APERTURE ANGLE OPTIMIZATION PROBLEMS

by

Elsa Patricia Omaña-Pulido

School Of Computer Sience
McGill University
Montreal, Quebec

August 1994

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

ISBN 0-612-05609-0

Contents

Abstract

Let Q be a convex polygon in E^2 and x be a point in E^2 not contained in Q. The *aperture angle* $\theta(x)$ with respect to Q is defined as the angle subtended by the cone that: (1) contains Q, (2) has apex at x, and (3) has its two rays emanating from x tangent to Q. *The Aperture-Angle Problem* in the Euclidean Plane E^2 is defined as follows: Given two disjoint convex sets P and Q in E^2, find a point $X \in P$ such that $\theta(X)$ is the maximum value of the aperture angle function, and a point Z in P such that $\theta(Z)$ is the minimum value. We present 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 a convex polygon P (n and m are the number of vertices, respectively). We also present algorithms with complexities $O(n \log m)$ and $O(n + m)$ for computing the *maximum* aperture angle with respect to Q. To compute the minimum aperture angle we modify the latter algorithm obtaining an $O(n + m)$ algorithm. Finally, we prove an $\Omega(n)$ time lower bound for the maximization and minimization problems, and an $\Omega(m)$ time lower bound for the minimization problem.

In three dimensions we find the solution to the following problem: Given a convex polyhedron $\mathbf{K}$ and given a segment $\mathbf{ab}$ that does not intersect $\mathbf{K}$, find a point $\mathbf{X} \in \mathbf{K}$ such that the aperture angle defined as $ang(\mathbf{aXb})$ is the maximum of the aperture angle function. We present a solution whose time complexity is linear with respect to the number of edges of $\mathbf{K}$, i. e., it is $O(n)$ if the corresponding convex polyhedron K has n vertices. We prove an $\Omega(n)$ time lower bound for this problem.

Résumé

Soit Q un polygone convexe dans E^2 et x un point de E^2 n'appartenant pas à Q. L'*angle d'ouverture* $\theta(x)$ par rapport à Q est défini comme l'angle soustendu par le cone qui: (1) contient Q, (2) admet x comme apex, et (3) dont les rayons originant à x sont tangents à Q. Le problême *Angle d'ouverture* dans le plan euclidien E^2 se définit comme suit: étant donné deux ensembles convexes P et Q de E^2, trouver un point $X \in P$ tel que $\theta(X)$ maximise la fonction "aperture angle", et un point Z dans P tel que $\theta(Z)$ la minimise.

Nous présentons un algorithme fonctionnant en temps $O(n + m)$ pour calculer l'angle d'ouverture *minimum* par rapport à un polygone convexe Q lorsque x peut varier dans un polygone convexe P (n et m sont le nombre de sommets de Q et P, respectivement). Nous donnons aussi des algorithmes de complexité $O(n \log m)$ et $O(n + m)$ pour calculer l'angle d'ouverture *maximum* par rapport à Q. Pour calculer l'angle d'ouverture minimum nous modifions l'algorithme précédent pour obtenir une solution en temps $O(n + m)$. Finalement, nous prouvons une borne inférieure de $\Omega(n)$ pour le problème de maximisation et minimisation, ainsi qu'une borne inférieure de $\Omega(m)$ pour celui de minimisation.

Dans le cas troidimensionnel nous résolvons le problême suivant: étant donné un polyhèdre convexe **K** et un segment **ab** qui n'intersecte pas **K**, trouver un point $\mathbf{X} \in \mathbf{K}$ tel que le "aperture angle" défini comme l'angle $ang(\mathbf{aXb})$ maximise la fonction angle d'ouverture. Nous presentons une solution exécutable en temps linéaire par rapport au nombre d'arètes de **K**, ce qui signifie $O(n)$ quand le polyhèdre convexe **K** possède n sommets. Nous prouvons une bonne inférieure de $\Omega(n)$ pour ce problême.

Acknowledgments

Special thanks are due to my supervisor Prof. Godfried T. Toussaint who introduced me to the world of Computational Geometry and originally suggested the subject of this thesis. His continuous encouragement and support contributed in many ways to the completion of this work.

The author wishes to express her sincere appreciation to Eduardo Rivera-Campo for his untiring support and guidance. The successful completion of this work would have not been possible without his assistance.

The author would also like to thank Prof. David Avis, Prof. Luc Devroye and Prof.-Ferrán Hurtado for the many helpful discussions and generous advice.

I am grateful to Douglas Rodgers, Hazel Everett and Patricia Butrón for patiently reviewing parts of the manuscript.

I also would like to thank all my friends, in Montreal, that made the Canadian winter a lovely time to live, in particular Patricia Butrón with whom I share wonderful memories.

The author wishes to recognize Franca, Lisa, Vicky and specially Lorraine for their support.

The *Consejo Nacional de Ciencia y Tecnología* of Mexico facilitated this investigation by awarding the author a post-graduate scholarship for which she is gratefully indebted. This investigation was also supported by NSERC Grant no. OGP0009293 and FCAR Grant no. 93-ER0291.

CHAPTER 1

INTRODUCTION

In this thesis we present some results related to field of *constrained visibility*, a subject belonging to the field of Computational and Combinatorial Geometry. The basic concepts and methods of computational geometry are related to the origins of Geometry in Greece [To92]. The most modern conception is due to Shamos [Sh77]. Since then many books have appeared. We refer to [PS88] for a good survey of the basic information needed. Computational Geometry uses many concepts and results from other fields such as Combinatorics ([Sl91] and [Tu84]), Computing Theory, Discrete Geometry, and Geometry in its most general meaning. In theoretical computer science the main tools needed are Design and Analysis of Algorithms, Complexity Theory and Data Structures. We suggest as good references [AHU74], [AHU83], [Kn73] and we refer to[CLR90] for one of the most recent text books in Analysis of Algorithms and Data Structures.

A polygon P is defined as an ordered sequence of at least three points $P = [p_1, p_2,..., p_n]$ in the plane, called vertices, and n ($n \geq 3$) line segments p_1p_2, p_2p_3,...., $p_{n-1}p_n$, $p_n\, p_1$ called edges. A simple polygon is a polygon with the constraint that non-consecutive edges do not intersect. A simple polygon is a Jordan curve and thus it divides the plane into three regions: The interior of the polygon, the exterior and the boundary. We consider any polygon as the boundary together with its interior. Then the polygons are *compact* sets. Recall that a set (contained in an euclidean space of any dimension) which is closed and bounded is called compact. The vertices of a simple polygon can be classified as convex and concave. A vertex p_i is convex if p_{i-1}, p_i and p_{i+1} define a right turn, if the vertices are ordered in clockwise order. Analogously, vertex p_j is concave if p_{j-1}, p_j and p_{j+1} form a left turn. A simple polygon whose vertices are all convex is called a convex polygon. If a convex polygon has n vertices we refer to it as a n-convex polygon. A more implicit way to define a convex polygon is as a finite intersection of halfplanes.

Visibility plays a singular role in the manufacturing industry in such problems as accessibility analysis in machining [Wo94], [TWG92], [CW92] and visual inspection [SR90] as well as computer graphics, robotics, computer vision, operations research and several other disciplines of computer science and computer engineering [O'R87], [Sh92]. In 1973 Victor Klee proposed the problem of finding the minimum number of points in a simple polygon to see (or illuminate) all the other points. Imagine the simple polygon as an art gallery, then imagine the set of points for watching (or illuminating) as guards (or lamps). The answer to this question was found by Chvatal [Ch75], who showed that $\left\lfloor \frac{n}{3} \right\rfloor$ guards are always sufficient and sometimes necessary to illuminate the gallery.

Later Fisk [Fi78] found a shorter proof of the same result. Although $\left\lfloor \frac{n}{3} \right\rfloor$ guards are sufficient it is often the case that a smaller number may do the job as well. From the point of view of Computer Science there exists a natural concern to find and to develop algorithms for finding the *minimum number of vertices* to see the whole polygon. However, Lee and Lin [LL86] have shown that this problem is NP-hard, and Aggarwal [Ag84] has proved that even if the guards are allowed to be *any point* in the polygon the problem is still NP-hard. In spite of those results, Avis and Toussaint [AT81] have shown that the required $\left\lfloor \frac{n}{3} \right\rfloor$ guards to see the gallery can be placed in polynomial time. Using the recent algorithm for triangulating a polygon developed by Chazelle [Ch91] and the last result about a three-coloring of the vertices of a simple polygon in linear time, the guarding problem can be solved in $O(n)$ time. In the special case of orthogonal polygons with n vertices, $\left\lfloor \frac{n}{4} \right\rfloor$ guards are sufficient and some times necessary [KKK83], and they can be placed in $O(n)$ time [EOW84].

We have been using many times the word "see" or " illuminate", so we say that two points x and y in a polygon P are visible if the line segment xy is totally contained in the polygon. The set of points in P which are visible from x is a polygon. Such a polygon is called the *visibility polygon* of x and it is denoted by $V(x, P)$.When the setting is an art gallery problem the distinguished set of points are called guards, and lamps if the setting is an illumination problem.

A polygon P is said to be *covered* by a collection of subsets of P if the union of these subsets is exactly P. A guard set G is said to cover P if the collection of sets $\{V(g, P): g \in G\}$ covers P. The *Art Gallery Problem* for a polygon is to find a minimum cardinality covering guard set G for P. Tom Shermer [Sh92] presented an excellent survey paper about art gallery problems.

The traditional model 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, more realistic models of visibility have been considered 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 the space where a camera may be placed in such a way that the polygon lies completely in the field of vision of the camera. A member x of 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. [Aal93] obtained optimal algorithms for finding all the θ-visible points among a set S. Devroye and Toussaint [DT93] investigated the cardinality of the θ-visible points among a set of special points which are the intersection of a set of random lines.

Jorge Urrutia proposed in 1992 the *illumination of a stage* problem as follows:

Given n points in the plane where n floodlights are to be placed, and given n angles representing the aperture of the floodlights, decide how to assign the floodlights to the points and how to fix their rotational angles, in order to light up some target. Jorge Urrutia posed the version of this problem for lighting up a stage. An intuitive way to solve this problem is illuminating by crossing the floodlights using a greedy technique. However, Bose et. al [Bal93] gave a counterexample where this technique fails. They also proved that given three angles summing up to 2π, and given n points in the plane and a partition $k_1 + k_2 + k_3 = n$, the plane can be partitioned into three wedges of the given angles in such a way that the i-th wedge contains k_i of the points. Using this result they proved that lights of specified angles (none of them exceeding π) can be placed at n fixed points in the plane to illuminate the entire plane if and only if the angles add up to at least 2π. Later, Czyzowicz et. al [CRU93] solved a particular case of the stage light problem. Given a set $F = \{f_1, f_2, \ldots, f_n\}$ of n floodlights each with angle α_i, they associate to the set of floodlights an angular cost:

$$\alpha(F) = \Sigma\, \alpha_i$$

They solved the following problem: Given a stage represented by a line segment S and a set of n points $P = \{p_1, p_2, \ldots, p_n\}$, determine a set of floodlights F that illuminates S, such that the angular cost of F is minimum and each floodlight is located at some point of P. They solved this problem in $O(n \log n)$ time allowing more than one lamp to be placed at a given point, and they proved that their algorithm is optimal. First of all, they solved the problem when S is extended to be the real line. Then, they constrained the problem to a segment S. Although the problem has been solved when more than one lamp is allowed per point, it is still open when only one lamp is allowed.

The simplest of these problems is often found as an exercise in calculus texts and it is 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 subtended at the observer's eye by the top and the 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]. The same solution holds for the more general problem where the picture may not be orthogonal to the floor [VG80].

In this thesis we consider a generalization of the "picture-on-the-wall" problem, in two and three dimensions. We define the aperture angle $\theta(x)$ from a point $x \in E^2$ with respect to a simple polygon Q in E^2 ($x \notin Q$) as the angle subtended by the cone such that: (1) it contains Q, (2) it has apex at x, and (3) its two rays emanating from x are tangent to Q. First of all, we are interested in computing the aperture angle from 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. Find a point $X \in P$ such that $\theta(X)$ is the maximum value of the aperture angle function, and find a point $Z \in P$ such that $\theta(Z)$ is the minimum value. We call this problem *The Aperture-Angle Optimization Problem in 2-D*. Note that we use the word "optimization" because we solve a maximization and a minimization problem.

In three dimensions we find the solution to the following problem: Given a convex polyhedron **K** and given a segment **ab** that does not intersect **K**, find a point $\mathbf{X} \in \mathbf{K}$ such that the aperture angle *ang*(**aXb**) is the maximum value of the aperture angle function. Note that the corresponding aperture angle is defined as in the two dimensional case. We present a solution whose time complexity is linear with respect to the number of edges of **K**, i. e., it is $\Theta(n)$, if the corresponding convex polyhedron **K** has n vertices.

This thesis is organized in two main stages, the first one is covered by Chapter 2, which contains a description of the required background and techniques. The other part contains chapters 3 and 4 and concerns the solution of the aperture angle problems described in the last two paragraphs.

In Chapter 2 we describe some methods to be used in the remaining chapters. The first section contains a brief description of the model of computation used by the different methods described in this thesis. Section 2.3 contains some basic concepts from Euclidean Geometry which are used along this work. The first method to be described in Section 2.4, is due to Chazelle and Dobkin [CD87] for determining whether a given line intersects a given convex polygon. Section 2.5 shows how to compute the minimum distance between two convex polygons, this method was found by Edelsbrunner [Ed85]. The last section of Chapter 2 describes two different methods for finding the *common external tangents* and the *critical separating lines of support* between two convex polygons. One was developed by Rohnert [Ro86] and the other one is due to Toussaint [To83] using "*Rotating Calipers*".

Chapter 3 describes a solution to the *Aperture-Angle Optimization Problem in 2-D*. The following instances of this problem are solved in Sections 3.2 to 3.5: the Segment-Line Problem, the Segment-Segment Problem, the Segment-Polygon Problem and the Polygon-Line Problem. Section 3.6 is dedicated to proving some results concerning the geometric properties for the general case, the Two Convex Polygons Problem, that allow us to characterize the solution of the problem as well as to find it. In Section 3.7 we present the algorithms to determine the solution for the *Aperture-Angle Optimization Problem in 2-D* and its complexity analysis. In the last section we prove lower bounds for the maximization and minimization problems. The Aperture-Angle Optimization Problem in 2-D was studied independently by Hurtado [Hu93].

In Chapter 4 we solve the *Aperture-Angle Optimization Problem in 3-D* in the case where the object to be maintained in the field of view is a segment. This chapter contains seven sections, the second one provides the basic definitions to be used in the chapter. Section 4.3 we solve the Segment-Plane Problem. In Section 4.4 we solve the Segment-Line

Problem which helps us to solve the segment-polyhedron case. The remaining sections are devoted to solving each of the following problems: the Segment-Segment Problem, the Segment-Polygon Problem and the Segment-Polyhedron Problem.

CHAPTER 2

FUNDAMENTAL GEOMETRIC TOOLS

2.1 Introduction

In the general introduction we have defined the aperture angle from a point $x \in E^d$ with respect to a convex polyhedron in E^d that does not contain x. This definition involves three points: one of them is x itself, and the other two points, p and q, which are the tangency points between the convex object and the rays that define the aperture angle $ang(p\ x\ q)$. These three points define a unique plane that contains the circle passing through p, q and x. Then, we use basic results from euclidean geometry in the plane which relate circles and angles. In fact, these basic results allow us to compare the aperture angles from points lying in a convex set contained in E^3. Those results are stated in the following section as observations.

The first goal in any subject related to computation is to find the solution of a problem. There are some methods of computation that require the solution of other problems as intermediate steps. In particular, the solution of the problems analyzed in this work use as intermediate steps the solution of some other problems that have been already solved. We describe from Section 2.4 until Section 2.6 the algorithms to be used in chapters three and four. Apart from finding a solution there also exists a great concern to develop efficient methods of solution. In Section 2.2 we describe the model of computation to be used by the algorithms developed in this work and which is the *Real Random Access Machine* (*RRAM*). The algorithms described in this chapter are also based on the *RRAM* model of computation (Refer to [PS88] and [AHU74]).

2.2 Models of Computation

To evaluate the algorithms in this thesis we need to fix a model of computation. This describes the idealized computer that we will use to "execute" our algorithms, and we will interpret the running time and storage requirement on this computer as a measure of the complexity of the algorithm.

The fundamental model of computation and complexity theory is the Turing Machine. A model that is more convenient for the description of our algorithms is the *Random Access Machine* (*RAM*). It has been proved to be equivalent to the Turing Machine in computational power and to be approximately equivalent (with a polynomial coefficient) in speed.

For our purposes, however, we will use a variation that is more suitable for Computational Geometry, the *Real RAM* (*RRAM*). Unlike the *RAM*, this model can handle real numbers of arbitrary precision. In the following, we give a brief description of this model. The *RRAM* contains a memory consisting of an unbounded sequence of registers $r_0, r_1, \ldots, r_j, \ldots$ each of which is capable of holding either a single real number or an integer. The memory is used to hold the input and the output for the algorithm. We allow the following primitive operations all of which are assumed to take unit time.

1. The arithmetic operations (+, -, *, /)
2. Comparisons between two real numbers ($<, \leq, =, \geq, >$)
3. Indirect addressing of memory (integer addressing only)
4. The k-th root, trigonometric functions, EXP and LOG (in general analytic functions)

Algorithms can be evaluated by several criteria. In this work we use the rate of growth of the time required to solve larger and larger instances of a problem. The time needed by an algorithm is expressed as a function of the size of a problem and it is called *time complexity*. The *size* of a problem is an integer which measures the quantity of input data. In this specific model the complexity can be taken over all inputs of a specific size, in which case the complexity is called *worst-case complexity*. If the complexity is taken as the average complexity over all inputs of given size, then the complexity is called the *expected complexity*. In this work we will use the *worst-case* complexity.

While the *RRAM* is an idealized computer it seems reasonably close to existing real-world computers, provided that:

1) the size of the problem is small enough to fit in the main memory of a computer,

2) the integers used in the computation are small enough to fit in one computer word, and

3) we ignore problems of precision due to rounding errors.

2.3 Geometric Preliminaries

Let A, B and X be points on a circle C and let P be the open halfplane defined by the line through A and B that contains X. Let Y be a point in P.

Observation 2.3.1: If $Y \in ext(C)$ then $ang(A\ Y\ B) < ang(A\ X\ B)$ (refer to Fig. 2.3.1.a).

Observation 2.3.2: If $Y \in C$ then $ang(A\ Y\ B) = ang(A\ X\ B)$ (refer to Fig. 2.3.1.b).

Observation 2.3.3: If $Y \in int(C)$ then $ang(A\ Y\ B) > ang(A\ X\ B)$ (refer to Fig. 2.3.1.c).B

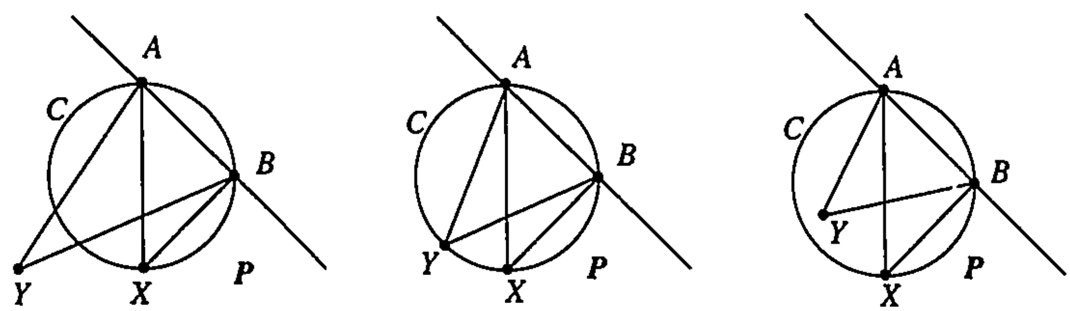

Fig. 2.3.1.a Fig. 2.3.1.b Fig. 2.3.1.c

Before we continue with the corresponding description of the algorithms to be used in chapter three let us define some concepts from convex geometry related to computational geometry.

Let P and Q be two disjoint polygons in the plane. We assume that $P = [p_1, p_2, \ldots, p_m]$ is represented by an array in clockwise order and $Q = [q_1, q_2, \ldots, q_n]$ is represented by an array in counterclockwise order. Then all the chains defined are represented in clockwise order for P and in counterclockwise order for Q.

Definition 2.2.1: A line L is a *critical separating line of support* of P and Q if (1) it separates P from Q, and (2) it is tangent to both P and Q.

Let the critical separating lines of support of P and Q be tangent at $\{p_i, q_t\}$ and $\{p_j, q_k\}$ respectively. These lines partition the boundaries of P and Q into four chains. They also divide the plane into four regions (cones), two of which are empty, one containing P and the other Q. 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. If the interior of the triangle does not contain a vertex of P, in fact the vertex $p_{i+1} = p_j$, then the edge $p_i p_{i+1}$ defines the *interior separating boundary of P with respect to Q*, denoted simply by $CSS(P)$. Otherwise, the chain $(p_i, p_{i+1}, \ldots, p_j)$ contained in the interior of the triangle in R_P defines the *interior separating boundary of P with respect to Q* (refer to Fig. 2.3.2). The complement chain, $bd(P)$ - $CSS(P)$, is denoted by $CSS(P)^c$. The other two chains $CSS(Q)$ and $CSS(Q)^c$ are similarly defined.

Definition 2.2.2: A line L is an *external common tangent* of P and Q if (1) it is tangent to P and Q, and (2) it leaves P and Q in one of the closed halfplanes defined by L.

Let the external common tangents to P and Q be tangents at $\{p_r, q_i\}$ and $\{p_s, q_j\}$ respectively. These four points define the convex quadrilateral $[p_r, q_i, q_j, p_s]$ if $p_r \neq p_s$ and $q_i \neq q_j$; or they define a triangle if $p_r = p_s$ or $q_i = q_j$. Without loss of generality suppose that $p_r \neq p_s$ and $q_i \neq q_j$. If $[p_r, q_i, q_j, p_s]$ contains p_{r+1} in its interior then the chain $(p_r, p_{r+1}, \ldots, p_i, p_{i+1}, \ldots, p_j, p_{j+1}, \ldots, p_s)$, denoted by $ECT(P)$, is contained in the interior of $[p_r, q_i, q_j, p_s]$ (refer to Fig. 2.3.2). If $[p_r, q_i, q_j, p_s]$ does not contain p_{r+1} in its interior then $p_{r+1} = p_s$ and $ECT(P)$ is defined by the edge $p_r p_{r+1}$. If $p_r = p_s$ or $q_i = q_j$ then the chain $ECT(P)$ is defined analogously and is contained in a triangle or is an edge of such a triangle. Let $ECT(P)^c$ be the complement chain of $ECT(P)$. Similarly the chains $ECT(Q) = (q_i, q_{i+1}, \ldots, q_k, q_{k+1}, \ldots, q_t, q_{t+1}, \ldots, q_j)$ and $ECT(Q)^c$ are defined.

