cost RAM model of computation with word length logarithmic in $n + m$.

6. Concluding Remarks

In this paper we considered the problem of computing the aperture angle of a camera that is allowed to travel in a convex region in the plane and is required to maintain some other convex region within its field of view at all times. We presented an $O(n + m)$ time algorithm for computing the minimum aperture angle with respect to a convex polygon Q when x is allowed to vary in P. We also presented algorithms with complexities $O(n \log m)$, $O(n + n \log (m/n))$ and $O(n + m)$ for computing the maximum aperture angle. Finally, we established an $\Omega(n + n \log (m/n))$ time lower bound for the maximization problem and an $\Omega(m + n)$ lower bound for the minimization problem thereby proving the optimality of our algorithms.

7. Acknowledgments

The authors would like to thank David Epstein and Tom Shermer for fruitful discussions on this topic and two anonymous referees for numerous valuable suggestions for improving the presentation of the paper.

8. References

[ABP93] B. Asberg, G. Blanco, P. Bose, J. Garcia-Lopez, M. Overmars, G. T. Toussaint, G. Wilfong and B. Zhu, "Feasibility of design in stereolithography," *Proc. 13th Conf. Foundations of Software Technology and Theoretical Computer Science*, Bombay, December 15-17, 1993, Springer-Verlag LNCS vol. 761, pp. 228-237.

[ABD93] Avis, D., Beresford-Smith, B., Devroye, L., ElGindy, H., Guevremont, E., Hurtado, F. and Zhu, B., "Unoriented θ-maxima in the plane," *Proc. 16th Australian Conference in Computer Science,* Brisbane, 1993.

[BGL93] Bose, P., Guibas, L., Lubiw, A., Overmars, M., Souvaine, D. and Urrutia, J., "The floodlight problem," *Fifth Canadian Conference on Computational Geometry,* Waterloo, Canada, August, 1993. pp. 399-405.

[CD87] Chazelle, B. and Dobkin, D. P., "Intersection of convex objects in two and three dimensions," *Journal of the ACM*, vol. 34, No. 1, January 1987, pp. 1-27.

[CDGP] Colly, J., Devy, M., Ghallab, M., and Pampagnin, L.-H., "Sensory control for 3D environment modeling," *IARP Workshop on Multisensor Fusion and Environment Modeling,* Tolouse, 16-17 October, 1989.

[Co88] Cowan, C. K., “Model-based synthesis of sensor location,” *Proc. IEEE International Conference on Robotics and Automation,* Philadelphia, April 1988, pp. 900-905.

[CR41] Courant, R. and Robbins, H., *What is Mathematics?* Oxford University Press, 1941.

[CW92] Chen, L.-L. and Woo, T. C., “Computational geometry on the sphere with application to automated machining,” *Transactions of the ASME,* vol. 114, June 1992, pp. 288-295.

[DT93] Devroye, L. and Toussaint, G. T., “Convex hulls for random lines,” *Journal of Algorithms* **14**, 1993, pp. 381-394.

[Ni81] Niven, I., *Maxima and Minima Without Calculus*, The Mathematical Association of America, 1981.

[O’R87] O’Rourke J., *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987.

[PS88] Preparata, F. P., Shamos, M.I., *Computational Geometry An Introduction,* Springer-Verlag (1988), pp. 119-128.

[Ro86] Rohnert, H., “Shortest paths in the plane with convex polygonal obstacles,” *Information Processing Letters*, vol. 23, No. 9, 1986, pp. 71-76.

[Sc60] Schwartz, A., *Analytic Geometry and Calculus*, Holt, Rinehart & Winston, 1960.

[Sh92] Shermer, T., “Recent results in art galleries,” *Proceedings of the IEEE*, vol. 80, No. 9, September 1992, pp. 1384-1399.

[SR90] Spyridy, A. J. and Requicha, A. A. G., “Accessibility analysis for the automatic inspection of parts,” *Proc. IEEE Int. Conf. Robotics and Automation*, Cincinnati, OH, USA, May 13-18, 1990, pp. 1284-1289.

[Te89] Teichman, M., *Wedge Placement Optimization Problems*, M.Sc. thesis, School of Computer Science. McGill University,1989.

[To83] Toussaint G. T., Solving geometric problems with the “rotating callipers”, *Proc. of IEEE MELECON*, Athens, Grece, May 1983.

[TWG92]Tang, K., Woo, T. and Gan, J., “Maximum intersection of spherical polygons and workpiece orientation for 4- and 5-axis machining,” *Journal of Mechanical Design,* vol. 114, September 1992, pp. 477-485.

[VG80] Vasilev, N. B. and Gutenmajer, V. L., *Rectas y Curvas*, (Spanish translation), MIR Publishers, Moscow, 1980.

[Wo94] Woo, T. C., “Visibility maps and spherical algorithms,” *Computer Aided Design*, vol. 26, No. 1, January 1994, pp. 1-16.