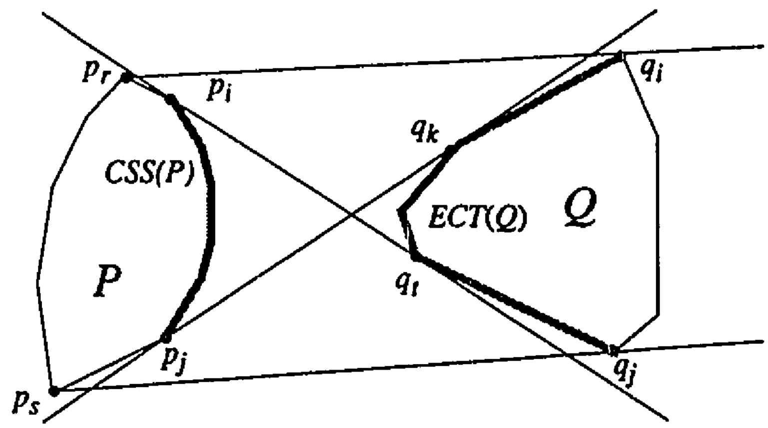

Fig. 2.3.2

We refer as *common tangents* of two convex polygons to the two pairs of lines of support, one pair is defined by the two external common tangents and the other pair is defined by the two critical separating lines of support.

Note that these definitions can be used for particular cases, for example one of the polygons is a segment AB. With out loss of generality suppose Q is such a segment. Then, the support points defined by the critical lines of support are $\{p_i, A\}$ and $\{p_j, B\}$ (see Fig. 2.3.3). If $L(A, B)$ intersects $int(P)$ then the chain $CSS(P)$ is contained in the triangle (p_j, C, p_i). The point C is the endpoint of AB that is closer to P (using the definition of distance from a point a to a polygon P as min $\{d(a, p): p \in P\}$, where d denotes the euclidean distance). Analogously the external common tangents define two pair of tangency points $\{p_r, A\}$ and $\{p_s, B\}$ (see Fig. 2.3.3). If $L(A, B)$ intersects $int(P)$ then the chain $ECT(P)$ is contained in the triangle (p_j, C, p_i). The point C is the endpoint of AB that is further to P (using the definition of distance from a point a to a polygon P as max $\{d(a, p): p \in P\}$ and d denoting the euclidean distance).

Recall that there exists a useful searching technique for a value v on a sorted sequence A of k numbers. It consists in comparing the value in the midpoint of the sequence A against v and eliminating half of the sequence each time. This well known process is called *Binary Search*, and it can be done in an iterative or recursive way in $O(\log k)$ time. In particular, if the vertices of a polygon P are stored in a linear array, this is already sorted in a radial order. Then, binary search can be performed over the array for finding some vertices of P with specific properties. This technique will be used in the following sections.

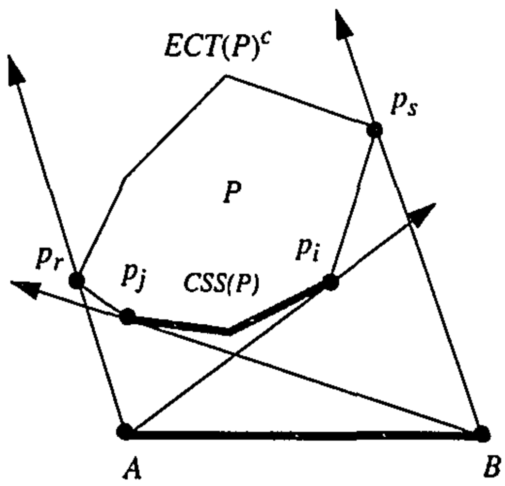

Fig. 2.3.3

2.4 Intersecting a Convex Polygon with a Line.

In this section we describe an algorithm to determine if a given line L intersects a given convex polygon P with m vertices. This problem and more general problems in two and three dimensions were solved by Chazelle and Dobkin in [CD87]. They assumed that the polygon is stored in a linear array.

To solve this decision problem they define a *unimodal* function as follows: A real function f defined on the integer subset $\{1, 2, ..., m\}$ is said to be *unimodal* if there exists an integer r $(1 \leq r \leq m)$ such that f is strictly increasing (respectively, decreasing) on $[1, r]$ and decreasing (increasing) on $[r+1, m]$, with $f(r) \leq f(r+1)$ $(f(r) \geq f(r+1))$. The point r is referred to as the *turning point.*

For finding the turning point r of a unimodal function $O(\log m)$ time is required using Fibonacci search. Chazelle and Dobkin have extended this search to find the turning point r of a *bimodal* function, defining such a function as follows.

Definition 2.4.1: A real function on $[1, m]$ is said to be *bimodal* if there exists a point $r \in [1, m]$ such that $f(r), f(r+1), ..., f(m), f(1), ..., f(r-1)$ is unimodal.

They show that the oriented distance $h(p_i, L, v)$ from $p_i \in P$ to L with respect to a point v is bimodal. Then, $h(p_i, L, v) = - d(p_i, L)$ if p_i and v lie on opposite sides of L, and $h(p_i, L, v) = d(p_i, L)$ if they lie on the same side. To find the respective turning point r they define an auxiliary unimodal function $f(x)$ as follows:

$$g(x) = min\{f(x), (x-1)(f(m) - f(1))/(m-1) + f(1)\}$$

Such a function can be evaluated in constant time. Then the minimum of $g(x)$, which is also the minimum of $f(x)$, can be evaluated in $O(\log m)$ time using Fibonacci search. Therefore it follows Lemma 2.4.1.

Lemma 2.4.1: The extrema of a bimodal function $f(1),\ldots,f(r\text{-}1), f(r), f(r+1),\ldots,f(m)$ can be computed in $O(\log m)$ time.

Using the fact that $h(p_i, L, v)$ is a bimodal function and by Lemma 2.4.1 the following theorem is stated.

Theorem 2.4.1: The intersection of an infinite line with a m-convex polygon P can be found in $O(\log m)$ time.

2.5 Computing The Minimum Distance between Two Convex Polygons.

The problem of finding the minimum distance $d^*(P, Q)$ between two disjoint convex polygons P and Q, with m and n vertices respectively, has been solved in $O(\log m + \log n)$ time by Edelsbrunner [Ed85]. He also determines a pair of points p_c and q_c that realize such minimum distance. He assumes that P and Q are stored in one-dimensional arrays. He proves the following lemmas on which his algorithm is based.

Lemma 2.5.1: If $d^*(P, Q) > 0$, then there exist points $p_c \in P$ and $q_c \in Q$ that realize $d^*(P, Q)$ such that p_c and q_c are vertices or either of them is a vertex and the other lies on an edge

Lemma 2.5.2: Let $d^*(P, Q) > 0$ and $p \in P$ such that there exists $q \in Q$ such that $d(p, q) = d^*(P, Q)$. Then p is the only common point of P and the segment pq' for every $q' \in Q$.

Edelsbrunner establishes a criterion of binary search over two chains P' and Q' simultaneously. These chains are contained in P and Q respectively. They are chosen in such a way that the following two conditions are satisfied:

i) If $p_c = p \in P$ and $q_c = q \in Q$ define $d^*(P, Q)$ then $p \in P'$ and $q \in Q'$, and

ii) the edges of P' and Q' do not belong to the convex hull of $P \cup Q$.

To build P' and Q' let $v \in P$ and $w \in Q$ be any points in P and Q respectively. Let $\theta(v) = ang(w_2 v w_1)$ be the aperture angle defined from v with respect to Q. Thus, w_1 and w_2 define the chain $Q' = (w_1 = q_i, q_{i+1}, \ldots, q_j = w_2)$ in counterclockwise order. Analogously by using $\theta(w)$ the chain $P' = (v_1 = p_r, p_{r+1}, \ldots, p_s = v_2)$ is defined in clockwise order. The algorithm by Chazelle and Dobkin [CD87] can be used to find the tangents to a polygon from a specific point. As we mentioned in Section 2.4, it can be performed in $O(\log m + \log n)$ time.

Once P' and Q' are determined, Edelsbrunner establishes a binary elimination as criterion for computing a pair of vertices, a vertex and an edge, or a pair of edges that contain all pairs $\{p, q\}$ realizing $d^*(P, Q)$ in $O(\log m + \log n)$. Finally he determines between all pairs $\{p, q\}$ such that $d(p, q) = d^*(P, Q)$ a specific pair for which $p = p_c$ and $q = q_c$. Therefore, the algorithm and the theorem that states its complexity can be summarized as follows.

<u>Algorithm for computing the minimum distance between two disjoint polygons</u>

Input: Two disjoint convex polygons P and Q, with m and n vertices respectively.

Output: The minimum distance $d^*(P, Q)$ between P and Q. The pair of points $p_c = p \in P$ and $q_c = q \in Q$ such that $d(p_c, q_c) = d^*(P, Q)$.

Begin

Step 1.- Build the chains P' and Q'.

Step 2.- Perform a binary elimination over P' and Q' to find a pair of vertices, a vertex and an edge, or a pair of edges that contain all pairs $\{p, q\}$ defining $d^*(P, Q)$.

Step 3.- Exit with $p_c = p \in P$ and $q_c = q \in Q$ such that $d(p_c, q_c) = d^*(P, Q)$.

End

Theorem 2.5.1: Let P and Q be two disjoints convex polygons on the plane with m and n vertices respectively. Then, the minimum distance $d^*(P, Q)$ between them and the points $p_c = p \in P$ and $q_c = q \in Q$ that realize it can be computed in $O(\log m + \log n)$.

2.6 Finding the Common Tangents Between Two Convex Polygons.

2.6.1 Algorithm used by Rohnert to find *Useful Supporting Segments*: Shortest Path in the Plane with Convex Polygons.

Hans Rohnert [Ro86] solved the following problem: Given a family of f disjoint convex polygons (called obstacles) $P_1, P_2, \ldots, P_f$ with $n_1, n_2, \ldots, n_f$ vertices respectively and such that $\sum_{i=n}^{f} n_i = n$, compute the shortest path between two arbitrary query points s and t such that $s,t \notin int(P_i)$, for all $1 \leq i \leq f$. To solve this problem he uses the fact that the shortest path between s and t is determined by edges of the convex polygons and common tangents between pairs of them. He defines *useful supporting segments* between pairs of obstacles P_i and P_j as those common tangents that do not intersect the interior of any polygon placed between the corresponding pair P_i and P_j. He defines a total order for the set of common tangents in order to distinguish those that are useful from those that are not. Finally he represents the problem as a graph $G = (V, E)$ in order to use Dijkstra's algorithm. The set of vertices V is defined by the n vertices plus the two points s and t. The corresponding set of edges E is defined by the union of the set of n boundary edges, of the f polygons, and the set of useful supporting segments. We show a general description of his algorithm, as well as a detailed description of the algorithm for finding the useful supporting segments between pairs of obstacles.

<u>*Algorithm Ro86*</u>

Input: A family of f disjoint convex polygons $P_1, P_2, \ldots, P_f$ and two points s and t, such that $s,t \notin int(P_i)$.

Output: The shortest path between s and t.

Begin

Step 1.- Construct for each pair of obstacles P_i and P_j the four common tangents.

Step 2.- Identify the useful supporting segments from the not useful ones.

Step 3.- Build a graph $G = (V, E)$ where the set of vertices V is defined by the n vertices plus the two points s and t. The corresponding set of edges E is defined by the union of the set of n boundary edges, of the f polygons, and the set of useful supporting segments. Use Dijkstra's algorithm.

End

Step one can be done in $O(\log n_i + \log n_j)$ time complexity for each pair P_i and P_j; for all pairs it is done in $O(f^2 \log n/f)$ time and $O(n + f^2)$ space. Finally Step 3 is done in $O(f^2 + n \log n)$ time. Therefore the shortest path between s and t can be found in $O(n + f^2)$ space and $O(f^2 + n \log n)$ time after $O(n + f^2 \log n)$ preprocessing time. The preprocessing time corresponds to the computation of that part of the visibility graph which is used to compute the shortest path.

Algorithm for Finding the four Common Tangents

Input: Two disjoint polygons P and Q in the plane with m and n vertices respectively.

Output: The pair of external common tangents and the pair of critical separating lines of support.

Begin

Step 1.- Determine the two points $p \in P$ and $q \in Q$ that determine the minimum distance between P and Q.

Step 2.- Find the bisector L_{pq} of pq.

Step 3.- Using binary search, determine for each polygon the pair of vertices that determine the minimum and maximum distances to L_{pq}. Denote them by $\{p_c, p^*\}$ and $\{q_c, q^*\}$. Let U_p and U_q be the upper chains determined by these two pair of vertices in clockwise order. Analogously, let L_p and L_q be the lower chains.

Step 4.- Using binary search over U_p and U_q determine the upper external common tangent. Analogously, determine the lower external common tangent by using binary search over L_p and L_q. The two external common tangents determine *ECT(P)*, and *ECT(Q)*. Using binary search over those chains simultaneously determine the critical separating lines of support.

End

Lemma 2.6.1: The four common tangents between P and Q can be computed in $O(\log n + \log m)$ time.

Proof: Step 1 and Step 2 are found using Edelsbrunner [Ed85] algorithm in $O(\log n + \log m)$ time. Using binary search the two pairs $\{p_c, p^*\}$ and $\{q_c, q^*\}$ and the four common tangents can be found in $O(\log n + \log m)$.

2.6.2 Finding The Common Tangents with "Rotating Calipers"

"Rotating calipers" is the name that Toussaint [To83] used to refer to the process of rotating parallel lines of support around a convex polygon P. The pair of vertices that admits parallel lines of support is called an *antipodal pair*. As it is known Shamos, in his Ph. D. thesis [Sh77], developed an algorithm to generate the n pairs of antipodal vertices in $O(n)$ time. Using parallel lines of support the algorithm of Shamos finds in $O(n)$ time a pair $\{p_i, p_j\}$ of antipodal vertices in a direction fixed previously and in such a way that the polygon lies always to the right, or to the left, of both lines. The principle behind the generation of all parallel lines of support is described as follows: Let θ_i be the angle that the supporting line at p_i defines with the edge (p_i, p_{i+1}) in clockwise order. Define θ_j analogously and suppose $\theta_i < \theta_j$. Then consider the new lines of support making the angles $\theta_j = \theta_j - \theta_i$ and $\theta_i = 0$ at p_i and p_j respectively. Thus, the pair $\{p_{i+1}, p_j\}$ becomes the next antipodal pair. This process is continued until the new pair of lines of support are in the initial position at p_i and p_j, respectively. When $\theta_i = \theta_j$ there are three antipodals pairs.

Toussaint extends this idea showing that several sets of lines of support can be used in one polygon or one pair of lines of support can be used on several polygons. In the latter case, one pair is used to merge two convex polygons as well as to find the critical separating lines of support. Toussaint observes that the problem of merging two convex polygons P

and Q with m and n vertices respectively consists of finding two pairs of vertices $\{p_i, p_j\}$ $\in P$ and $\{q_k, q_l\} \in Q$ such that $\{p_i q_k\} \cup ECT(P)^c \cup \{q_l p_j\} \cup ECT(Q)^c$ form the convex hull of $P \cup Q$. He calls *bridges* the segments $p_i q_k$ and $q_l p_j$. In order to establish his result about merging two convex polygons he defines a *co-podal* pair of points $\{p_i, q_j\}$with respect to a support line $L(p_i, q_j)$ if it supports both points simultaneously. To determine each one of the co-podal pairs that define the union of P and Q he stated the following theorem.

Theorem 2.6.1: Two vertices $p_i \in P$ and $q_j \in Q$ are *bridge* points if, and only if they form a co-podal pair, and the vertices $p_{i-1}, p_{i+1}, q_{j-1}$ and q_{j+1} all lie on the same side of $L(p_i, q_j)$.

Therefore, using support lines through co-podal pairs of points it is possible to merge two convex polygons in $O(n)$ time. To determine if a co-podal pair is a bridge is done in $O(1)$ time and thus the algorithm runs in $O(n)$ time.

For finding critical separating lines of support Toussaint uses antipodal pairs and the following result.

Theorem 2.6.2: Two vertices p_i and q_j determine the critical separating lines of support if and only if they form an antipodal pair and $\{p_{i-1}, p_{i+1}\}$ lies on one side of $L(p_i, q_j)$ while $\{q_{j-1}, q_{j+1}\}$ lies on the other side of $L(p_i, q_j)$.

Since determining whether an antipodal pair defines a critical separating line of support is done in constant time and because there are $O(n)$ antipodal pairs, the time complexity to determine the two critical separating lines of support is $O(n)$.

CHAPTER 3

SOME APERTURE ANGLE OPTIMIZATION PROBLEMS IN 2 - D

3.1 Introduction

We have mentioned, in the introduction, that this work is related to visibility problems where the angle of vision is constrained in order to see an object with an angle as large as possible. In this chapter we consider *The Aperture-Angle Problem* in the Euclidean Plane E^2. Let Q be a convex polygon in E^2 and x be a point in E^2 not contained in Q. The *aperture angle* $\theta(x)$ with respect to Q is defined as the angle subtended by the cone that: (1) contains Q, (2) has apex at x, and (3) has its two rays emanating from x tangent to Q.

We study several instances of the following general problem:

Given two disjoint convex sets P and Q in E^2, find a point $X \in P$ such that $\theta(X)$ is the maximum value of the aperture angle function, and a point Z in P such that $\theta(Z)$ is the minimum value.

3.2. The Segment-Line Problem

The first simplification will be referred to as the *Segment-Line Problem*, where the convex polygon Q is replaced by a segment AB, and P 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 obtain a characterization of the solution. We will, being motivated by the desire to obtain efficient algorithms, also characterize the aperture angle function itself.

Problem Statement: Given a segment AB and a line L that does not meet AB, find a point $X \in L$ such that the angle AXB is the maximum value for $\theta(x) = ang(AxB)$, and find a point $Z \in L$ such that the angle AZB is the minimum value.

Maximization Problem

We assume that the given segment AB is not parallel to the given line L. For this simple case the reader can easily verify that the solution point X must lie at the perpendicular projection of the midpoint of AB on L. Without loss of generality L can be assumed to be the real line.

Let D be the intersection point between the line $L(A, B)$ containing the segment AB and L. Since A, B and D lie on a line, the aperture angle $ang(ADB) = \theta(D)$ is zero. If x moves in the direction of $+\infty$ or $-\infty$ then $\theta(x)$ asymptotically goes to zero. Thus, the solution for the minimization problem becomes trivial. However, we may conclude that the maximum aperture angle is attained at some point X which lies in either $(-\infty, D)$ or (D, ∞). Let $\theta(x) = ang(AxB)$ 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 $+\infty$. Let X be a point in L where $\theta(x)$ reaches its maximum value $\theta(X) = ang(AXB)$. We characterize, in Lemma 3.2.1, the maximum over one of the unbounded intervals. Suppose the unbounded interval is $(-\infty, D]$. In order to obtain efficient algorithms (through binary search for example) we would like to give an appropriate characterization of $\theta(x)$. For that purpose we also establish in Lemma 3.2.1 that $\theta(x)$ is an *upwards unimodal* function in the interval $(-\infty, D]$, i.e., $\theta(x)$ has no more than one local maximum in $(-\infty, D]$.

Lemma 3.2.1: If x is constrained to move in $(-\infty, D]$, then the function $\theta(x)$ reaches its maximum value at the point $Y \in (-\infty, D]$, where the circle through A, B and Y is tangent to L. Furthermore, $\theta(x)$ is upwards unimodal in $(-\infty, D]$.

Proof: Let C be the circle through A and B that is tangent to L at a point $Y \in (-\infty, D]$. Any point $z \in (-\infty, D] \setminus Y$ lies outside of C, then by Observation 2.2.1 it follws that $\theta(Y) \geq \theta(z)$ for all $z \in (-\infty, D]$. Let $z_1, z_2 \in (-\infty, D]$ be such that $z_1 < z_2 < Y$ and refer to Fig. 3.2.1. Since the circle C is tangent to L at Y, when C is enlarged continuously with the constraint that it passes through A and B, it must intersect L at a point $z < Y$. In this way the growing circle first intersects z_2 and subsequently z_1. Therefore the circle through A, B and z_2 is smaller than the circle through A, B and z_1. But, since the chord AB is the same length in both circles, the angle it subtends is smaller in the larger circle. Therefore $\theta(z_1) < \theta(z_2)$. It follows that $\theta(x)$ is strictly increasing in $(-\infty, Y]$. A similar argument shows that $\theta(x)$ is strictly decreasing in $[Y, D]$. Therefore $\theta(x)$ is upwards unimodal in $(-\infty, D]$. Q.E.D.

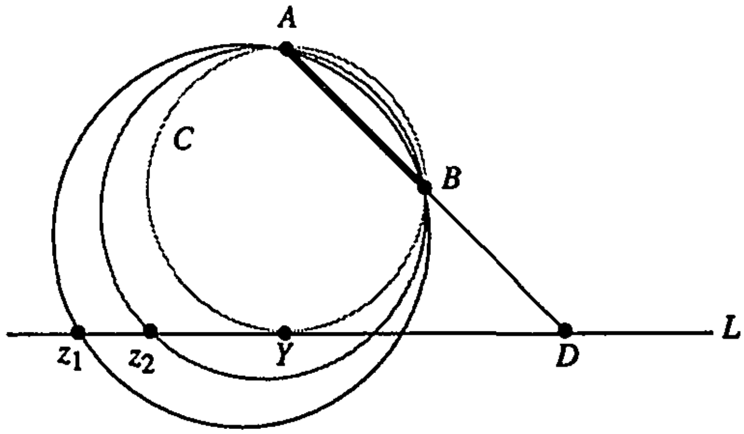

Fig. 3.2.1

Clearly Lemma 3.2.1 can also be established for $x \in [D, +\infty)$ in a similar way. Then, $\theta(x) = ang(AxB)$ is a bimodal function over L. Let $Y_1 \in (-\infty, D]$ and $Y_2 \in [D, +\infty)$ be the two points where the two circles C_1 and C_2 through A and B are tangent to L. Thus, these two points define a maximum over the corresponding interval where they belong, i.e., they are local maxima over L. Therefore, the solution is determined by the point X such that $ang(AXB) = \max\{ang(AY_1B), ang(AY_2B)\}$. Note that the point X defining $\theta(X)$ is determined by the tangent circle with minimum radius. When AB is parallel to L there exists a unique tangent circle. In the particular case when AB is perpendicular to L the points Y_1 and Y_2 are equidistant from D and the two circles that determine them have the same size.

To establish the corresponding algorithm observe that the centers of the tangent circles C_1 and C_2 are determined by the locus of the points $(h, k) \in E^2$ which are equidistant from the two points A and B, as well as from the line L. Then, such a locus describes a parabola for each end point of AB, i.e., two parabolas $P_1(x)$ and $P_2(x)$. The intersection of these parabolas is represented by a quadratic equation in h. The algebraic expression to the corresponding equation is given by:

$$(b_2 - a_2)\, h^2 - 2\,(a_1\, b_2 - a_2\, b_1)\, h + (b_2\, A^2 - a_2\, B^2) = 0$$

Let $a = (b_2 - a_2)$, $b = 2\,(a_1\, b_2 - a_2\, b_1)$ and $c = (b_2\, A^2 - a_2\, B^2)$ be the coefficients on the last equation, then the solution set is defined by:

$$h_1 = \frac{(-b) + \sqrt{b^2 - 4ac}}{2a}$$

$$h_2 = \frac{(-b) - \sqrt{b^2 - 4ac}}{2a}$$

The solution set $\{h_1, h_2\}$ of this equation determines the x-coordinate of the center of C_1 and C_2. Since L is tangent to these circles the orthogonal projections of their centers on the x-axis are the tangency points $Y_1 = h_1$ and $Y_2 = h_2$.

Algorithm LABMAX: Line-Segment Problem (maximization)

Input: The endpoints $A = (a_1, a_2)$ and $B = (b_1, b_2)$ of a segment AB and a line L that does not intersect AB.

Output: A point $X \in L$ for which $\theta(X)$, with respect to AB, is maximum over L.

Begin

Compute $Y_1 = h_1$ and $Y_2 = h_2$, which are the intersection points between $P_1(x)$ and $P_2(x)$. Exit with $X = Y_1$ if $\theta(Y_1) \geq \theta(Y_2)$, otherwise exit with $X = Y_2$.

End

Theorem 3.2.1: Algorithm *LABMAX* finds in $O(1)$ time a point $X \in L$ such that $\theta(X)$ is maximum with respect to the segment AB.

Note that this problem could be solved using optimization tools from Calculus when the analytic expression for $\theta(x)$ is obtained. Thus, if the given line L is the x-axis, the aperture angle function is defined as $\theta(x) = \operatorname{acos}\frac{\mathbf{u}_1 \cdot \mathbf{u}_2}{\|\mathbf{u}_1\|\|\mathbf{u}_2\|}$, where $\mathbf{u}_1 = (x - a_1, -a_2)$ and

$\mathbf{u}_2 = (x - b_1, -b_2)$. To obtain the maximum value of such a function it is necessary to find the corresponding critical points. Such points are obtained by setting the derivative of the aperture angle function $\theta(x)$ equal to zero. The corresponding derivative is defined by the equation:

$$\frac{d}{dx}\theta = \frac{d}{dx}\left(\operatorname{acos}\frac{\mathbf{u}_1 \cdot \mathbf{u}_2}{\|\mathbf{u}_1\|\|\mathbf{u}_2\|}\right)$$

By developing this derivative we obtain the following equation:

$$\frac{d}{dx}\left(\operatorname{acos}\frac{\mathbf{u}_1 \cdot \mathbf{u}_2}{\|\mathbf{u}_1\|\|\mathbf{u}_2\|}\right) = -\frac{\|\mathbf{u}_1\|\|\mathbf{u}_2\|}{\sqrt{\|\mathbf{u}_1\|^2\|\mathbf{u}_2\|^2 - (\mathbf{u}_1 \cdot \mathbf{u}_2)^2}}\frac{d}{dx}\frac{\mathbf{u}_1 \cdot \mathbf{u}_2}{\|\mathbf{u}_1\|\|\mathbf{u}_2\|}$$

However, the derivative function is non-differentiable at the intersection point x_0 between L and the line $L(A, B)$. Since the function $\theta(x)$ is defined at x_0 and because the expression $\|\mathbf{u}_1\|^2\|\mathbf{u}_2\|^2 - (\mathbf{u}_1 \cdot \mathbf{u}_2)^2$ defined in the square root of the denominator of $\frac{d}{dx}\theta$ is not defined at x_0, it follows that the derivative function is non-differentiable at this point.

Let $m = \frac{b_2 - a_2}{b_1 - a_1}$ denote the slope of $L(A, B)$, then x_0 is defined as $x_0 = a_1 - \frac{a_2}{m}$. To show that the derivative function $\frac{d}{dx}\theta$ of $\theta(x)$ is not defined at x_0 observe that:

$$\|\mathbf{u}_1\|^2\|\mathbf{u}_2\|^2 - (\mathbf{u}_1 \cdot \mathbf{u}_2)^2 = [4a_1b_1 + \|A\|_2^2 + \|B\|_2^2 - (a_1 + b_1)^2 - 2(A \cdot B)]x^2$$

$$-2[a_1\|B\|_2^2 + b_1\|A\|_2^2 - (a_1 + b_1)(A \cdot B)]x + [\|A\|_2^2\|B\|_2^2 - (A \cdot B)] .$$

This expression can be reduced to $[(b_2 - a_2)x - (a_1b_2 - a_2b_1)]^2$, which is equal to zero for $x = x_0$. Thus, we have proved the following result.

Lemma 3.2.2: The aperture angle function is non-differentiable at $x_0 = a_1 - \frac{a_2}{m}$, where $m = \frac{b_2 - a_2}{b_1 - a_1}$.

3.3 The Segment-Segment Problem

Note that x may be constrained to move on a segment. Suppose the segment is represented by a closed interval I. Note that the endpoints of this interval define a line L containing it.

Maximization Problem

By Lemma 3.2.1, $\theta(x)$ is an upwards unimodal function over $(-\infty, D]$ and also over $[D, +\infty)$.

Observation 3.3.1: For any interval I that does not contain D, Y_1 and Y_2 the function $\theta(x)$ is strictly increasing or decreasing over I.

Thus we state the following proposition as a consequence of Observation 3.3.1.

Proposition 3.3.1: The maximum aperture angle over I with respect to AB is reached at either Y_1, Y_2, or at an endpoint of I.

Algorithm IABMAX: Segment-Segment Problem (maximization)

Input: The endpoints $A = (a_1, a_2)$ and $B = (b_1, b_2)$ of a segment AB and a closed interval $I = [c, d]$ that does not intersect AB.

Output: A point $X \in I$ for which $\theta(X) = ang(AXB)$ is maximum over I.

Begin

Step 1.- Determine the points Y_1 and Y_2 by using algorithm *LABMAX*.

Step 2.- If $Y_1 \in I$ then exit with $X = Y_1$, or if $Y_2 \in I$ then exit with $X = Y_2$.

else

exit with $X = c$ if $\theta(c) \geq \theta(d)$, otherwise exit with $X = d$.

End

Theorem 3.3.1: Algorithm *IABMAX* finds in $O(1)$ time the point $X \in I$ for which $\theta(X) = ang(AXB)$ is maximum over I.

Proof: Step 1 is correct by Theorem 3.2.1 and it is performed in $O(1)$ time. The test to determine whether I contains one of the local maxima Y_1 or Y_2 is done in constant time. Finally, the evaluation and comparison in Step 2 are done in $O(1)$ time. Then, Step 2 is done in $O(1)$ time.

Minimization Problem

If the interval I does not contain the point D then $\theta(x) = ang(AxB) \neq 0$ for all $x \in I$, and the minimization problem becomes interesting. Let Z be a point in L where $\theta(x)$ reaches its minimum value $\theta(Z) = ang(AZB)$.

As a consequence of Observation 3.3.1 we can establish the following proposition which allows us to develop an algorithm for this case.

Proposition 3.3.2: If the point x is constrained to be in a closed interval $I = [a, b]$ contained in $(-\infty, D) \cup (D, +\infty) \setminus \{Y_1, Y_2\}$ the point which is the solution for the minimization problem lies at one of the endpoints of such an interval. If one of the maxima, Y_1 or Y_2, belongs to I the solution can be reached at either of the endpoints.

Algorithm IABMIN Segment-Segment Problem (minimization)

Input: The endpoints of a segment AB and a closed interval $I = [c, d]$.

Output: A point $Z \in I$ for which $\theta(Z)$, with respect to AB, is minimum over I.

Begin

If $\theta(c) \geq \theta(d)$ then exit with $Z = d$, otherwise exit with $Z = c$.

End

Theorem 3.3.2: Algorithm *IABMIN* finds in $O(1)$ time a point $Z \in I$ such that $\theta(Z)$ is minimum with respect to the segment AB.

3.4 The Segment-Polygon Problem

Problem Statement: Given an m-convex polygon P and a segment AB that does not intersect P, find a point X in P such that $ang(AXB) = \theta(X)$ is the maximum value of the aperture angle function $\theta(x) = ang(AxB)$, and a point Z in P such that $ang(AZB) = \theta(Z)$ is the minimum value.

Assume that the m vertices of P are stored in an array in counterclockwise order.

Maximization Problem

For our purpose we will suppose that the line $L(A, B)$ passing through AB does not intersect $int(P)$. Because, if $L(A, B)$ intersects $int(P)$, this line divides P into two convex polygons P_1 and P_2 such that $L(A, B)$ does not intersect $int(P_1)$ or $int(P_2)$. Then, this case is solved by solving two problems for which $L(A, B)$ does not intersect the interior of a convex polygon. Let $X_1 \in P_1$ be the point that determines the maximum aperture angle $ang(AX_1B) = \theta(X_1)$ in P_1. Let $X_2 \in P_2$ be the point such that $ang(AX_2B) = \theta(X_2)$ defines the maximum aperture angle in P_2. Thus, the solution when $L(A, B)$ intersects $int(P)$ is determined by a point X such that $\theta(X) = \max\{\theta(X_1), \theta(X_2)\}$. We characterize in the following lemma the subset of P which contains X.

Lemma 3.4.1: A point $X \in P$ where the aperture angle function reaches the maximum value lies on the chain $CSS(P)$.

Proof: (by contradiction) Let X be the point that maximizes the aperture angle and suppose that it is not contained in $CSS(P)$. Let $\theta(X)$ denote this maximum angle. Let the supporting rays from X be denoted by $ray(X, A)$ and $ray(X, B)$, referring to Figure 3.4.1. First we observe that $CSS(P)$ must intersect the $cone(X)$ that defines $\theta(X)$. If it does not, then the rays must both intersect the same line of support in the unbounded portion R_p. But then one ray cannot be a line of support of AB, a contradiction. In particular, $CSS(P)$ must intersect $triangle(AXB) \subset cone(X)$. Let y be a point in the intersection of $triangle(AXB)$ and $CSS(P)$. Then the circle $C(A, X, B)$ contains in its interior the point $y \in CSS(P)$. By Observation 2.3.3, $\theta(y) > \theta(X)$, a contradiction. Q.E.D.

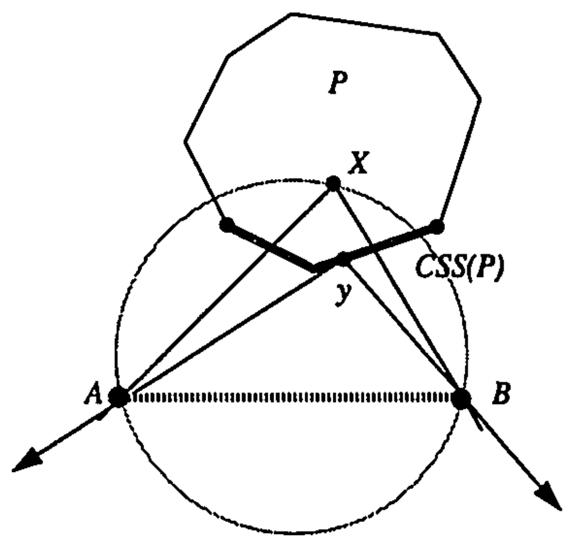

Fig. 3.4.1

The following result shows that the function $\theta(x)$ has a unique point where the maximum is reached.

Lemma 3.4.2: The maximum aperture angle is realized at a unique point $X \in CSS(P)$.

Proof: Suppose that the maximum is reached at two different points X_1 and X_2, which are determined by the circle C which contains A and B and is tangent to P. Because P is a convex polygon, the segment $X_1X_2 \in P$ and it is a chord of C. Therefore there exists a point Y in the interior of the segment X_1X_2 such that $\theta(Y) > \theta(X)$ (by Observation 2.3.3), a contradiction. Q.E.D.

Lemmas 3.4.1 and 3.4.2 establish the existence of a global maximum over $CSS(P)$. However, this 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 $CSS(P)$ by using Observation 3.3.1 in each edge (or interval determined by an edge) of this chain.

Lemma 3.4.3: The function $\theta(x) = ang(AxB)$ is an *upwards unimodal function* over $CSS(P)$.

Proof: Let C be the circle through A and B that is tangent to P and let X be the point of tangency. By Lemma 3.4.2, X is on the chain $CSS(P)$. Clearly X can be a vertex p_s or an interior point of an edge $p_{s-1}\, p_s$ of $CSS(P)$. For each edge $p_i\, p_{i+1} \in CSS(P)$ that does not contain X in its interior let $L(p_i, p_{i+1})$ be the line passing through $p_i\, p_{i+1}$. Then, the aperture angle $\theta(x)$ defined over the edge $p_i\, p_{i+1}$ with respect to AB is equal to ang (AxB) over the interval $I = [p_i, p_{i+1}] \subset L(p_i, p_{i+1})$. Without loss of generality suppose that the intersection point D between $L(A, B)$ and $L(p_i, p_{i+1})$ is to the left of the interval I and that A lies between B and D. If D is to the right of I the analysis is analogous.

Note that for each edge $p_i\, p_{i+1}$ that does not contain X the corresponding line $L(p_i, p_{i+1})$ intersects the circle C at two different points z_1 and z_2 lying in the exterior of $I=[p_i, p_{i+1}]$. When traversing the circle C in a counterclockwise direction from the point A, we define the order $(A\text{-}\, z_1\text{-}\, z_2\text{-}\, B)$. Thus, there are two possible arrangements of points over $L(p_i, p_{i+1})$.

1) $(D\text{-}\, p_{i+1}\text{-}\, p_i\text{-}\, z_1\text{-}\, z_2)$ which occurs if $I = [p_i, p_{i+1}]$ is ahead of $X \in CSS(P)$ (refer to Fig. 3.4.2.a).

2) $(D\text{-}\, z_1\text{-}\, z_2\text{-}\, p_{i+1}\text{-}\, p_i)$ holds if $I = [p_i, p_{i+1}]$ is behind $X \in CSS(P)$ (refer to Fig. 3.4.2.b).

By Corollary 3.3.1 the maximum over the interval $[z_1, z_2]$ is obtained by a point $Y \in [z_1, z_2]$. Since $(p_i, p_{i+1}) \cap (z_1, z_2) = \emptyset$ then $Y \notin I$. By Observation 3.3.1 it follows that:

The function $\theta(x)$ is strictly increasing over $I = [p_i, p_{i+1}]$ if $(D\text{-}\, p_{i+1}\text{-}\, p_i\text{-}\, z_1\text{-}\, z_2)$ holds. If $(D\text{-}\, z_1\text{-}\, z_2\text{-}\, p_{i+1}\text{-}\, p_i)$ is satisfied then $\theta(x)$ is strictly decreasing over I.

Thus, X is a vertex p_k in the chain $CSS(P) = (p_1, p_2,\ldots, p_k) \cup (p_k, p_{k+1},\ldots, p_r)$ if $\theta(x)$ is strictly decreasing in all the edges contained in the chain $(p_1, p_2,\ldots, p_k)$, and $\theta(x)$ is strictly increasing in all the edges contained in the corresponding chain $(p_k, p_{ks+1},\ldots, p_r)$. The respective orders must be satisfied in all edges of each subchain of $CSS(P)$.

In particular, (D- z_1- $z_2 = p_k$ - p_{k-1}) is satisfied in the edge $p_{k-1}\, p_k \subset (p_1, p_2,\ldots, p_k)$ and (D- p_{k+1}- $p_k = z_1$ - z_2) is valid for $p_k\, p_{k+1} \subset (p_k, p_{k+1},\ldots, p_r)$. Then, the function $\theta(x)$ is unimodal.

The tangency point X is in the interior of an edge $p_{k-1}\, p_k$ if $\theta(x)$ is strictly increasing in each edge contained in $(p_k, p_{k+1},\ldots, p_r)$ and it is strictly decreasing in all the edges contained in the chain $(p_1, p_2,\ldots, p_{k-1})$. Then, the function $\theta(x)$ is unimodal. Q.E.D.

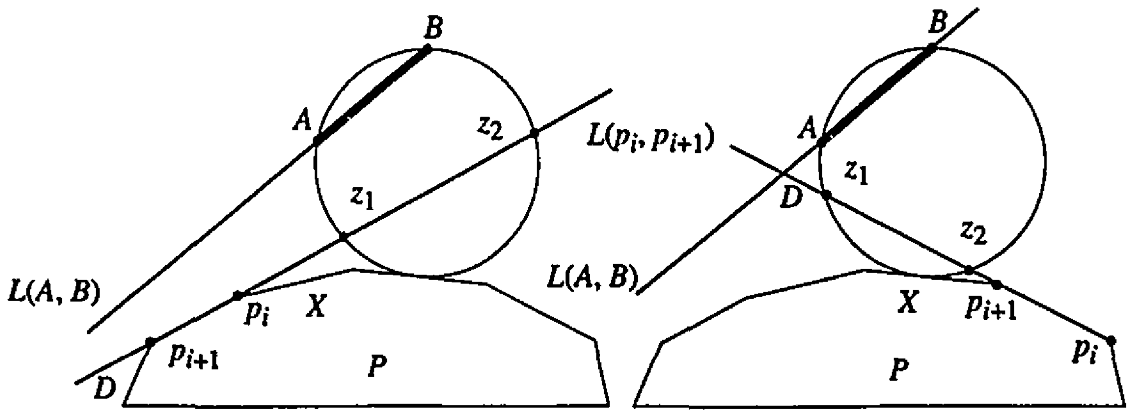

Fig. 3.4.2.a Fig. 3.4.2.b

Algorithm PABMAX

Input: An m-convex polygon P and a segment AB such that $L(A, B)$ does not meet P.

Output: A point $X \in P$ for which $\theta(X) = ang(AXB)$ is the maximum value of $\theta(x)$ over P.

Begin

Step 1.- Determine the chain $CSS(P)$.

Step 2.- Determine the point X where a circle C through A and B is tangent to P, using binary search over $CSS(P)$.

Let $\{e_1, e_2,\ldots, e_k\}$ be the sequence of edges that defines $CSS(P)$ in counterclockwise order.

Solve the segment line problem for AB and the line $L(e_{k/2})$. Denote by X_1 the solution to this problem.

If $X_1 \in e_{k/2}$ then exit with $X = X_1$

else

(Test whether $\theta(x)$ is increasing or decreasing over the interval defined by $e_{k/2}$)

If $\theta(x)$ is increasing then search X in the chain $\{e_{k/2-1},\ldots, e_1\}$ using binary search

else

search X in the chain $\{e_{k/2+1},\ldots, e_k\}$ using binary search.

End

Theorem 3.4.1: Algorithm *PABMAX* finds in $O(\log m)$ time a point $X \in P$, such that $\theta(X)$ is the maximum value of $\theta(x)$ with respect to the segment AB.

Proof: Since the angle function from a point outside of a convex polygon to a point that travels along the boundary of the polygon is a bimodal function [CD87], the support vertices of $CSS(P)$ are determined using binary search on $bd(P)$ from each of the endpoints of AB in $O(\log m)$ time. Therefore Step 1 is done in $O(\log m)$ time. By Lemma 3.4.1 we may use binary search on $CSS(P)$ to find the point X where the aperture angle is maximum. Therefore the complexity of Step 2 is bounded by $O(\log m)$ time. Q.E.D.

Minimization Problem

We now turn our attention to the minimization problem. We also assume that $L(A, B)$ does not intersect P. If it does the solution is reached at the segment contained in P obtained by intersecting $L(A, B)$ and P.

We immediately conclude from Lemma 3.4.1 that the point Z, where $\theta(x)$ attains its minimum value $\theta(Z)$, must be on P - $CSS(P)$. However, Z can be constrained to move in the chain $ECT(P)^c$.

Lemma 3.4.4: Any point Z in P where the aperture angle reaches the minimum value lies on the chain $ECT(P)^c$.

Proof: (by contradiction) By Lemma 3.4.1 the point Z where the minimum value of $\theta(x)$ is attained has to belong to the set *P-CSS(P)*. Let Z be a point in the interior of P that defines the minimum aperture angle $ang(A\ Z\ B)$, refer to Fig. 3.4.3. Let $cone(Z)$ be the cone that defines the aperture angle $ang(A\ Z\ B)$. Then, by an argument similar to the one used in the proof of Lemma 3.4.1, the $cone(Z)^c$ with apex at Z determined by the rays (Z, A) and (Z, B) in opposite direction intersects the chain $ECT(P)^c$. Let y be a point in the intersection of $cone(Z)^c$ and $ECT(P)^c$. To define $cone(y)$ translate $ray(Z, A)$ and $ray(Z, B)$ in the direction of $ECT(P)^c$ until they intersect at y. By construction, $cone(y)$ defines an angle whose value is equal to $ang(A\ Z\ B)$ and it contains AB. However, since the rays defining $cone(y)$ are not tangents to AB then the cone does not define the aperture angle for y. In order to define $\theta(y)$ the rays that define $cone(y)$ must be rotated in the direction of the endpoints of AB. Therefore $\theta(y) < ang(A\ Z\ B)$, a contradiction. Q.E.D.

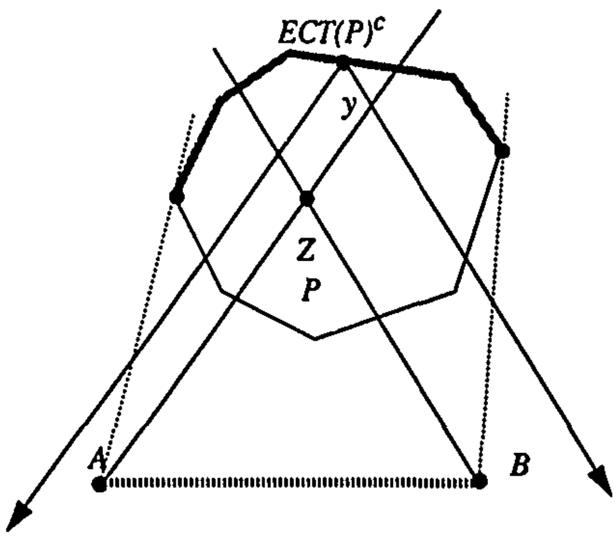

Fig. 3.4.3

For finding the minimum aperture angle $ang(A\ Z\ B)$ we have to evaluate $\theta(x)$ at the endpoints of each vertex in $ECT(P)^c$. Thus, we obtain one candidate as the minimum for each vertex and we select the minimum over all those local minima.

Algorithm PABMIN

Input: A segment AB and a convex polygon P that does not intersect AB.

Output: A point $Z \in P$ for which $\theta(Z) = ang(A\ Z\ B)$ is minimum over P.

Begin

Step 1.- Determine the chain $ECT(P)^c$.

Step 2.- For each vertex Z_i of $ECT(P)^c$ determine $\theta(Z_i)$.

Step 3.- Exit with $Z = Z_i$ such that $\theta(Z_i) \leq \theta(Z_j)$ for all $j \neq i$.

End

Theorem 3.4.2: Algorithm *PABMIN* finds in $O(m)$ time a point $Z \in P$ such that $\theta(Z)$ is the minimum value of $\theta(x)$ with respect to the segment AB.

Proof: The support vertices of P that determine $ECT(P)^c$ are determined in $O(\log m)$ time by performing binary search on $bd(P)$, as was pointed out in Theorem 3.4.1 [CD87], from each endpoint of AB. Then, Step 1 is done in $O(\log m)$ time. The evaluation of $\theta(x)$ at each point Z_i is done in constant time. Hence, the evaluation of $\theta(x)$ for all the vertices in the chain $ECT(P)^c$ is done in $O(m)$ time. Finding the maximum over m values is done in $O(m)$ time. Therefore to find $Z \in P$ for which $\theta(Z)$, with respect to AB, is the minimum value of $\theta(x)$ over P is done in $O(m)$ time. Q.E.D.

3.5 The Polygon-Line Problem

We now take a final step towards the general problem and study a simplification referred to as the *Polygon-Line Problem*, where the segment AB is replaced by a convex polygon Q, and the convex polygon P is replaced by a line L. It is assumed that the line does not intersect the polygon.

Problem Statement: Given an n-convex polygon Q and a line L, find a point $X \in L$ such that the aperture angle $\theta(X)$ is the maximum value of $\theta(x)$ with respect to Q.

We assume that $Q = [q_1, q_2,..., q_n]$ is represented by an array with the vertices in counterclockwise order and that the vertices of Q are in general position, i.e., no three vertices lie on a line and no four vertices lie on a circle. To simplify the notation, we also assume that no edge of Q is parallel to the line L. Without loss of generality we assume L is the real axis x; let q_h be the vertex with the highest y coordinate and let q_l be vertex with the lowest y coordinate of Q. Then the polygon Q is the union of a left chain $Q_a=\{q_h, q_{h+1},..., q_l\}$ and a right chain $Q_b=\{q_l, q_{l+1},..., q_h\}$. A partition of L is obtained as follows: extend each edge $q_i\, q_{i+1}$ of Q_a until it intersects L at a point a_i and extend each edge $q_j\, q_{j+1}$ of Q_b until it intersects L at a point b_j. 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 is the link between the *segment-line problem* and the *polygon-line problem.*

Lemma 3.5.1: For each interval $I_k = [r_k, r_{k+1}]$ in the partition, there are two vertices $\alpha(k) \in Q_a$ and $\beta(k) \in Q_b$ such that for each point $x \in I_k$, the aperture angle $\theta(x)$ with respect to Q is given by $ang(\alpha(k)\, x\, \beta(k))$.

Proof: Since q_h and q_l are the highest and lowest points of Q respectively, then for all $x \in I_0 = (-\infty, r_1]$, $\theta(x)$ is given by $ang(q_h\, x\, q_l)$ and therefore $\alpha(0) = q_h$ and $\beta(0) = q_l$ (see Fig. 3.5.1.a).

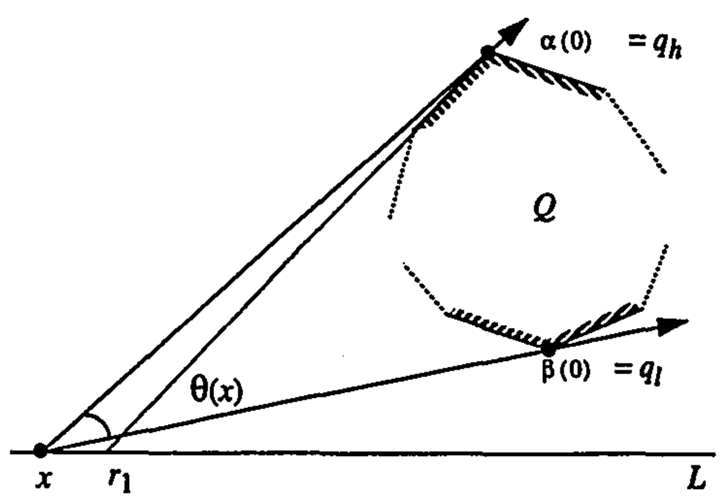

Fig. 3.5.1.a

Once $\alpha(t)$ and $\beta(t)$ have been determined for $t = 0, 1,..., k$, then if $r_{k+1} = a_i$ (i.e. r_{k+1} is the intersection point of L with the extension of the edge $q_i\,q_{i+1}$ of Q_a) then $\alpha(k+1) = q_{i+1}$ where $\alpha(k) = q_i$, and $\beta(k) = \beta(k+1)$ (see Fig. 3.5.1.b).

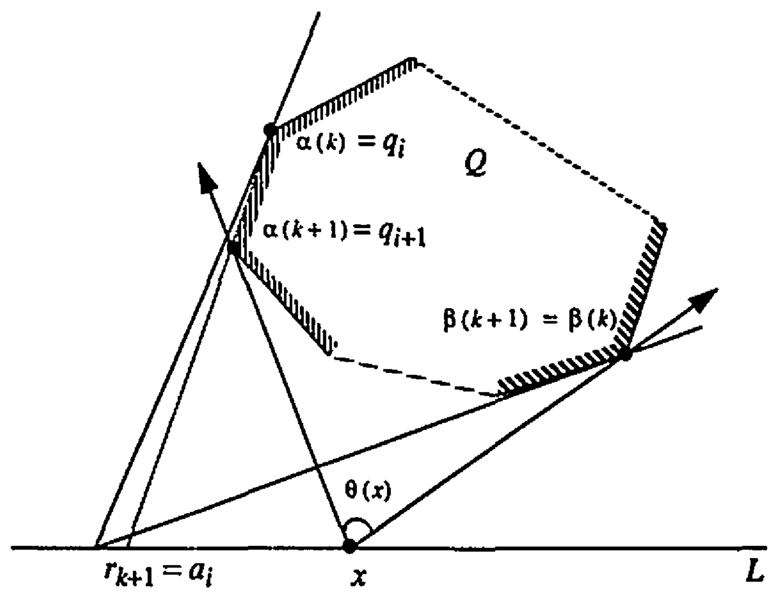

Fig. 3.5.1.b

If $r_{k+1} = b_j$ (i.e. r_{k+1} is the intersection point of L with the extension of the edge $q_j\,q_{j+1}$ of Q_b), then $\alpha(k) = \alpha(k+1)$ and $\beta(k+1) = q_{j+1}$ where $\beta(k) = q_j$ (see Fig. 3.5.1.c).

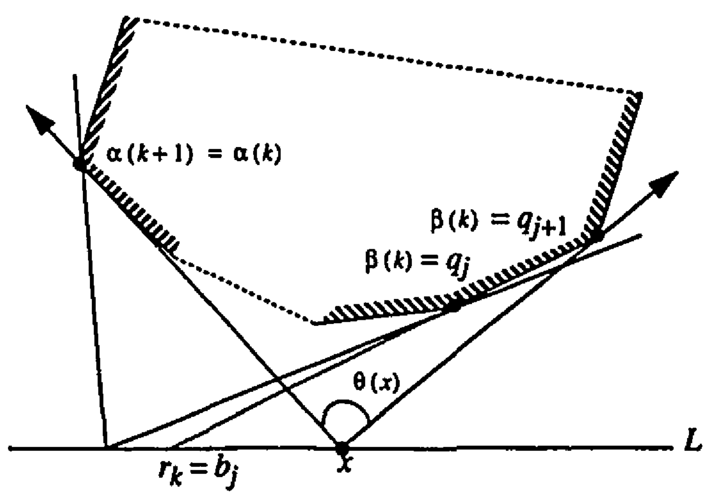

Fig. 3.5.1.c

By induction $\alpha(k)$ and $\beta(k)$ can be determined for $k = 0, 1, \ldots, n$. Q.E.D.

As a consequence of Lemma 3.5.1 the aperture angle function $\theta(x)$ with respect to Q is piece-wise defined over L and for each interval I_k, $\theta(x)$ coincides with the aperture angle function with respect to the corresponding segment $\alpha(k)\beta(k)$. For $k = 1, 2, \ldots, n$ we say that $\alpha(k)\beta(k)$ is the diagonal of Q associated with the interval I_k and it is denoted by d_k.

Maximization Problem

Using Proposition 3.3.1 we can determine the points $X_0, X_1, X_2, \ldots, X_n$ where $\theta(x)$ reaches their set of local maxima over $I_0, I_1, I_2, \ldots, I_n$ respectively, and then the maximum aperture angle with respect to Q is reached at a point X such that $\theta(X) = \max\{\theta(X_0), \theta(X_1), \theta(X_2), \ldots, \theta(X_n)\}$.

Algorithm LQMAX: Polygon-Line Problem

Input: An n-convex polygon Q 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 $I_0, I_1, \ldots, I_n$.

Step 2.- For each interval I_k find the diagonal $d_k = \alpha(k)\beta(k)$ such that the aperture angle functions with respect to Q and d_k coincide over I_k.

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

Step 4.- Exit with X_i, where X_i is such that $\theta(X_i) \geq \theta(X_j)$ for all $j = 0, 1, \ldots, n$.

End

Theorem 3.5.1: Algorithm *LQMAX* finds in $O(n)$ time a point $X \in L$ such that $\theta(X)$ is the maximum value of $\theta(x)$ with respect to Q.

Proof: Step 1 and Step 2 require $O(n)$ time by Lemma 3.5.1. To compute each point X_k in step three requires $O(1)$ time using algorithm *LABMAX* and Proposition 3.3.1.

Finding $X_0, X_1,..., X_n$ is done in $O(n)$ time. Clearly Step 4 is also done in $O(n)$ time. Thus the entire algorithm requires $O(n)$ time to find a point X in L for which the aperture angle $\theta(X)$, with respect to Q, is maximum. Q.E.D.

Note that the minimization problem is not of interest for this case, since $\theta(x)$ asymptotically approaches to zero when x moves along L towards $(-\infty)$ or $(+\infty)$.

3.6 The Case of Two Convex Polygons: Geometric Considerations

Problem Statement: Given two disjoint convex polygons P and Q in the plane with m and n vertices respectively, find a point $X \in P$ such that $\theta(X)$ is the maximum value of the aperture angle function with respect to the polygon Q, and a point $Z \in P$ such that $\theta(Z)$ is the minimum value.

In order to simplify the analysis, 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 counter-clockwise order. In this section we extend Lemma 3.4.1 and Lemma 3.4.4 to the problem of maximizing and minimizing the aperture angle with respect to Q when $x \in P$. We also give an analysis for computing the maximum aperture angle that allows us to develop, in the next section, an algorithm with two different complexities.

Maximization Problem

Lemma 3.6.1: The points in P for which the aperture angle is a maximum lie on the boundary of P.

Proof: (by contradiction) Let Y be a point in the interior of P, and let q_i and q_j be the two vertices of Q such that $\theta(Y) = ang(q_i\, Y\, q_j)$. By the Jordan curve theorem (see for instance [CR41]) the rays $ray(Y, q_i)$ and $ray(Y, q_j)$ intersect the boundary of P, and by convexity they intersect it only once. Let x be the point where the segment Yq_j intersects $bd(P)$ (refer to Fig. 3.6.1). Since the interior angles of triangle yxq_i sum 180 degrees and $ang(y\, x\, q_i) + ang(q_i\, x\, q_j) = 180$, it follows that $ang(q_i\, x\, q_j) = ang(q_i\, Y\, q_j) + ang(y\, q_i\, x) > \theta(Y)$. Since $ang(q_i\, x\, q_j)$ is smaller than $ang(q_k\, x\, q_j) = \theta(x)$, then $\theta(x) > \theta(Y)$. Thus there exists a point x in $bd(P)$ for which the aperture angle is larger than $\theta(Y)$, a contradiction. Q.E.D.

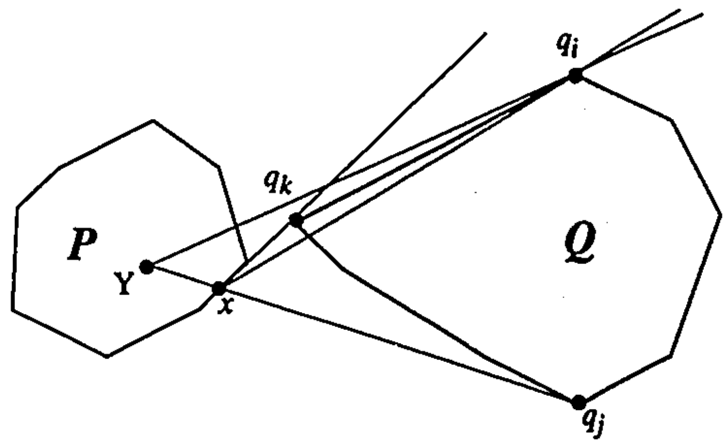

Fig 3.6.1

By Lemma 3.6.1 we constrain the space of possible solutions to the $bd(P)$. We can actually strengthen this result by showing that the solution lies in a specific section of the $bd(P)$.

Let the critical separating lines of support to P and Q be tangent at $\{p_i, q_t\}$ and $\{p_j, q_k\}$ respectively. These lines separate the boundaries of P and Q into four chains (refer to Sec. 2.2). The chains for P are $CSS(P)$ and $CSS(P)^c$. The respective chains $CSS(Q)$ and $CSS(Q)^c$ for Q are defined in the same way.

Lemma 3.6.2: Any point $x \in bd(P)$ where the aperture angle reaches the maximum value lies on the chain $CSS(P)$.

Proof: It is analogous to the proof of Lemma 3.4.1.

Let the common tangents to P and Q be tangents at $\{p_r, q_i\}$ and $\{p_s, q_j\}$ respectively. Then, the chain $ECT(P)$ is defined as $(p_r, p_{r+1},..., p_i, p_{i+1},..., p_j, p_{j+1},..., p_s)$. The chains $ECT(Q) = (q_i, q_{i+1},..., q_k, q_{k+1},..., q_t, q_{t+1},..., q_j)$ and $ECT(Q)^c$ are similarly defined.

Having characterized the chain of P that contains the points where the maximum value can be reached we can define the partition over $bd(P)$, as we did in the *polygon-line* problem. Let $CSS(P)$, $CSS(Q)$, $ECT(P)$ and $ECT(Q)$ be defined as above. For each edge e contained in the chain $ECT(Q)$, extend e until it intersects P or extends indefinitely. These intersection points determine a partition of $bd(P)$. For the maximization problem, we focus our attention on the partition defined on $CSS(P)$.

Observe that the *external common tangents* of P and Q define two tangent vertices q_i and q_j in Q which are the extreme points of $ECT(Q) = (q_i, \ldots, q_j)$. Analogously, the *critical separating lines of support* of P and Q define two support vertices in Q denoted by q_k and q_t (refer to Fig. 3.6.2). These points are the endpoints of $CSS(Q) = (q_k, q_{k+1}, \ldots, q_t)$. Then the chains $ECT(Q)$ and $CSS(Q)$ are equal, or $CSS(Q)$ is contained in $ECT(Q)$. If $CSS(Q) = ECT(Q)$, there is not a partition on the boundary of P and the problem is reduced to the *segment-polygon* case, where the segment is determined by $A = q_i$ and $B = q_j$. If $CSS(Q) \subset ECT(Q)$ then the chain $ECT(Q)$ can be expressed as the union of the chains $(q_i, q_{i+1}, \ldots, q_k)$, $CSS(Q)$, and $(q_t, q_{t+1}, \ldots, q_j)$. Let Q_a and Q_b be defined as $(q_i, q_{i+1}, \ldots, q_k)$ and $(q_t, q_{t+1}, \ldots, q_j)$ respectively. The chain Q_a is referred to as the left chain of Q and Q_b as the corresponding right chain of Q. Thus, the chain $ECT(Q)$ has been divided into three chains Q_a, Q_b and $CSS(Q)$. For Q_a and Q_b the corresponding extended edges intersect P. The corresponding extended edges of $ECT(Q)$ do not intersect P.

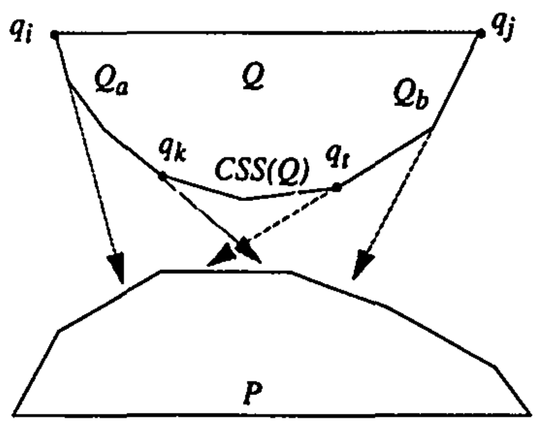

Fig. 3.6.2

Since the extended edges of Q_a or Q_b can intersect the boundary of P in some points that are on $ECT(P)$ but not on $CSS(P)$ we need to use $ECT(P)$ and discard all those intersection points that do not belong to $CSS(P)$. We denote by a_k the intersection of the k-th edge in Q_a with $CSS(P)$ and by A the ordered set of intersection points a_k's. Analogously, B is the ordered set of intersection points between the extended edges in Q_b and $CSS(P)$. The partition R is determined by merging the two ordered sets A and B. It is formed by polygonal chains R_k which are determined by two consecutive intersection points r_{k-1} and r_k

in R. Note that the polygonal chains are convex chains with respect to Q. For each of them the aperture angle is defined by the same diagonal. Thus, Lemma 3.5.1 can be extended for the polygonal regions R_k.

Lemma 3.6.3: For each convex chain $R_k \subseteq CSS(P)$ in the partition of $CSS(P)$, there are two vertices $\alpha(k) \in Q_a$ and $\beta(k) \in Q_b$ such that for each point $x \in R_k$, the aperture angle $\theta(x)$ with respect to Q is given by $ang(\alpha(k)\ x\ \beta(k))$.

As a consequence of Lemma 3.6.3 the aperture angle function $\theta(x)$ with respect to Q is piece-wise defined over the chain $CSS(P)$. For each convex chain R_k, the function $\theta(x)$ is determined by $ang(\alpha(k)\ x\ \beta(k))$. Then the segment $d_k = \alpha(k)\beta(k)$ is the diagonal of Q that defines $\theta(x)$ for $x \in R_k$. For each k $(0 \le k \le n)$ we say that $d_k = \alpha(k)\beta(k)$ is the diagonal of Q associated with the convex chain R_k. Thus, the problem is reduced to finding the maximum aperture angle for each k with respect to a given segment d_k, when x is allowed to move on a convex chain R_k. By Lemma 3.4.2, for each chain there exists a unique point where the maximum aperture angle is reached with respect to the segment d_k. Let $X_k \in R_k$ denote this point. Lemma 3.4.3 implies that $\theta(x)$ is upwards unimodal for each chain R_k. Then $\theta(x)$ is a K-modal function over $CSS(P)$.

Proposition 3.6.1: The number K of convex chains contained in $CSS(P)$ is $O(n)$.

Minimization Problem

We characterize the subset of points of the boundary of P where the minimum value $\theta(x)$ can be attained.

Lemma 3.6.4: Any point x in P where the aperture angle reaches its minimum value lies on the chain $ECT(P)^c$.

Proof: It is analogous to the proof of Lemma 3.4.4.

Employing the same method used for the maximization problem, we determine a partition of the chain that contains the points where the minimum can be attained, i.e., the chain $ECT(P)^c$. This partition is obtained by extending the edges of Q_a and Q_b until they intersect $ECT(P)^c$. In this way, the partition obtained contains chains R_k, which are concave with respect to Q. In order to use the results and algorithms developed previously we state the following lemma.

Lemma 3.6.5: For each polygonal $R_k \subseteq ECT(P)^c$ in the partition of $ECT(P)^c$, there are two vertices $\alpha(k) \in Q_a$ and $\beta(k) \in Q_b$ such that for each point $x \in R_k$ the aperture angle $\theta(x)$ with respect to Q is given by $ang(\alpha(k)\, x\, \beta(k))$.

Proof: It is analogous to the proof of Lemma 3.5.1.

Having stated this lemma we are in a position to take advantage of the results developed in Section 3.4 for the minimization problem. To solve the problem we use Algorithm *PABMIN* for each concave chain R_k. Observe that when Algorithm *PABMIN* is used for each concave chain R_k, the endpoints that define it become vertices. Then, the solution consists in the evaluation of $\theta(x)$ at the vertices of $ECT(P)^c$ and at the intersection points that define the partition. Therefore the solution is given by a point $Z \in P$ such that $\theta(Z) = \min \{\theta(Z_K)\}$ $(0 \leq K \leq n+m)$, where Z_K is a vertex of $e_K \in ECT(P)^c$ or an endpoint of a chain R_k.

3.7 The Case of Two Convex Polygons: Algorithms

In this section we describe the algorithms for solving the general problems of finding the maximum and minimum aperture angles. The object that must be kept in the field of view is an n-convex polygon Q, and the region where the camera is allowed to roam is an m-convex polygon P. The first algorithm to be described corresponds to the maximization problem.

Algorithm QPMAX: Polygon-polygon (maximization)

Input: A convex polygon Q with n vertices and a convex polygon P with m vertices that does not intersect Q.

Output: A point X in P for which the aperture angle $\theta(X)$, with respect to Q, is maximum.

Begin

Step 1.- Find the partititon of $CSS(P)$ into chains $R_0, R_1, \ldots, R_n$.

Step 2.- For each convex chain R_k find the diagonal $d_k = \alpha(k)\beta(k)$ such that the aperture angle function with respect to Q and d_k coincide over R_k.

Step 3.- For each chain R_k find $X_k \in R_k$ such that the aperture angle with respect to d_k is maximum over R_k, using Algorithm *PABMAX*.

Step 4.- Exit with $X = X_i$, where X is such that $\theta(X) \geq \theta(X_j)$ for all j $(0 \leq j \leq n+m)$.

End

Theorem 3.7.1: Algorithm *QPMAX* finds a point $X \in P$, such that $\theta(X)$ is maximum with respect to Q in $O(n+m)$ time using *Rotating Calipers* [To83] in Step 1 or in $O(n \log m)$ time using Chazelle and Dobkin's Algorithm [CD87], also in Step 1.

Because the complexity depends on the time complexity used when partitioning the chain $CSS(P)$ into chains $R_0, R_1,..., R_n$, we begin by analyzing Step 1. This step can be performed by using two different methods (as is pointed out in Theorem 3.7.1). We first describe the method based on *Rotating Calipers*, whose complexity is $O(n+m)$ time.

Algorithm P1: to Find the Partition of P in O(n+m)

Input: Two disjoint convex polygons Q and P, with n and m vertices respectively.

Output: A partition of the chain $CSS(P)$ into convex chains $R_0, R_1,..., R_M$.

Begin

Step 1.- Find the chains $CSS(P)$, $CSS(Q)$, $ECT(P)$, $ECT(Q)$, Q_a and Q_b.

Step 2.- For the first edge in Q_a, find the intersection point a_1 and detect which edge in $ECT(P)$ contains a_1.

Step 3.- While $Q_a \neq \varnothing$ advance one edge on the chain Q_a and extend it.

Step 4.- If the extended edge intersects the current edge in $ECT(P)$ then go to Step 5.

else

While $ECT(P) \neq \varnothing$ advance one edge in $ECT(P)$.

Step 5.- Define A as the set of intersection points a_k such that a_k belongs to $CSS(P)$.

Step 6.- Repeat steps from 3 to 5 for the chain Q_b. Define B as the set of intersection points b_k such that b_k belongs to $CSS(P)$.

Step 7.- Merge A and B to obtain the partition R of $CSS(P)$. It is defined as $\{R_1, R_2, \ldots, R_K\}$ $(K \leq n)$. Exit with R.

End

Lemma 3.7.1: Algorithm $P1$ finds in $O(n+m)$ time a partition of $CSS(P)$ in convex chains which are convex with respect to Q.

Proof: Using rotating calipers [To83] we find the tangent points and the support points that define the chains required in Step 1 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. Step 2 is done in $O(\log m)$ time using Chazelle and Dobkin [CD87]. For finding each intersection point a_k in Step 3, $O(1)$ time is used. Then, all of the intersection points can be determined in $O(n)$ time. Advancing over all the chain $ECT(P)$ takes $O(m)$ time. Clearly the definition of the set A is done in $O(n)$ time. Then, steps 3, 4 and 5 are done in $O(n+m)$ time. Analogously Step 6 is done in $O(n+m)$ time. The merging of two sets to obtain one that has at most n elements is done in $O(n)$ time, then Step 7 is done in $O(n)$ time. Therefore, Algorithm $P1$ finds the partition of $CSS(P)$ in $O(n+m)$ time. Q.E.D.

To obtain the partition of $CSS(P)$ with time complexity $O(n \log m)$ observe that for each edge $q_j q_{j+1} \in ECT(Q)^c$ the line $L(q_j, q_{j+1})$ does not intersect $bd(P)$, and $det(q_j, q_{j+1}, p)$ has positive sign for each point $p \in P$. For each edge $q_j q_{j+1} \in CSS(Q)$ the line $L(q_j, q_{j+1})$ does not intersects $bd(P)$, and $det(q_j, q_{j+1}, p)$ has negative sign for each point $p \in P$. If an edge $q_j q_{j+1} \in Q_a$ is extended, the semiline from q_j in the direction of q_{j+1} intersects $bd(P)$ at two points $a_k \in ECT(P)$ and $z_k \in ECT(Q)^c$, These two points together with q_j and q_{j+1} are ordered as $(q_j\text{-}q_{j+1}\text{-}a_k\text{-}z_k)$. Similarly, if an edge $q_j q_{j+1} \in Q_b$ is extended, the semiline from q_{j+1} in the direction of q_j intersects $bd(P)$ at two points $b_k \in ECT(P)$ and $z_k \in ECT(Q)^c$, that together with q_j and $q_{j+1}$are ordered as $(z_k\text{-}b_k\text{-}q_{j+1}\text{-}q_j)$. These properties, which are satisfied by the edges of Q define other characterizations of the chains Q_a, $CSS(Q)$ and Q_b. Thus we do not need to know a priori the tangent vertices and support vertices of Q for decompose $bd(Q)$ into $ECT(Q)^c$, Q_a, $CSS(Q)$ and Q_b.

Algorithm P2: to Find the Partition in $O(n \log m)$ time.

Input: Two convex polygons Q and P, with n and m vertices respectively.

Output: A partition of the chain $CSS(P)$ into convex chains $R_0, R_1, \ldots, R_K$.

Begin

Step 1.- Extend each edge of Q to obtain:

1.1 The sets A and B of intersection points a_k and b_k respectively.

1.2 The chains Q_a, Q_b, $CSS(Q)$ and $ECT(Q)^c$.

Step 2.- While advancing over the edges of Q find the transition points from Q_a to $CSS(Q)$ and from $CSS(Q)$ to Q_b, and use them to determine the end points of $CSS(P)$. Then discard from A and B all those intersection points that do not belong to $CSS(P)$.

Step 3.- Merge A and B to obtain the partition $R = \{R_1, R_2, \ldots, R_K\}$ $(K \leq n)$. Exit with R.

End

Lemma 3.7.2: Algorithm $P2$ finds in $O(n \log m)$ time a partition of $CSS(P)$ into chains which are convex with respect to Q.

Proof: In contrast with Algorithm $P1$, this method uses 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-1 intersections and each one is found in $O(\log m)$ time, the sets A and B are found in $O(n \log m)$ time. Using the alternate characterization discussed above, we classify the edges of Q into four groups which are the required chains in Step 1.2. Hence Step 1 is done in $O(n \log m)$ time. To obtain the chain $CSS(P)$ we use the support vertices of Q that are the transition points from one chain to another. To determine the support vertices of P in $O(\log n)$ time we use binary search. Finally, the merging of A and B is done in $O(n)$ time. Therefore, Algorithm $P2$ finds the partition in $O(n \log m)$ time. Q.E.D.

Lemma 3.7.3: Let $f(n,m)$ denote the function that counts the number of operations that are done for finding the set of local maxima over $CSS(P)$ with respect to Q. Then

$$1)\ f(n,m) \le n+m$$

and

$$2)\ f(n,m) \le n \log m/n.$$

Proof: For each convex chain R_k of $CSS(P)$ with length n_i, computing the point X_k where the maximum is reached over $CSS(P)$ with respect to d_k is done in $O(\log n_i)$ time by Theorem 3.4.1. Thus,

$$f(n,m) = \sum_{i=1}^{s} c_i log n_i$$

1) Let $c = max\ \{c_i\}$, then $f(n,m) \le c \sum_{i=1}^{s} log n_i \le c\,(n+m)$.

2) We have $n_i \le m$ since P has m vertices, therefore

$$\left(c \sum_{i=1}^{s} log n_i\right) \le c\left(\sum_{i=1}^{s} log m\right) \le cnlogm.$$

From 1) and 2) Lemma 4.3 holds. Q.E.D.

Proof of Theorem 3.7.1. Using Lemma 3.6.3, Step 2 is done in $O(n)$ time. Step 4 is also done in $O(n)$ time because to find the maximum of n elements $O(n)$ time is used.

If we use Algorithm $P1$ for Step 1, by Lemma 3.7.1 the time complexity is $O(n+m)$. By Lemma 3.7.3, Step 3 is done in $O(n+m)$ time. Therefore Algorithm $QPMAX$ finds a point $X \in P$, such that $\theta(X)$ is the maximum value of $\theta(x)$ with respect to Q in $O(n+m)$ time.

If we use Algorithm *P2*, by Lemma 3.7.2, Step 1 is performed in $O(n \log m)$ time. By Lemma 3.7.3, Step 3 is done in $O(n \log m)$ time. Therefore Algorithm *QPMAX* finds a point $X \in P$, such that $\theta(X)$ is the maximum value of $\theta(x)$ with respect to Q in $O(n \log m)$ time. Q.E.D.

Algorithm QPMIN: Polygon-Polygon Problem (minimization)

Input: A convex polygon Q with n vertices and a convex polygon P that does not intersect Q.

Output: A point Z in P for which the aperture angle $\theta(Z)$, with respect to Q, is minimum.

Begin

Step 1.- Find the partititon of $ECT(P)^c$ into chains $R_0, R_1, \ldots, R_n$.

Step 2.- For each region R_k find the diagonal $d_k = \alpha(k)\beta(k)$ such that the aperture angle function with respect to Q and d_k coincides over R_k.

Step 3.- For each chain R_k find $X_k \in R_k$ such that the aperture angle, with respect to d_k, is minimum over R_k.

Step 4.- Exit with $X = X_j$, where X is such that $\theta(X) \geq \theta(X_j)$ for all $1 \leq j \leq n+m$.

End

Theorem 3.7.2: Algorithm *QPMIN* finds a point $Z \in P$, such that $\theta(Z)$ is minimum with respect to Q in $O(n+m)$ time.

Using Algorithm *P1* we may determine the chain $ECT(P)^c$ and the partition over it in $O(n+m)$ time. Using Lemma 3.6.5 Step 2 is done in $O(n)$ time. Using Algorithm *PABMIN* for each chain R_k of length n_k, Step 3 is done in $O(n)$ time. Since there are at most $(n+m)$ local minima, Step 4 is done in $O(n+m)$ time. Therefore, Algorithm *QPMIN* finds a point $Z \in P$, such that $\theta(Z)$ is minimum with respect to Q in $O(n+m)$ time. Q.E.D.

3.8 Lower Bounds

In the previous section we described algorithms for computing the maximum aperture angle $\theta(X)$ and the minimum aperture angle $\theta(Z)$ with respect to a polygon Q. We presented two algorithms for computing $\theta(X)$. One has $O(n + m)$ time complexity and the other has $O(n \log m)$ time complexity. We also determined an algorithm for computing $\theta(Z)$ in $O(n + m)$ time. In this section we show that the complexity of computing $\theta(Z)$ is $\Omega(\max\{m, n\})$, thus establishing the optimality of our corresponding algorithm. We also show a time complexity of $\Omega(n)$ for computing $\theta(X)$. We begin by describing a construction that proves an $\Omega(n)$ time lower bound for computing $\theta(X)$. Then we describe the corresponding construction that shows that $\Omega(n)$ is a lower bound for computing $\theta(Z)$. Finally, we describe a construction proving that $\Omega(m)$ is a lower bound for computing $\theta(Z)$. The lower bounds are found by considering a linear array representation of P and Q.

For the first construction we have a polygon P whose chain $CSS(P)$ is a segment p_1p_n on the x-axis. The edge p_1p_n of P will contain all the points that determine the partition of $CSS(P)$. The corresponding polygon Q will be lying on the first quadrant of the x-y plane. Before building Q consider the family $\Im$ of lines such that its intersection point with the perpendicular lines through (0, 1) is on the x-axis. Then this family is defined by $\Im = \{(x, y): y = \alpha x - \alpha^2;\ \alpha \in \Re\}$ and its envelope is given by a parabola $E_{\Im} = \{(x, y): y = x^2 / 4\}$ (refer to Fig. 3.8.1).

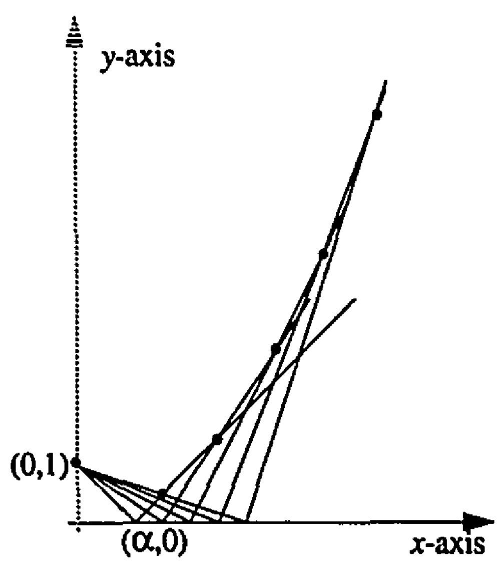

Fig 3.8.1

Using the family $\mathfrak{I}$ and its envelope $E_{\mathfrak{I}}$, we present the construction of Q simultaneously with the construction of the points that define the possible local maxima for $\theta(x)$ over the edge p_1p_n of P in an inductive way. Let $q_0 = (0, 1)$ be the first vertex of Q. To define q_1 we suppose $T_1 = (1, 0)$ is the point where the aperture angle $\theta(x)$ attains its maximum value over an interval $I_1 = [r_1, r_2]$ with respect to the diagonal $d_1 = q_0\,q_1$. Then, consider the line $L(T_1) \in \mathfrak{I}$ passing through T_1 and perpendicular to the line $L(q_0, T_1)$ which contains q_0 and T_1. By construction, $L(T_1)$ is tangent to the parabola $E_{\mathfrak{I}}$ at the point $q_1 = (2, 1)$. In this way q_1 is a tangency point between the parabola $E_{\mathfrak{I}}$ and the line $L(T_1)$ in $\mathfrak{I}$, and d_1 is defined as $q_0\,q_1$. Since d_1 is parallel to the edge p_1p_n defined on the x-axis, the maximum over p_1p_n with respect to d_1 is reached at the orthogonal projection T_1 of the midpoint of d_1 on p_1p_n. This projection of the midpoint of d_1 corresponds to T_1. By Proposition 3.3.1 the maximum over I_1 is reached at T_1 or an end point of I_1. Assume that $T_1 \in I_1 = [r_1, r_2]$. Then, $\theta(T_1)$ is the maximum value of $\theta(x)$ when $x \in I_1$, by construction its value is $\pi/2$. Let $T_2 = (2, 0)$ be the orthogonal projection of q_1 on the x-axis (see Fig. 3.8.2).

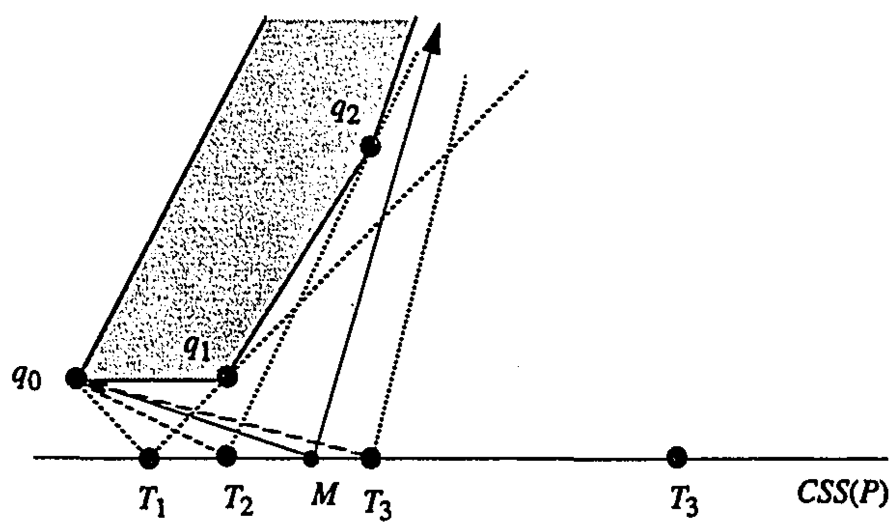

Fig 3.8.2

Let $q_{k\text{-}1}$ be the k-th vertex of Q and let $T_k = (2^{k-1}, 0)$ be the orthogonal projection of $q_{k\text{-}1}$ on the x-axis. For finding q_k, suppose that T_k is the point where the aperture angle will reach its maximum for $x \in I_k = [r_k, r_{k+1}]$ with respect to the diagonal $d_k = q_0\,q_k$. By construction, there exists a line $L(T_k) \in \Im$ that is perpendicular to the line $L(q_0, T_k)$ and that is tangent to $E_\Im$ at $q_k = (2^k, 2^{2k-2})$. Let $T_{k+1} = (2^k, 0)$ be the orthogonal projection of q_k on the x-axis.

Thus, by induction we have determined Q as $[q_0, q_1, q_2,..., q_{n-1}]$, where $q_0 = (0, 1)$ and $q_i = (2^i, 2^{2i-2})$ for $i = 1,..., n-1$. Using the above construction we obtain a set of points which define: 1) a polygon $Q = [q_0, q_1, q_2,..., q_{n-1}]$ where $q_0 = (0, 1)$ and $q_i = (2^i, 2^{2i-2})$ for $i = 1,..., n-1$, and 2) a set of points $T_1, T_2, T_3,..., T_{n-1}$ such that $T_i = (2^{i-1}, 0)$ for $i = 1,..., n-1$.

Using this information we will prove the following claims:

1) Q is convex, 2) $T_i \in I_i = [r_i, r_{i+1}]$ where the endpoints r_i and r_{i+1} are determined by using the process described in Section 3.5, and 3) T_i is a point where $\theta(x)$ attains its maximum value over I_i.

Proof of 1) Since $[q_0, q_1, q_2,..., q_{n-1}]$ lie on a parabola, for any pair of consecutive points q_i and q_{i+1} their coordinates satisfy

$$x_i = 2^{i-1} < 2^i = x_{i+1} \text{ and } y_i = 2^{2i-2} < 2^{2i} = y_{i+1}.$$

Thus, the vertices $q_1, q_2,..., q_{n-1}$ define a dominance relation that yields an order between the slopes of the corresponding edges of Q. Such an order is given by: *slope* $(q_0\, q_1) = 0 <$ *slope* $(q_1\, q_2) <$ *slope* $(q_2\, q_3) <...<$ *slope* $(q_{n-2}\, q_{n-1})$. Therefore, $[q_0, q_1, q_2,..., q_{n-1}]$ define a convex polygon.

Proof of 2) The next step is to find the partition of p_1p_n into intervals $I_0, I_1,..., I_{n-1}$. By using Lemma 3.5.1 it follows that the diagonal $d_i = q_0\, q_i$ defines the aperture angle $\theta(x)$ for each $x \in I_i$, $i = 1,..., n-1$. The aperture angle function for the interval I_0 is defined with respect to the diagonal $d_n = q_1\, q_{n-1}$.

Let $Q_a = (q_0, q_{n-1})$ and $Q_b = (q_1, q_2,..., q_{n-1})$ be the right and left chains of Q respectively. To define the partition over p_1p_n into intervals $I_0, I_1,..., I_n$, we have to extend the edges in Q_a and Q_b. First consider the extension of the edge $q_0\, q_{n-1}$ to obtain the set $A = \{a_1\}$. For each edge $q_{i-1}\, q_i \in Q_b$ consider the line $L(q_{i-1}, q_i)$ (for $i = 2,..., n-1$) to be passing through it. Then such a line is given by $\{(x,y): y - 2^{2i-2} = 3 \cdot 2^{i-3}(x - 2^i)\}$ and the corresponding intersection point with the x-axis is given by b_i. Thus $B = \{b_1, b_2,..., b_{n-2}\}$. Merge A and B to determine $R = \{r_1, r_2,..., r_{n-1}\}$ where $r_i = (2^i / 3, 0)$ for $i = 2,..., n-1$ and $r_1 = (2^{n-1} / (1 - 2^{2n-4}), 0)$. Therefore, the intervals are defined by:

$I_0 = \left(-\infty, \dfrac{2^{n-1}}{1-2^{2n-4}}\right], I_1 = \left[\dfrac{2^{n-1}}{1-2^{2n-4}}, \dfrac{2^2}{3}\right], I_i = \left[\dfrac{2^i}{3}, \dfrac{2^{i+1}}{3}\right]$ for $i = 2,..., n$-2;

and $I_{n-1} = \left[\dfrac{2^{n-1}}{3}, \infty\right)$.

Finally, we show that each point T_i previously constructed belongs to the interval I_i for $i = 1,..., n$-1. Since $2 \bullet 2^{i-1} \leq 3 \bullet 2^{i-1} \leq 2^2 \bullet 2^{i-1}$, then $2^{i-1} / 3 \leq 2^{i-1} \leq 2^{i+1} / 3$. Therefore, $T_i \in I_i$. By Lemma 3.5.1 for each I_i the diagonal $d_i = q_0 q_i$ defines the aperture angle for each $x \in I_i$, in particular for $x = T_{i-1} \in I_i$ (for $i = 1,..., n$-1).

Proof of 3) To show that $X_i = T_i$ is the maximum value $\theta(X_i)$ for $x \in I_i$ consider an arbitrary point M between T_i and T_{i+1}, i.e., $M \in [2^{i-1}, 2^i]$. Such interval is decomposed as $[2^{i-1}, 2^{i+1}/3] \cup [2^{i+1}/3, 2^i]$, suppose then $M \in [2^{i-1}, 2^{i+1}/3]$. By construction, there exists a line $L(M) \in \mathfrak{S}$ through M that is tangent to E_3 at $X_M = (x_M, y_M)$ that is also perpendicular to $L(q_0, M)$, i.e., $ang(q_0\, M\, X_M) = \pi/2$. Let $E_3^+ = \{(x,y): y \geq x^2 / 4\}$ be the parabola E_3 and its interior. Because E_3^+ is a convex set, the edge $q_{i-1}q_i$ is contained in E_3^+. Thus X_M is in the exterior of Q. Then, $ang(q_0\, M\, q_i) < ang(q_0\, M\, X_M)$. Since $ang(q_0\, M\, q_i) = \theta(M)$ and $ang(q_0\, M\, X_M) = \pi/2 = ang(q_0\, T_i\, q_i) = \theta(T_i)$, then $\theta(M) < \theta(T_i)$.

The arguments to prove that $\theta(M) < \theta(T_i)$ when $M \in [2^{i+1}/3, 2^i]$ are similar. Therefore $\theta(T_i)$ is maximum over I_i. For $i = 1,..., n$-1 denote by $X_i = T_i$ the point where the maximum value $\theta(X_i)$ is reached in I_i.

Finally, we show that this example can be modified in order to obtain a unique point X where the global maximum of $\theta(x)$ is attained and that such a point can be located in any element I_i of the partition.

Let q_i be any vertex of Q. Since $\theta(T_i) = ang(q_0\, T_i\, q_i) = \pi/2$, there exists a circle $C(q_0, T_i, q_i)$ that is tangent to p_1p_n for which $d_i = q_0\, q_i$ is a diameter. Let $ray(T_i)$ be the ray defined from T_i in direction of q_i. Rotate this ray in a clockwise direction, with T_i as the center of rotation, by an ε-positive angle. Define q_i^* as $ray(T_i) \cap C(q_0, T_i, q_i)$ (refer to Fig. 3.8.3, pag. 51). Because $C(q_0, X_i, q_i) = C(q_0, T_i, q_i^*)$ remains tangent to p_1p_n, it still defines a point $X_i = T_i$ where the maximum $\theta(X_i)$ is reached for the interval I_i with respect to the new diagonal $d_i^* = q_0\, q_i^*$. However, $d_i^* = q_0\, q_i^*$ is not a diameter of $C(q_0, T_i, q_i^*)$. It is a chord of $C(q_0, T_i, q_i^*)$ that leaves $ang(q_0\, T_i\, q_i^*) = \theta(X_i)$ in the $arc(q_0, T_i, q_i^*)$ whose length is less than $\pi/2$, being $\theta(X_i) > \pi/2$. Since the vertex has been chosen arbitrarily then $\theta(X_i)$ can be reached at any interval.

Theorem 3.8.1: Given two disjoint convex polygons P and Q, with m and n vertices respectively, computing the maximum aperture angle from P with respect to Q has time complexity $\Omega(n)$.

Then, by Theorem 3.8.1, we state the following corollary.

Corollary 3.8.1: If $m = o(n \log m)$ the algorithm for computing the maximum aperture angle in $O(n + m)$ time is faster that the corresponding algorithm which has time complexity $O(n \log m)$. In contrast, when $m = \Omega(n \log^{1+\varepsilon} m)$, for any $\varepsilon > 0$, the algorithm whose complexity is $O(n \log m)$ becomes faster than the corresponding algorithm with complexity $O(n+ m)$.

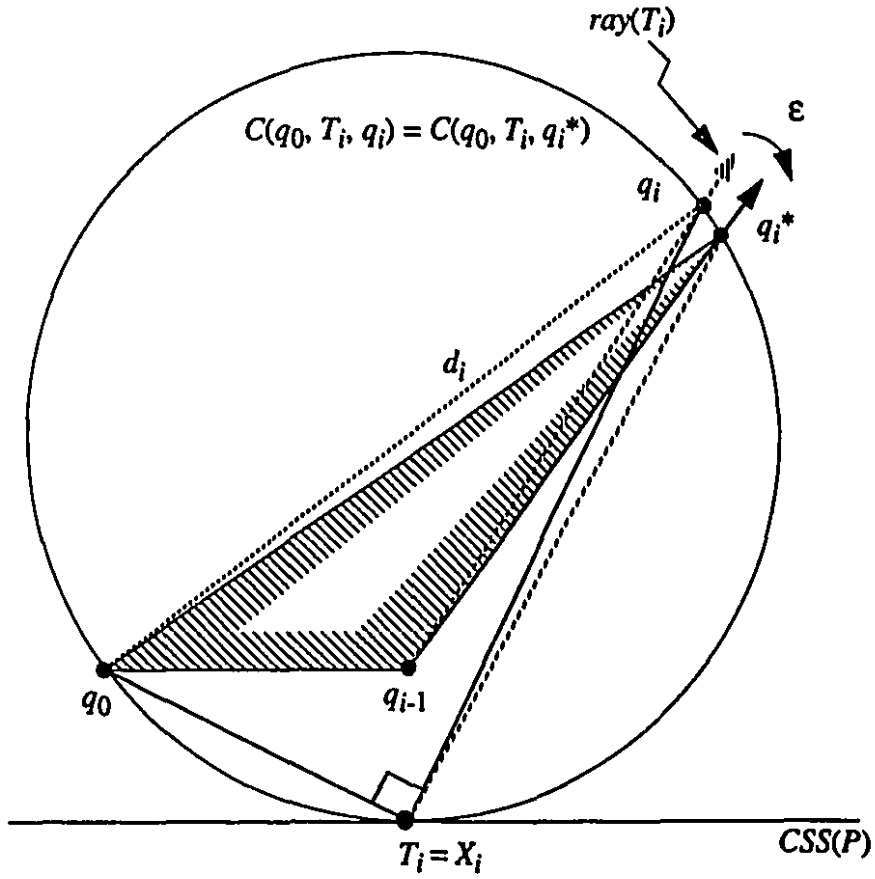

Fig. 3.8.3

The next part is to show that the minimization problem also has complexity $\Omega(n)$. The construction of the corresponding example is similar to the construction done to show Theorem 3.8.1, and is also developed by induction. We now show that for this example $\Omega(n)$ is a lower bound for the minimization problem. Consider a polygon P lying in the strip defined by $\{(x, y): 0 \leq y \leq 1/2\}$, whose external chain $ECT(P)^c$ is the segment line $p_1 p_n$ on the x-axis containing all the elements $r_0, r_1, \ldots, r_{n-1}$ of the partition R of $p_1 p_n$.

Consider the family $\Im$ as defined earlier, as well as the construction of q_0 and q_1. Denote by r_1 the orthogonal projection of the midpoint of $q_0 q_1$. Let $L(r_1)$ be the line in $\Im$ that defines q_1. Then this line passes through r_1 and it is orthogonal to $L(q_0, r_1)$. The subsequent vertex of Q, called q_2, will lie on $L(r_1)$ in order to make r_1 an element of the partition of the x-axis. Let T_1 be the orthogonal projection of q_1 on the x-axis (refer to Fig. 3.8.4). To locate q_2 on $L(r_1)$ select an arbitrary point on the x-axis a small distance to the right of T_1 and call it r_2. Then, there exists a line $L(r_2) \in \Im$ passing through r_2 that is perpendicular to $L(q_0, r_2)$. Then, $q_2 = L(r_1) \cap L(r_2)$. Consider the circle $C(q_0, r_2, q_2)$. Because $\theta(r_2) = \pi/2$ then the diagonal $d_2 = q_0 q_2$ is a diameter of $C(q_0, r_2, q_2)$. By construction r_1, q_1 and q_2 are colinear and because $ang(q_0\, r_1\, q_2) = \pi/2 = ang(q_0\, r_2\, q_2)$ then the four points lie on the circle $C(q_0, r_2, q_2)$. Note that $\theta(r_1) = \pi/2$ is defined by $ang(q_0\, r_1\, q_1)$. Thus, $p_1 p_n$ is intersected by $C(q_0, r_2, q_2)$ at r_1 and r_2 which define the interval $I_2 = [r_1, r_2]$ in the partition of $p_1 p_n$. Then, $\theta(r_1) = \pi/2 = \theta(r_2)$ is the value of $\theta(Z_2)$ in I_2. Therefore, $\theta(x)$ is unimodal over I_2.

To define the k+1-th vertex of Q let q_{k-1} be the k-th vertex and let T_{k-1} be its orthogonal projection on the x-axis. Let r_k be any point on the x-axis to the right of T_{k-1}. Consider the line $L(r_k) \in \Im$ passing through r_k and perpendicular to the line $L(q_0, r_k)$. Then, $q_k = L(r_k) \cap L(r_{k-1})$. Because $\theta(r_k) = ang(q_0\, r_k\, q_k) = \pi/2$, then the diagonal $d_k = q_0\, r_k$ is a diameter of the circle $C(q_0, r_k, q_k)$. By construction r_{k-1}, q_{k-1} and q_k lie on $L(r_{k-1})$. Since $ang(q_0\, r_{k-1}\, q_k) = \pi/2$ it implies that $r_{k-1} \in C(q_0, r_k, q_k)$. Then, the circle $C(q_0, r_k, q_k)$ intersects $p_1 p_n = ECT(P)^c$ at r_{k-1} and r_k, which are the endpoints of the interval I_k. Then r_{k-1} and r_k define the minimum value $\theta(Z_k)$ in I_k.Thus, $\theta(x)$ is unimodal over I_k. Observe that the element r_0 in the partition of $p_1 p_n = ECT(P)^c$ is determined by the intersection of $L(q_0, q_{n-1})$ with the x-axis. By the arguments used in the previous example the polygon Q obtained is convex. By construction the function $\theta(x)$ has n local minima whose value is $\pi/2$. Let us show that this example can be modified to obtain a lower bound $\Omega(n)$ for the minimization problem.

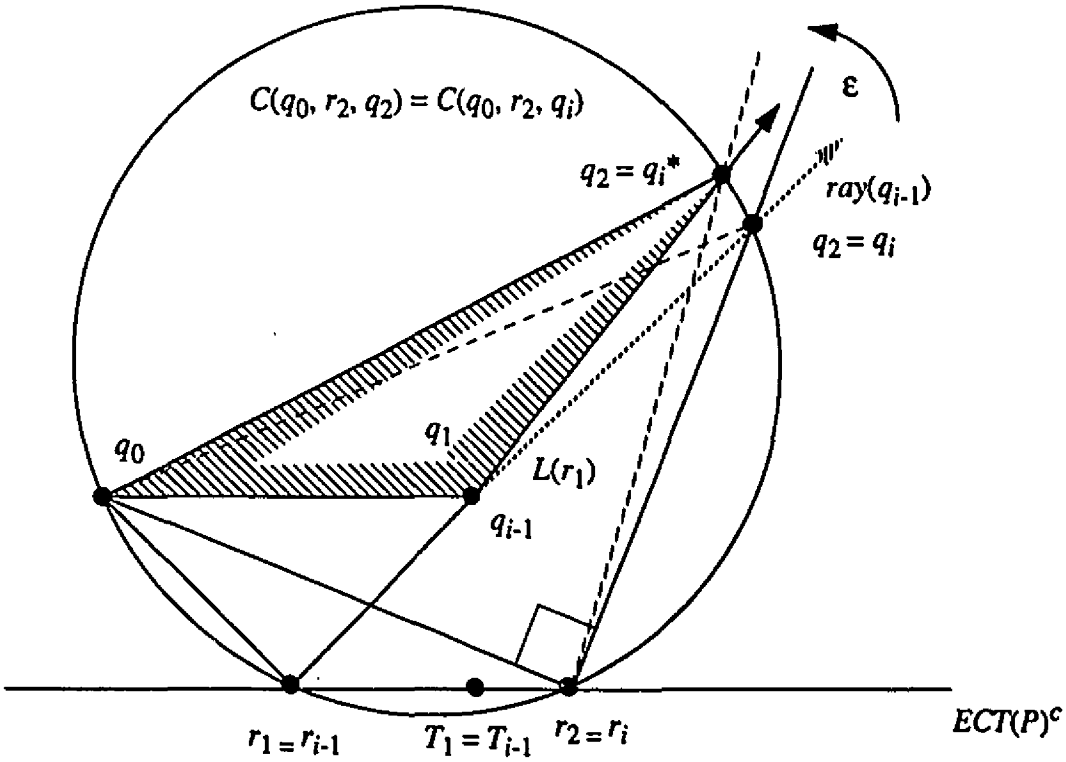

Fig. 3.8.4

Let q_i be any vertex of Q to be displaced in order to obtain a vertex q_i^* that defines a new angle $ang(q_0\ r_i\ q_i^*) = \theta(r_i)$. Such angle $\theta(Z) = \theta(r_i)$ will define the global minimum over the interval I_i. Consider the $ray(q_{i-1})$ from q_{i-1} towards q_i. Rotate this ray by an ε-positive angle counterclockwise by using q_{i-1} as center of rotation (see Fig. 3.8.5). Define q_i^* as $ray(q_{i-1}) \cap C(q_0, r_i, q_i)$. Then, the new diagonal $d_i^* = q_0\ q_i^*$ is no longer a diameter of $C(q_0, r_k, q_i) = C(q_0, r_i, q_i^*)$. In fact, d_i^* is a chord lying in $arc(q_0, r_i, q_i^*)$ of $C(q_0, r_i, q_i^*)$ whose length is greater than $\pi/2$. Therefore, $ang(q_0, r_i, q_i^*)$ is an acute angle. By construction the value of the other angles remains as $\pi/2$. Thus, $\theta(Z) = ang(q_0, r_i, q_i^*)$ is the global minimum over $ECT(P)^c$. Since q_i can be any vertex of Q, the minimum value of $\theta(Z)$ can be attained in any interval I_i in the partition of $ECT(P)^c$. Therefore, with any algorithm for solving the minimization problem we have to test each interval in order to find the point Z that defines the global minimum. Then we have proved the following result.

Theorem 3.8.2: The complexity of computing $\theta(Z)$ is $\Omega(n)$.

The following results show that the complexity of computing the minimum value is $\Omega(m)$, thus establishing the optimality of the corresponding algorithm for finding the minimum value of $\theta(x)$.

Theorem 3.8.3: The complexity of computing $\theta(Z)$ is $\Omega(m)$.

Proof: Let C be the unitary circle and let $q_n q_1 \neq diameter(C)$ be a chord of C in the upper semiplane defined by the x-axis (refer to Fig. 3.8.5). Place the remaining vertices of Q on C above the chord $q_n q_1$. Let $p_m p_1 \neq diameter(C)$ be another chord of that circle. Place the corresponding (m -2) vertices of P on C and below $p_m p_1$. Then, the minimum value of $\theta(x)$ is attained at each vertex of P. Let $T(p_i)$ be the tangent line of C at p_i. Then we can reduce the value of $\theta(Z)$ by moving p_i in an orthogonal direction of $T(p_i)$ by an ε-distance. To maintain the convexity of the polygon it is necessary to bound ε in such a way that the point p_i remains in the cone defined by the semiplanes $H^-(p_{i-2}, p_{i-1})$ and $H^+(p_{i+1}, p_{i+2})$, which are also used to define P as an intersection of semiplanes. Thus, p_i defines a smaller angle since it lies in the exterior of C. Because p_i can be any of the vertices of P, we must test each vertex in order to find the vertex p_i that determines $\theta(Z)$.

Corollary 3.8.2: The complexity of computing $\theta(Z)$ is $\Omega(\max\{m, n\})$.

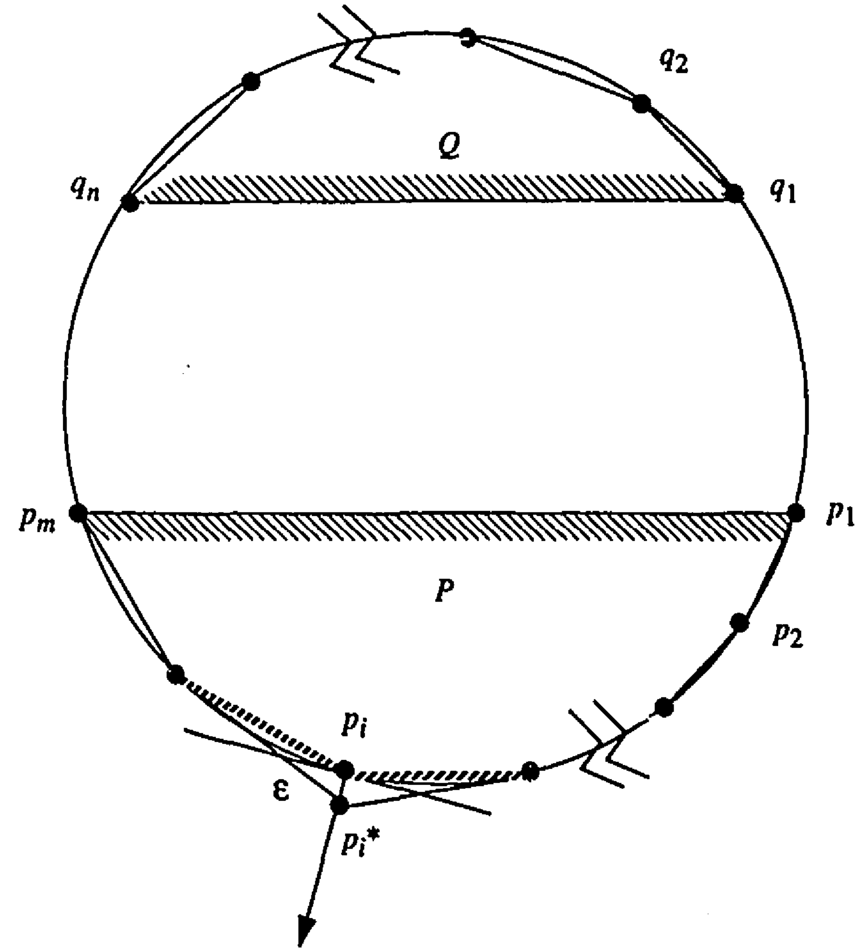

Fig. 3.8.5

CHAPTER 4

SOME APERTURE ANGLE OPTIMIZATION PROBLEMS IN 3 - D

4.1 Introduction

In Chapter 3 we solved the problem of computing the maximum and minimum aperture angles of a camera that is allowed to roam in a convex polygon in the euclidean plane E^2. In this chapter we compute the maximum aperture angle of a camera that is allowed to move in a convex polyhedron in space.

In the previous chapter we solved some basic problems of aperture angle in two dimensions. Following the same procedure in this chapter we solve the following cases: 1) the camera is allowed to move on a plane, 2) it is constrained to move along a line, 3) the camera is allowed to move in a convex polygon and 4) it may lie in a convex polyhedron.

4.2 Preliminaries

Definition 4.2.1: A subset K of E^d is called a polyhedral set if K is the intersection of a finite number of closed half-spaces, or $\mathbf{K} = E^d$.

Note that a polyhedral set is a convex set because it is the intersection of convex sets. As examples of convex polyhedral sets we have the convex polygons in E^2, and the convex polyhedra in E^3.

A subset F of a polyhedral set **K** is called a *face* of **K** if for any two distinct points **x**,**y** $\in$ **K** *int*(**xy**) $\cap$ F is non-empty. Actually we have *int*(**xy**) $\subset F$, since F is convex. A face F of **K** is called a *k-face* if the dimension of F is k. Thus, the *0-faces* of a convex polyhedron are the vertices, the *1-faces* are the edges, and the *2-faces* are the convex polygons that together form the convex polyhedron.

A set in E^3 is a *convex body* if it is compact, convex, and has a non-empty interior. A set is a closed *convex surface* if it is the boundary of a convex body.

Definition 4.4.1: A hyperplane H in E^d ($d \geq 2$) is a *supporting hyperplane* to a closed set S if H intersects the boundary of S and one of the closed half-spaces defined by H contains S. Points in the set $H \cap bd(S)$ are said to be *contact or support points* of H (with $bd(S)$), and H is said to support S at each contact point.

In the particular case when S is a polyhedral set, H is a hyperplane that supports S at any d-1 face of S. If S is a line any plane containing the line is a support plane. There exists another concept which involves support planes, a *hyperplane separates weakly* two sets, $\mathbf{K}_1$ and $\mathbf{K}_2$, if it leaves $\mathbf{K}_1$ in one closed half-space and the other closed side contains $\mathbf{K}_2$. Two sets are *separated* by a hyperplane if they are contained in opposite sides.

Let θ be a function from E^2 to E defining the surface determined by the set $\{(x, y, z) \in E^3 : z = \theta(x, y) \text{ and } (x, y) \in E^2\}$. When the domain of θ is constrained to be a subset $S \subset E^2$, we denote it by $\theta \mid S$, and we call it θ-constrained.

The circle passing through $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ is denoted by $C(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$. If the circle is determined by the center and the radius we denote it by $C(\mathbf{c}, R)$. Normally the interior is not included when referring to a circle. In cases when the interior is of interest it will be referred to as a disk $Disk(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$.

Consider a given segment **ab** in the E^3 and a given angle $\theta \notin \{0, \pi\}$. To find the locus of the points in the euclidean space that define a given angle θ consider any plane containing the segment **ab** and determine the locus of the points in this plane that define θ. Since the curve defined by such a locus does not change for any of the planes containing **ab**, the surface obtained in E^3 that determines the desired locus is obtained by rotating this curve around $L(\mathbf{a}, \mathbf{b})$.

The locus of the points on a plane containing **ab** is the union of two arcs of two symmetric circles C and C' passing trough **a** and **b**, with L(**a**, **b**) as symmetric axis. Observe that **ab** is a chord of C and it partitions C into two arcs. Let C_L and C_S denote these two arcs, where C_L denotes the large arc and C_S denotes the small one. Analogously, **ab** divides C' into two arcs C'_L and C'_S. The locus of points in the plane with a given constant angle less than $\pi/2$ is the union of two arcs C_L and C'_L (see Fig. 4.2.1.a). Analogously, the union of the arcs C_S and C'_S define the locus of points in the plane with a given constant obtuse angle (see Fig. 4.2.1.b). We will call such loci for different values of θ the *iso-aperture-angle* contours (IAA contours). Using the observations 2.3.1, 2.3.2 and 2.3.3, any point in the interior defines an angle larger that the given angle θ, and for any point in the exterior it defines an angle smaller than θ. Thus, the angle defined by each IAA contour decreases as the size of the circles defining the corresponding IAA contour increases.

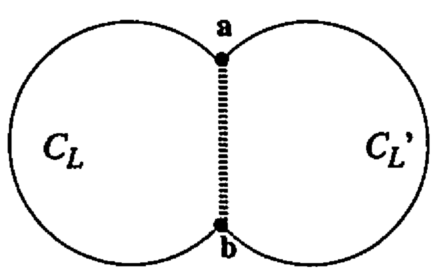

Fig. 4.2.1.a

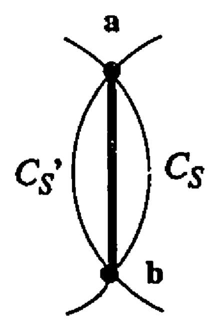

Fig. 4.2.1.b

As we have mentioned, we obtain the surface in E^3 that defines the locus of points **x** in E^3 with an aperture angle θ by rotating an IAA contour (contained in a plane containing **ab**) around L(**a**, **b**). Because such a surface is similar to a torus we call it *toroidal lunoid.* Finally, we establish the following proposition to characterize the loci defined by a toroidal lunoid.

Proposition 4.2.1: The locus of points in E^3 of constant aperture angle θ consists of an open line segment **ab** when $\theta = \pi$. The union of points at infinity and the line through **ab** excluding segment **ab** defines the locus when θ equals zero. The locus of points in E^3 of constant aperture angle $\theta \notin \{0, \pi\}$ consists of a toroidal lunoid defined for the segment **ab**.

As a generalization and a consequence of observations 2.3.1, 2.3.2 and 2.3.3, we state the following proposition.

Proposition 4.2.2: Let $T(\mathbf{a}, \mathbf{b})$ be a toroidal lunoid defined for **ab**, and let **X** be a point on $bd(T(\mathbf{a}, \mathbf{b}))$. Let **Y** be any point in $E^3 \setminus \{\mathbf{a}, \mathbf{b}\}$. Then:

$$\text{If } \mathbf{Y} \in ext(T(\mathbf{a}, \mathbf{b})) \text{ then } ang(\mathbf{aYb}) < ang(\mathbf{aXb})$$

$$\text{If } \mathbf{Y} \in bd(T(\mathbf{a}, \mathbf{b})) \text{ then } ang(\mathbf{aYb}) = ang(\mathbf{aXb})$$

$$\text{If } \mathbf{Y} \in int(T(\mathbf{a}, \mathbf{b})) \text{ then } ang(\mathbf{aYb}) > ang(\mathbf{aXb})$$

4.3 The Segment-Plane Problem

The segment-plane aperture angle problem is defined as follows: let **ab** be a line segment in E^3 lying in one of the open half-spaces defined by a given plane H. Determine a point **X** on the plane H such that the aperture angle $ang(\mathbf{aXb}) = \theta(\mathbf{X})$ is maximum. Without loss of generality suppose that the given plane H is determined by $\{(x, y, z): z = 0\}$.

Using the definition of the locus of the points in the space that have a fixed aperture angle we are able to solve this problem. The set of points which define the maximum aperture angle $ang(\mathbf{aXb}) = \theta(\mathbf{X})$ has to lie on some toroidal lunoid $T(\mathbf{a}, \mathbf{b})$, with respect to **ab**, and simultaneously it lies on the given plane H. Thus, we have to consider the family of toroidal lunoids $T(\mathbf{a}, \mathbf{b})$ such that $T(\mathbf{a}, \mathbf{b}) \cap H \neq \emptyset$. For the particular case in which $int(T(\mathbf{a}, \mathbf{b})) \cap H \neq \emptyset$, there exists a point $\mathbf{Y} \in int(T(\mathbf{a}, \mathbf{b})) \cap H$ such that $\theta(\mathbf{Y})$ is larger than $\theta(\mathbf{x})$ for any point on $bd(T(\mathbf{a}, \mathbf{b}))$, by Proposition 4.2.2. Then, these kinds of toroidal lunoids $T(\mathbf{a}, \mathbf{b})$ which are intersected in their interior by H do not define the set of points $\mathbf{X} \in H$ for which the aperture angle $ang(\mathbf{aXb})$ is maximum. Therefore, the only way to define those points is using toroidal lunoids which are *tangent* to the given plane H. We say that a toroidal lunoid $T(\mathbf{a}, \mathbf{b})$ is *tangent* to a plane H (or tangent to a polyhedral set S) if $bd(T(\mathbf{a}, \mathbf{b})) \cap H \neq \emptyset$ ($bd(T(\mathbf{a}, \mathbf{b})) \cap S \neq \emptyset$) and $int(T(\mathbf{a}, \mathbf{b})) \cap int(H) = \emptyset$ ($int(T(\mathbf{a}, \mathbf{b})) \cap int(S) = \emptyset$) using the topology defined for E^3. Observe that under this topology $int(H) = \emptyset$. We denote by $T^*(\mathbf{a}, \mathbf{b})$ a toroidal which is tangent to H.

Lemma 4.3.1: If a toroidal lunoid $T^*(\mathbf{a}, \mathbf{b})$ is tangent to a plane H at a set of points $M_H \subset H$, then, $M_H = \{\mathbf{X} \in H: \theta(\mathbf{X}) \text{ is the maximum value of } \theta(\mathbf{x}) \text{ for } \mathbf{x} \in H\}$.

Proof: Let $T^*(\mathbf{a}, \mathbf{b})$ be a toroidal lunoid that is tangent to H at M. Let $\mathbf{y} \in H$ be any other point such that $\mathbf{y} \neq \mathbf{X}$ for each $\mathbf{X} \in M_H$. By Proposition 4.2.2 the angle $\theta(\mathbf{y})$ is smaller than $\theta(\mathbf{X})$ for each $\mathbf{X} \in M_H$. Q. E. D.

The following lemma characterizes the circle whose arc defines the toroidal lunoid which is tangent to H.

Lemma 4.3.2: A toroidal lunoid $T^*(\mathbf{a}, \mathbf{b})$ determines the set of points $M_H = \{\mathbf{X} \in H\colon \theta(\mathbf{X})$ is the maximum value of $\theta(\mathbf{x})$ for $\mathbf{x} \in H\}$ if and only if it is generated by an arc defined by a circle $C(\mathbf{a}, \mathbf{b}, \mathbf{X})$ where $\mathbf{X}$ is the point where the maximum is attained for $\theta(\mathbf{x})$, when $\mathbf{x}$ is allowed to move on a line $L(H, H^{\perp})$ for some plane $H^{\perp}$ which is orthogonal to H and that contains $\mathbf{ab}$.

Proof: $\Rightarrow$) For each $\mathbf{X} \in M_H$ there exists a unique plane $H(\mathbf{a}, \mathbf{b}, \mathbf{X})$ that intersects $T^*(\mathbf{a}, \mathbf{b})$ in an IAA contour. Such contour is defined by a circle $C(\mathbf{a}, \mathbf{b}, \mathbf{X})$ which is contained in $H(\mathbf{a}, \mathbf{b}, \mathbf{X})$ and is tangent to H at $\mathbf{X}$. In particular, it is tangent to the line L defined by intersecting H and $H(\mathbf{a}, \mathbf{b}, \mathbf{X})$. Because this line and the circle $C(\mathbf{a}, \mathbf{b}, \mathbf{X})$ lie on the same plane, $H(\mathbf{a}, \mathbf{b}, \mathbf{X})$, its vector radius $\mathbf{r_x}$ is perpendicular to L. Then $\mathbf{r_x}$ is a normal vector for the plane H. Thus H and $H(\mathbf{a}, \mathbf{b}, \mathbf{X})$ are orthogonal planes, i.e., $H^{\perp} = H(\mathbf{a}, \mathbf{b}, \mathbf{X})$ and L is the intersection set between the planes H and $H^{\perp}$. We denote in this case L by $L(H, H^{\perp})$. Then it follows that there exists a circle $C(\mathbf{a}, \mathbf{b}, \mathbf{X})$ that is tangent to $L = L(H, H^{\perp})$ and contained in a plane $H^{\perp} = H(\mathbf{a}, \mathbf{b}, \mathbf{X})$ orthogonal to H. Since $\mathbf{X} \in L(H, H^{\perp})$ is a point where $\theta(\mathbf{x})$ is maximum for all $\mathbf{x} \in H$ it follows that $\theta(\mathbf{X})$ is maximum when $\mathbf{x} \in L(H, H^{\perp})$. Therefore, $\theta(\mathbf{X})$ is the maximum for the Line-Segment problem in $H^{\perp}$.

$\Leftarrow$) Let $\mathbf{X}$ be the point where $\theta(\mathbf{x})$ attains its maximum value $\theta(\mathbf{X})$, when $\mathbf{x}$ is allowed to move on $L(H, H^{\perp})$. This point is determined by the circle $C(\mathbf{a}, \mathbf{b}, \mathbf{X}) \subset H^{\perp}$ that contains $\mathbf{a}$, $\mathbf{b}$ and that is tangent to $L(H, H^{\perp})$ at $\mathbf{X}$. Let $T^*(\mathbf{a}, \mathbf{b})$ be the toroidal lunoid obtained when the corresponding tangent arc of $C(\mathbf{a}, \mathbf{b}, \mathbf{X})$ is rotated around $L(\mathbf{a}, \mathbf{b})$. Note that the path followed by each point $\mathbf{x} \in C(\mathbf{a}, \mathbf{b}, \mathbf{X})$ during the rotation around $L(\mathbf{a}, \mathbf{b})$ is a circle (see Fig. 4.3.1). This circle has center at the orthogonal projection $\mathbf{d}(\mathbf{x})$ of $\mathbf{x}$ on $L(\mathbf{a}, \mathbf{b})$ and has radius $r = \|\mathbf{x} - \mathbf{d}(\mathbf{x})\|_2$, where $\| \, \|_2$ denotes the euclidean norm. Denote that circle by $C(\mathbf{d}(\mathbf{x}), r)$. We claim that $T^*(\mathbf{a}, \mathbf{b})$ is tangent to H at $\mathbf{X}$. To prove this claim suppose H intersects the interior of $T^*(\mathbf{a}, \mathbf{b})$. Then there exists a point $\mathbf{x} \in C(\mathbf{a}, \mathbf{b}, \mathbf{X})$ such that $C(\mathbf{d}(\mathbf{x}), r)$ is intersected by H in a segment $\mathbf{X}_1\mathbf{X}_2 \subset H$

$\cap$ *int*(T(**a**, **b**)) which defines a chord of $C(\mathbf{d}(\mathbf{x}), r)$. Such a chord intersects L at a point $\mathbf{X}_0 \in \mathbf{X}_1\mathbf{X}_2 \subset$ *int*(T(**a**, **b**)) such that $\theta(\mathbf{X}_0) > \theta(\mathbf{x})$, by Proposition 4.2.2. Because $\theta(\mathbf{x}) = \theta(\mathbf{X})$ it follows that $\theta(\mathbf{X}_0) > \theta(\mathbf{X})$, which is a contradiction. Q. E. D.

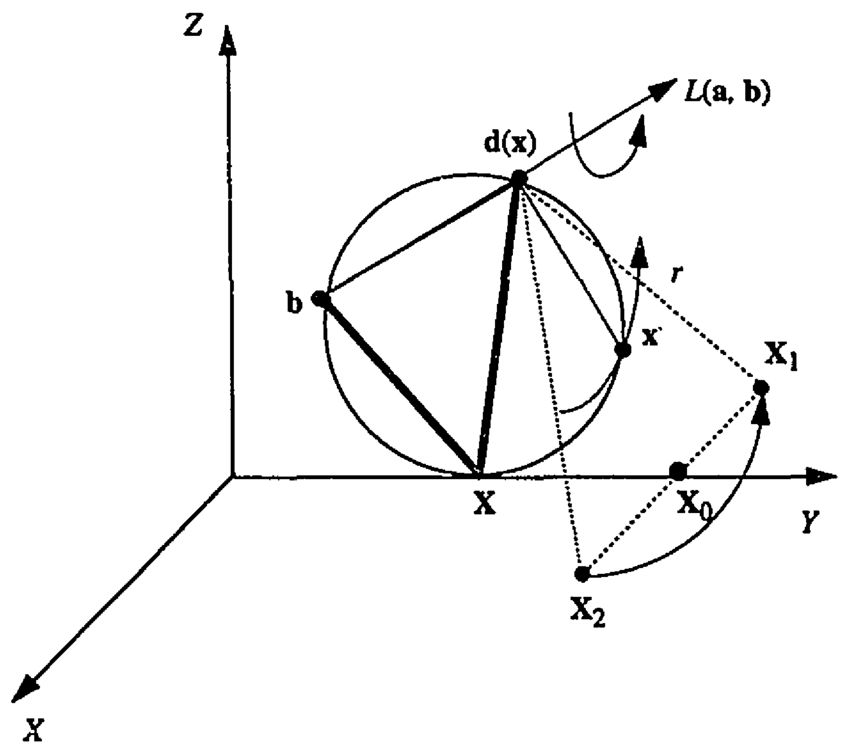

Fig. 4.3.1

The most important consequence of this lemma is that it provides a characterization of a circle that defines T^*(**a**, **b**).

We note that for each $\mathbf{X} \in M_H$ there exists a plane defined by **a**, **b**, and **X**. This plane intersects T^*(**a**, **b**) in a IIA contour. Such a contour is generated by a circle C(**a**, **b**, **X**) that is tangent to H at **X** and contained on a plane which is orthogonal to H. Then, the vector that defines the radius $r_{\mathbf{X}} = \mathbf{c} - \mathbf{X}$ of C(**a**, **b**, **X**) is perpendicular to H, and the center **c** of that circle together with the set of points **a**, **b**, and **X** lie on the same plane. Thus, the center of C(**a**, **b**, **X**) satisfies the following properties:

$$d^2(\mathbf{c}, \mathbf{a}) = d^2(\mathbf{c}, \mathbf{b}) \qquad \text{...(1)}$$

$$d^2(\mathbf{c}, \mathbf{x}) = d^2(\mathbf{c}, \mathbf{a}) \text{ or } d^2(\mathbf{c}, \mathbf{x}) = d^2(\mathbf{c}, \mathbf{b}) \text{ for } \mathbf{x} \in H \qquad \text{...(2)}$$

$$< \mathbf{c} - \mathbf{x}, \mathbf{v} > = 0 \text{ for all vector } \mathbf{v} \in H \qquad ...(3)$$

$$det\,(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{x}, \mathbf{1}) = 0$$

$$\mathbf{x} \in H \qquad ...(4)$$

For some particular cases it is not needed to solve the system determined by these conditions. When the segment **ab** is parallel to H the maximum value of $\theta(\mathbf{x})$ is reached at a unique point which is the projection **X** of the midpoint of **ab** on the line $L = L(H, H^{\perp})$. This line is obtained by intersecting the given plane H and a plane $H^{\perp}$containing the given segment **ab** and orthogonal to H. If the segment **ab** is parallel to z-axis, we can suppose **ab** is on the z-axis. Let $\mathbf{a} = (0, 0, a_3)$ and $\mathbf{b} = (0, 0, b_3)$ be the end points of the segment **ab**. The line $L(H, H^{\perp})$ is defined by the y-axis. By Lemma 3.2.1 there are two points Y_1 and Y_2 where the maximum aperture angle reaches its maximum value with respect to **ab** (see Fig. 4.3.2, pag. 62). In fact these two points are symmetric with respect to the z-axis. Let $C(\mathbf{a}, \mathbf{b}, Y_1)$ and $C(\mathbf{a}, \mathbf{b}, Y_2)$ be the two tangent circles to the y-axis that define the points Y_1 and Y_2. Without loss of generality suppose the large arc of $C(\mathbf{a}, \mathbf{b}, Y_1)$ is rotated around the z-axis in order to obtain $T^*(\mathbf{a}, \mathbf{b})$. Then the path followed by Y_1 is the circle $C(0, R)$, and $T^*(\mathbf{a}, \mathbf{b})$ is tangent to H at $C(0, R)$. By Lemma 4.3.1, $M_H = \{\mathbf{X} \in H: \mathbf{X} \in C(0, R)\} = \{\mathbf{X} \in H: \theta(\mathbf{X})$ is the maximum value of $\theta(\mathbf{x})$ for $\mathbf{x} \in H\}$.

From the last analysis follows the following proposition.

Proposition 4.3.1: The set of tangency points M_H between $T^*(\mathbf{a}, \mathbf{b})$ and H is determined by:

a) a unique point **X** if **ab** is parallel to H, or

b) a circle $C(\mathbf{c}, R)$, if **ab** is perpendicular to H, with center **c** at the orthogonal projection of **ab** on H and radius $R = \|\mathbf{c} - \mathbf{X}\|_2$, or

c) a unique point **X** when **ab** is neither parallel nor orthogonal to H.

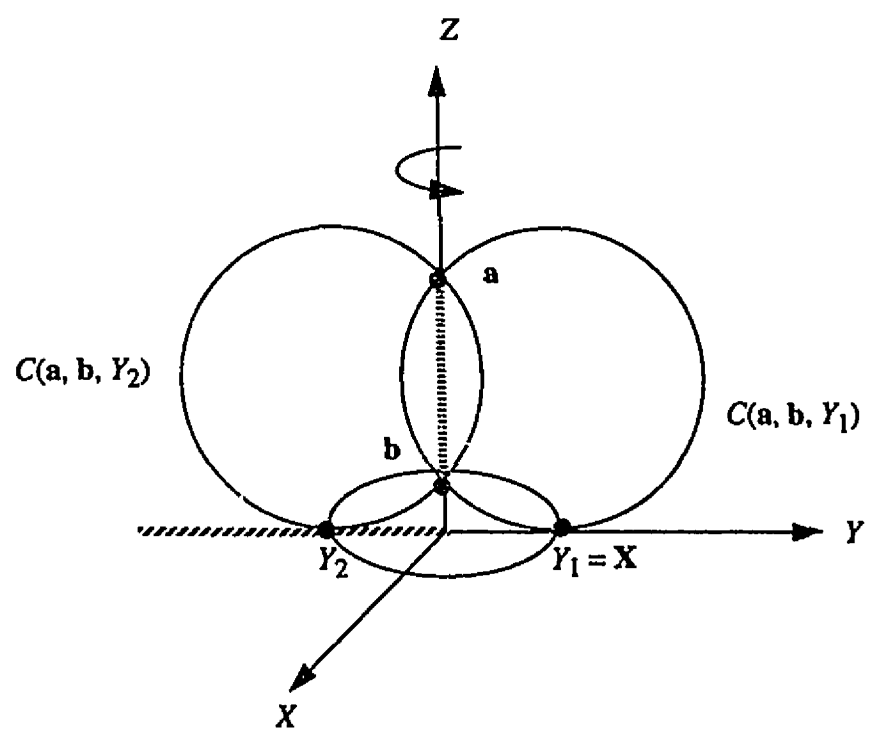

Fig. 4.3.2

Lemma 4.3.2 allows us to characterize the global maximum of $\Theta(\mathbf{x})$, but it does not provide information about the number of possible local maxima. For the segment-line problem in *two dimensions* this information was found by using the fact that $\Theta(x)$ is a bimodal function. However, there is not a concept of a function k-modal when the function is defined from E^3 to E. Nevertheless, we can develope a geometric analysis using the toroidal lunoids. By Lemma 4.3.2 the toroidal lunoid that defines the set of points M_H is generated by a circle $C(\mathbf{a}, \mathbf{b}) \subset H^{\perp}$ that defines a point in $\mathbf{X} \in L(H, H^{\perp})$, which is also the solution to the segment-line problem on the plane $H^{\perp}$. If **ab** is non-perpendicular to H, by Lemma 3.2.1 there exists other circle $C'(\mathbf{a}, \mathbf{b}) \subset H^{\perp}$ (containing **a** and **b**) which is tangent to $L(H, H^{\perp})$ at some point **X'**. Then, the IAA contour generated by $C'(\mathbf{a}, \mathbf{b})$ intersects $L(H, H^{\perp})$ at two different points. Furthermore, the path followed by **X'** after the rotation is a circle that is totally separated from $C'(\mathbf{a}, \mathbf{b})$ by H. Then, the toroidal lunoid $T'(\mathbf{a}, \mathbf{b})$ generated by the circle $C'(\mathbf{a}, \mathbf{b})$ is intersected by the plane H in its interior. Hence $C'(\mathbf{a}, \mathbf{b})$ is a circle satisfying conditions (1) to (4), however it does not define a local maxima for the aperture angle function $\Theta(\mathbf{x})$ for $\mathbf{x} \in H$.

4.4 The Segment-Line Problem

Let **ab** be a line segment in E^3 and given a line L in E^3 that does not intersect **ab**. The segment line aperture angle problem is to find a point **X** on the line L such that the aperture angle *ang*(**aXb**) = θ(**X**) is the maximum value of θ(**x**) = *ang*(**axb**) for $\mathbf{x} \in L$. In this case we denote the aperture angle function by $\theta \mid L$. Without lose of generality suppose L is the x-axis.

Using the same idea as in Section 4.3, the set $M_L = \{\mathbf{X} \in L\text{: } \theta(\mathbf{X}) \text{ is the maximum value of } \theta \mid L\}$ is defined by the tangency points between a toroidal lunoid $T^*(\mathbf{a}, \mathbf{b})$, defined for **ab**, and L. The following result characterizes the set of circles which can define the toroidal lunoid that is tangent to the given line L.

Lemma 4.4.1: The set of tangency points M_L between $T^*(\mathbf{a}, \mathbf{b})$ and L are determined by rotating a circle $C(\mathbf{a}, \mathbf{b})$ (containing **a** and **b**) which satisfies: a) $C(\mathbf{a}, \mathbf{b})$ is tangent to L at **X** and b) its vector radius $r_{\mathbf{X}} = \mathbf{c} - \mathbf{X}$ is orthogonal to the director vector of L.

Proof: Let $\mathbf{a} = (a_1, a_2, a_3)$ and $\mathbf{b} = (b_1, b_2, b_3)$ be the coordinates of **a** and **b** respectively. Let $\mathbf{c} = (h(\mathbf{x}), k(\mathbf{x}), l(\mathbf{x}))$ be the center of a circle $C(\mathbf{a}, \mathbf{b}, \mathbf{x})$ that contains **a**, **b** and $\mathbf{x} = (x, y, z)$. To find the point **X** where the maximum value is attained consider the family of circles $C(\mathbf{a}, \mathbf{b}, \mathbf{x})$ which are tangent to $L = \{(x, y, z)\text{: } z = 0 \text{ and } y = 0\}$ at **x**, then these circles satisfy:

$$d^2(\mathbf{a}, \mathbf{c}) = d^2(\mathbf{b}, \mathbf{c})$$

$$d^2(\mathbf{a}, \mathbf{c}) = d^2(\mathbf{x}, \mathbf{c}) \text{ or } d^2(\mathbf{b}, \mathbf{c}) = d^2(\mathbf{x}, \mathbf{c})$$

$$\mathbf{x} \in L$$

The first equation denotes the set of points which are equidistant from **a** and **b**, i.e., it represents the equation of a plane whose normal is defined by **ab**. This plane contains the midpoint of **ab**. Then it can be rewritten as:

$$\langle \mathbf{a} - \mathbf{b}, \mathbf{c} \rangle = (\| \mathbf{a} \|_2^2 - \| \mathbf{b} \|_2^2) / 2 \qquad \text{...(1)}$$

Analogously the second condition is a plane represented by the equation:

$$\langle \mathbf{a} - \mathbf{x}, \mathbf{c} \rangle = (\| \mathbf{a} \|_2^2 - \| \mathbf{x} \|_2^2) / 2 \qquad \text{...(2)}$$

The meaning of the last condition is that $\mathbf{x} = (x, 0, 0)$. By Lemma 4.4.1 we must represent a condition of orthogonality between the vector radius $\mathbf{r_x} = \mathbf{c} - \mathbf{x}$ of $C(\mathbf{a}, \mathbf{b}, \mathbf{x})$ and the vector (1, 0, 0), which is the direction of L. This condition is:

$$h = x \qquad \text{...(3)}$$

Finally, since $C(\mathbf{a}, \mathbf{b}, \mathbf{x})$ is determined by $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{x}$, its center $\mathbf{c}$ belongs to the plane defined by these three points, i.e., the four points ($\mathbf{a}$, $\mathbf{b}$, $\mathbf{x}$ and $\mathbf{c}$) are coplanar. This condition is stated as follows:

$$det(\mathbf{a}\ 1, \mathbf{b}\ 1, \mathbf{c}\ 1, \mathbf{x}\ 1) = 0 \qquad \text{...(4)}$$

We first solve the system of the two linear equations for k and l, defined by the equations (1) and (2). The solution of that system yields to an expression of the two variables as a function of x, i.e., $k = k(x)$, and $l = l(x)$. The corresponding solution is:

$$k(x) = -\frac{1}{\Delta}\{\frac{(a_3 - b_3)}{2}x^2 + (a_1 b_3 - a_3 b_1)x + \frac{(\|b\|_2^2 a_3 - \|a\|_2^2 b_3)}{2}\}$$

and

$$l(x) = \frac{1}{\Delta}\{\frac{(a_2 - b_2)}{2}x^2 + (a_1 b_2 - a_2 b_1)x + \frac{(\|b\|_2^2 a_2 - \|a\|_2^2 b_2)}{2}\}$$

where $\Delta = a_3(a_2 - b_2) - a_2(a_3 - b_3) \neq 0$, if and only if $\mathbf{ab}$ and L are non-coplanars, in which case the solution is found by using the *LABMAX* algorithm. This algorithm is described in Sec. 3.2.

Condition (4) is represented as:

$$det\begin{bmatrix} a_1 & a_2 & a_3 & 1 \\ b_1 & b_2 & b_3 & 1 \\ h & k & l & 1 \\ x & 0 & 0 & 1 \end{bmatrix} = -x\,det\begin{bmatrix} a_2 & a_3 & 1 \\ b_2 & b_3 & 1 \\ k & l & 1 \end{bmatrix} + det\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ h & k & l \end{bmatrix} = \Psi = 0$$

By evaluating each of these determinants we obtain an expression for Ψ in terms of k and l.

$$\Psi = [(a_2 - b_2)l - (a_3 - b_3)k\ ]x + (a_1 b_2 - a_2 b_1)l - (a_1 b_3 - a_3 b_1)k$$

This expression yields a polynomial equation of degree three for $\Psi = \Psi(x)$ in the variable x, because $k(x)$ and $l(x)$ are polynomial equations of degree two in x.

$$\Psi(x) = \frac{1}{2\Delta}(((a_3-b_3)^2+(a_2-b_2)^2)x^3 +$$

$$3[(a_3-b_3)(a_1b_3-b_1a_3)+(a_2-b_2)(a_1b_2-b_1a_2)]x^2 +$$

$$(2(a_1b_3-b_1a_3)^2+(\|b\|_2^2a_3-\|a\|_2^2b_3)(a_3-b_3) +$$

$$2(a_1b_2-b_1a_2)^2+(\|b\|_2^2a_2-\|a\|_2^2b_2)(a_2-b_2))x +$$

$$(a_1b_3-b_1a_3)(\|b\|_2^2a_3-\|a\|_2^2b_3)+(a_1b_2-b_1a_2)(\|b\|_2^2a_2-\|a\|_2^2b_2))$$

Let A, B and C be defined as follows:

$$A = (a_3-b_3)^2+(a_2-b_2)^2,$$

$$B = (a_3-b_3)(a_1b_3-b_1a_3)+(a_2-b_2)(a_1b_2-b_1a_2),$$

$$C = (2(a_1b_3-b_1a_3)^2+(\|b\|_2^2a_3-\|a\|_2^2b_3)(a_3-b_3)) +$$

$$(2(a_1b_2-b_1a_2)^2+(\|b\|_2^2a_2-\|a\|_2^2b_2)(a_2-b_2))$$

and $D = (a_1b_3-b_1a_3)(\|b\|_2^2a_3-\|a\|_2^2b_3)+(a_1b_2-b_1a_2)(\|b\|_2^2a_2-\|a\|_2^2b_2)$.

Thus, the cubic equation for $\Psi(x)$ is expressed as:

$$\frac{1}{2\Delta}(Ax^3+3Bx^2+Cx+D) = 0$$

Let $\mathbf{u}_1 = (\mathbf{x} - \mathbf{a})$ and $\mathbf{u}_2 = (\mathbf{x} - \mathbf{b})$ be the vectors which are used to define the aperture angle as:

$$\theta(\mathbf{x}) = \mathrm{acos}\frac{\mathbf{u}_1 \cdot \mathbf{u}_2}{\|\mathbf{u}_1\|_2\|\mathbf{u}_2\|_2}$$

The *critical points* of a real function, in particular for the aperture angle function, are obtained by solving the equation defined by setting its derivative equal to zero, i.e., $\frac{d}{dx}\theta = \frac{d}{dx}\left(\operatorname{acos}\frac{\mathbf{u}_1 \cdot \mathbf{u}_2}{\|\mathbf{u}_1\|_2\|\mathbf{u}_2\|_2}\right) = 0$. The corresponding derivative is given by the following equation:

$$\frac{d}{dx}\left(\operatorname{acos}\frac{\mathbf{u}_1 \cdot \mathbf{u}_2}{\|\mathbf{u}_1\|_2\|\mathbf{u}_2\|_2}\right) = -\frac{\|\mathbf{u}_1\|\|\mathbf{u}_2\|}{\sqrt{\|\mathbf{u}_1\|_2^2\|\mathbf{u}_2\|_2^2 - (\mathbf{u}_1 \cdot \mathbf{u}_2)^2}}\frac{d}{dx}\frac{\mathbf{u}_1 \cdot \mathbf{u}_2}{\|\mathbf{u}_1\|_2\|\mathbf{u}_2\|_2}$$

$$-\frac{1}{\|\mathbf{u}_1\|_2^2\|\mathbf{u}_2\|_2^2\sqrt{\|\mathbf{u}_1\|_2^2\|\mathbf{u}_2\|_2^2 - (\mathbf{u}_1 \cdot \mathbf{u}_2)^2}}\{(x^2 - (a_1+b_1) + \mathbf{a}\cdot\mathbf{b})\,(2x^3$$

$$-3(a_1+b_1)x^2 + (\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + 4a_1b_1)x - (\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1)\,)$$

$$-[x^4 - 2(a_1+b_1)x^3 + (\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + 4a_1b_1)x^2 - 2(\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1)x$$

$$+ (\|\mathbf{a}\|_2^2\|\mathbf{b}\|_2^2)\,]\,(2x - (a_1+b_1))\,\}$$

Thus, the following factor must be equal zero:

$$\{(x^2 - (a_1+b_1) + \mathbf{a}\cdot\mathbf{b})\,(2x^3 - 3(a_1+b_1)x^2 + (\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + 4a_1b_1)x$$

$$-(\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1)\,) - [x^4 - 2(a_1+b_1)x^3 + (\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + 4a_1b_1)x^2$$

$$-2(\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1)x + (\|\mathbf{a}\|_2^2\|\mathbf{b}\|_2^2)\,]\,(2x - (a_1+b_1))\,\} = 0$$

This expression is reduced to a cubic equation in the x variable,

$$[2(\mathbf{a}\cdot\mathbf{b}) + (a_1+b_1)^2 - (\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + 4a_1b_1)\,]x^3 +$$

$$3[(\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1) - (a_1+b_1)(\mathbf{a}\cdot\mathbf{b})\,]x^2 +$$

$$[(\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + 4a_1b_1)(\mathbf{a}\cdot\mathbf{b})$$

$$-(\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1)(a_1+b_1) - 2(\|\mathbf{a}\|_2^2\|\mathbf{b}\|_2^2)\ ]x +$$

$$[\|\mathbf{a}\|_2^2\|\mathbf{b}\|_2^2(a_1+b_1) - (\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1)(\mathbf{a}\cdot\mathbf{b})]$$

Let $\alpha = [2(\mathbf{a}\cdot\mathbf{b}) + (a_1+b_1)^2 - (\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + 4a_1b_1)\ ]$,

$$\beta = (\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1) - (a_1+b_1)(\mathbf{a}\cdot\mathbf{b}),$$

$$\gamma = (\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + 4a_1b_1)(\mathbf{a}\cdot\mathbf{b}) - (\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1)(a_1+b_1) - 2(\|\mathbf{a}\|_2^2\|\mathbf{b}\|_2^2)$$

and $\lambda = \|\mathbf{a}\|_2^2\|\mathbf{b}\|_2^2(a_1+b_1) - (\|\mathbf{a}\|_2^2 b_1 + \|\mathbf{b}\|_2^2 a_1)(\mathbf{a}\cdot\mathbf{b})$ be the coefficients of the cubic equation $(\alpha x^3 + 3\beta x^2 + \gamma x + \lambda) = 0$. Then, after some algebraic manipulations in the coefficients of $(Ax^3 + 3Bx^2 + Cx + D) = 0$ it follows that $\alpha = A$, $\beta = B$, $\gamma = C$ and $\lambda = D$. Q. E. D.

Note that the aperture angle function may be non-differentiable in the set of points where the denominator of $\frac{d}{dx}\theta$ is zero, i.e., where $(\|\mathbf{u}_1\|_2^2\|\mathbf{u}_2\|_2^2 - (\mathbf{u}_1\cdot\mathbf{u}_2)^2)$ is zero. In fact it is non-differentiable at the intersection point between the given line L and the line $L(\mathbf{a}, \mathbf{b})$, as a consequence of Lemma 3.2.2.

The most important conclusion of Lemma 4.4.1 is that using our formulation we solve a problem, which can be solved with optimization tools, by using geometric arguments. This allows us to use the model of computation that we have defined in Section 2.2 to obtain an exact solution.

The solutions of an equation of degree three are given by Cardan's formulas [Us48]. We describe these formulas in the Appendix. Since Lemma 4.4.1 states the equivalence of the two cubic equations, $(\alpha x^3 + 3\beta x^2 + \gamma x + \lambda) = 0$ and $(Ax^3 + 3Bx^2 + Cx + D) = 0$ which represent the solution of the segment-line problem in three dimensions, the solution of one or the other is the same. Then we refer to the solution set for both equations as x_1, x_2 and x_3.

If the solution set has three real roots (x_1, x_2 and x_3) then define $\mathbf{X}_1 = (x_1, 0, 0)$, $\mathbf{X}_2 = (x_2, 0, 0)$, and $\mathbf{X}_3 = (x_3, 0, 0)$ as the *critical points*, or the points on the line L which are candidates to define the maximum value $\theta(\mathbf{x})$. To find which point yields a maximum value can be done by evaluating $\theta(\mathbf{x})$ at each of these points and considering the largest value as the maximum value of $\theta|L$. In this case, one of those points is the global maximum $\mathbf{X}$ of $\theta|L$, the other two points can correspond to local maxima or to saddle points of $\theta|L$, or minima. If there exists a unique real root $\mathbf{X}$, it determines the point where the maximum value is reached.

Algorithm SLNO: Segment-Line Problem

Input: A given line $L = \{(x, y, z): z = 0 \text{ and } y = 0\}$ that lies on the plane $H = \{(x, y, z): z = 0\}$, and the coordinates of the endpoints of **ab**. The segment **ab** is non-orthogonal to H and it lies in one of the open half-spaces defined by H.

Output: A point $\mathbf{X}$ where $\theta(\mathbf{X})$ is maximum.

Begin

Step 1.- Determine the point $\mathbf{Y}$ where the aperture angle is maximum when $\mathbf{x}$ is allowed to move on the plane H.

Step 2.- Test whether L contains $\mathbf{Y}$:

If $\mathbf{Y} \in L$ then exit with $\mathbf{X} = \mathbf{Y}$.

else

Find Δ, if $\Delta = 0$ then call *LABMAX* algorithm,

else

Let $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\}$ be the corresponding critical points.

Exit with $\mathbf{X}$ such that $\theta(\mathbf{X}) = \max\{\theta(\mathbf{X}_1), \theta(\mathbf{X}_2), \theta(\mathbf{X}_3)\}$

end

Theorem 4.4.1: Algorithm *SLNO* finds in constant time a point $\mathbf{X} \in L$ where $\theta(\mathbf{X})$ is the maximum value of $\theta(\mathbf{x})$.

Proof: The solution of the segment-plane problem is done in constant time using the *LABMAX* algorithm (Sec. 3.2) on $L(H, H^{\perp})$. Therefore Step 1 may be done in $O(1)$ time. To test if $\mathbf{Y} \in L$ can be done in constant time. Since the algebraic expression for Δ is determined by sums, differences, products and divisions, to find it is done in $O(1)$ time, as well as the call to *LABMAX* algorithm. Evaluating $\theta(\mathbf{X}_i)$ is done in constant time. The comparison between the possible values of $\theta(\mathbf{X}_i)$ is done in $O(1)$ time. Therefore, *MLNO* algorithm finds in constant time a point $\mathbf{X} \in L$ where $\theta(\mathbf{X})$ is the maximum value of $\theta \mid L$. Q. E. D.

4.5 The Segment-Segment Problem

Let **ab** be a segment in E^3 and let **ed** be a segment such that $\mathbf{ed} \cap \mathbf{ab} = \emptyset$. Find a point **X** in the interval I (defined by **ed**) where the maximum value is reached for $\theta(\mathbf{x})$ when $\mathbf{X} \in I$. We denote in this case the corresponding function by $\theta \mid I$.

Since the segment is contained on the line $L(\mathbf{e}, \mathbf{d})$ defined by the two end points of **ed**, we can express it as a closed interval $I = [\mathbf{e}, \mathbf{d}] \subset L(\mathbf{e}, \mathbf{d})$. Without loss of generality we assume that $L(\mathbf{e}, \mathbf{d}) \cap \mathbf{ab} = \emptyset$. If $L(\mathbf{e}, \mathbf{d}) \cap \mathbf{ab} \neq \emptyset$, the four points **a**, **b**, **d**, and **e** lie on a plane and we solve the corresponding problem (Segment-Segment, Sec. 3.3) on a plane.

As in the previous cases, the set of points where the maximum value is attained for $\theta \mid \mathbf{ed}$ is determined by the tangency points between the segment **ed** and a toroidal lunoid $T^*(\mathbf{a}, \mathbf{b})$. The tangency set between **ed** and the toroidal lunoid $T^*(\mathbf{a}, \mathbf{b})$ can be realized at one of the endpoints of **ed**. Then, the maximum value can be defined by **e** or **d**, or by one of the critical points for $\theta \mid L(\mathbf{e}, \mathbf{d})$. Let $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$ be the critical points for $\theta \mid L(\mathbf{e}, \mathbf{d})$.

Lemma 4.5.1: The function $\theta \mid I$ reaches its maximum value at one of the endpoints of I or at one of the points $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\}$ which define the critical points for $\theta \mid L(\mathbf{e}, \mathbf{d})$.

Proof: Let $\mathbf{Y} = \mathbf{X}_1$ be the point that defines the solution for the corresponding Segment-Line problem. If $\mathbf{Y} \in I$, then $\mathbf{X} = \mathbf{Y}$ is the solution. Otherwise **X** lies at one of the points in the set $\{\mathbf{X}_2, \mathbf{X}_3, \mathbf{e}, \mathbf{d}\}$.

Algorithm SSNO: Segment-Segment Problem

Input: A segment **ab** in E^3 and a closed interval I=[**c**, **d**] $\subset H$, such that **ab** $\cap I = \varnothing$ and **ab** $\cap H = \varnothing$.

Output: A point **X** such that θ(**X**) = *ang*(**aXb**) is maximum when **x** is allowed to move on I.

Let **Y** = $\mathbf{X}_1$ be the point where $\theta \mid L$(**e**, **d**) reaches its maximum value.

If **Y** $\in I$ then exit with **X** = **Y**,

else

Exit with **X** such that $\theta(\mathbf{X}) = \max\ \{\theta(\mathbf{X}_2), \theta(\mathbf{X}_3), \theta(\mathbf{d}), \theta(\mathbf{e})\}$.

Theorem 4.5.1: Algorithm *SSO* finds in constant time a point **X** $\in I$ where $\theta \mid I$ attains its maximum value.

Proof: Find the point **Y** where $\theta \mid L$(**e**, **d**) attains its maximum value in constant time using *SLNO* algorithm. The test to find out if **Y** belongs to the interval is determined in $O(1)$ time. Finding the corresponding values of $\theta(\mathbf{X}_2)$, $\theta(\mathbf{X}_3)$, $\theta(\mathbf{d})$ and $\theta(\mathbf{e})$ is done in constant time. The comparison between these four values is also done in $O(1)$ time. Therefore, algorithm *SSNO* finds in constant time a point where $\theta \mid I$ attains its maximum value. Q. E. D.

4.6 The Segment-Convex Polygon Problem

Let **ab** be a segment in E^3 and let P be an n-convex polygon in E^3 not intersecting **ab**. Suppose the polygon is stored in an array. Determine a point **X** $\in P$ such that θ(**X**) = *ang* (**aXb**) is the maximum value of θ(**x**) when **x** $\in P$, in this case we denote the aperture angle function by $\theta \mid P$.

We develop this section by cases, when **ab** is orthogonal to $H \supset P$ and when it is not. Assume that P lies on the plane $H = \{(x, y, z)\colon z = 0\}$ and that the segment lies in the open-upper-half-space defined by H. If $H \cap$ **ab** $\neq \varnothing$ then the solution set lies on the boundary of P. Thus, we also analyze the particular situation in which **x** is constrained to roam on the boundary of P.

First we prove a basic lemma that will be used frequently, which concerns the determination of whether a given point lies in the interior of a given n-convex polygon in $O(\log n)$ time. It is well known that given an n-convex polygon stored in an array, it is possible to construct from it a data structure, in $O(n)$ time and space, such that subsequently point inclusion queries can be determined in $O(\log n)$ time. Two such data structures are star-descomposition [PS88] and the balanced hierarchical decomposition [Me84]. We strengthen these results for a polygon stored in a simple array.

Lemma 4.6.1: Given an n-convex polygon P in E^3 stored in an array and given a point q, finding whether q lies in P can be determined in $O(\log n)$ time.

Proof: Let L_L and L_R denote the vertical lines through the left-most and right-most points of P. These points partition P in two convex chains P_{up} and P_{down}, respectively. If q lies to the left of L_L or to the right of L_R, then q does not lie in P. Suppose c lies in between L_L and L_R. Perform binary search among the x coordinate of the vertices of P_{up} to locate q within a slab determined by an edge of P_{up}. Find the intersection points of the vertical projections of p_i and p_{i+1} onto P_{down} and call these points z_1 and z_2. Perform again binary search on the section of P_{down} that lies between z_1 and z_2, to locate q inside a slab determined by an edge of P_{down}. If q lies between two edges then it lies in P, otherwise it lies outside P. The correctness is immediate. By using Chazelle and Dobkin [CD87], the lines L_L and L_R as well as the intersection points z_1 and z_2 can be found in $O(\log n)$. Q. E. D.

We turn now to the problem of finding a point in the set $M_P = \{X \in P: \theta(X) \text{ is the maximum value of } \theta \mid P\}$.

<u>*The segment ab is orthogonal to P*</u>

Let $C(c, R)$ be the solution for the segment-plane case and let $M^* = P \cap C(c, R)$.

Lemma 4.6.2: Let ab be a segment which is orthogonal to the plane H and let x be a point on the plane constrained to lie in a polygon P. Then computing a point $\mathbf{X} \in P$ such that $\theta(\mathbf{X})$ is maximum has time complexity $\Theta(n)$.

Proof: First of all we have to verify whether M^* is empty or not. In order to obtain an answer, we test if $bd(P) \cap C(c, R)$ is empty or not by verifying if each edge of P intersects $C(c, R)$. If $bd(P) \cap C(c, R) \neq \varnothing$ then the intersection points on $C(c, R)$ describe all the arcs of $C(\mathbf{c}, R)$ that form $M_P = M^*$. Thus, any point in the intersection set M^* defines the maximum value. Hence, assume $C(\mathbf{c}, R)$ does not intersect $bd(P)$. Three cases arise:

Case 1) If $C(\mathbf{c}, R)$ is contained in the interior of P. Then, $M^* = C(\mathbf{c}, R)$ is the solution set M.

Case 2) If P is totally contained in the interior of $C(\mathbf{c}, R)$, then $M^* = \emptyset$. The solution is a vertex and it is determined by the furthest point of $bd(P)$ from $\mathbf{c}$.

Case 3) If P and the disk defined by $C(\mathbf{c}, R)$ are disjoint sets, the solution is determined by computing the nearest point of the boundary of P from $\mathbf{c}$.

Consider the complexity of this construction. To verify the intersection of each edge of P with $C(\mathbf{c}, R)$ is done in $O(n)$ time for all edges. To distinguish between the three cases, i.e., to determine whether $C(\mathbf{c}, R)$ lies completely inside or outside of P, it suffices to take any point on $C(\mathbf{c}, R)$ and test it for inclusion in P (by Lemma 4.6.1) this can be done in $O(\log n)$ time. Finally to test whether P lies completely inside or outside of $C(\mathbf{c}, R)$, it suffices to take any vertex of P and test it by inclusion in $C(\mathbf{c}, R)$, which can be done in constant time. In case 2) the furthest point of $bd(P)$ from $\mathbf{c}$ may be computed in $O(n)$ time by using Algorithm *SSO* in the n edges. In the latest case the nearest point of the boundary of P from $\mathbf{c}$ can be computed in $O(\log n)$ time with the algorithm of Edelsbrunner [Ed85]. Therefore, $O(n)$ suffices for the entire algorithm. Since the problem of computing $\mathbf{X} \in P$ such that $\theta(\mathbf{X})$ is maximum is equivalent to computing the maximum distance from a point to a convex polygon in the plane, and the latter problem has complexity $\Omega(n)$ [Ed85], the lemma follows. Q. E. D.

The following algorithm finds a point $\mathbf{X} \in P$ such that $\theta(\mathbf{X})$ is the maximum value of $\theta \mid P$, its time complexity and the correctness of the algorithm follows from Lemma 4.6.2.

Algorithm SPO: Segment-Polygon Problem Orthogonal Case

Input: A convex polygon P on a plane H that is stored as an array. A segment $\mathbf{ab}$ that neither intersects P nor H, and is orthogonal to H.

Output: A point $\mathbf{X}$ such that $\theta(\mathbf{X})$ is the maximum value of $\theta \mid P$.

Begin

Step 1.- Solve the problem when x is allowed to move on H. Let $C(\mathbf{c}, R)$ be the solution set.

Step 2.- Test whether $C(\mathbf{c}, R)$ intersects P.

If $C(\mathbf{c}, R) \cap bd(P) \neq \emptyset$ then exit with $\mathbf{X} = \mathbf{Y} \in C(\mathbf{c}, R) \cap bd(P) \neq \emptyset$.

else

Test whether $C(\mathbf{c}, R) \subset P$, $C(\mathbf{c}, R) \supset P$, or $Disk(C(\mathbf{c}, R)) \cap P \neq \emptyset$.

If $C(\mathbf{c}, R) \subset P$ then exit with a point $\mathbf{X} \in C(\mathbf{c}, R)$.

If $C(\mathbf{c}, R) \supset P$ then exit with a vertex p^* in P such that $\| p^* - \mathbf{c}\|_2 = \max \{\|p_i - \mathbf{c}\|_2 : p_i$ is a vertex of $P\}$.

If $Disk(C(\mathbf{c}, R)) \cap P \neq \emptyset$ then exit with a vertex p^* in P such that $\|p^* - \mathbf{c}\|_2 = \max \{\|p_i - \mathbf{c}\|_2 : p_i$ is a vertex of $P\}$.

End

Lemma 4.6.3: Let **ab** be an orthogonal segment to the *xy*-plane (and above it). Then computing a point $\mathbf{X} \in bd(P)$ such that $\theta(\mathbf{X})$ is the maximum value of $\theta \mid P$ has time complexity $\Theta(n)$.

Proof: If P is totally contained in $C(\mathbf{c}, R)$, or if $Disk(C(\mathbf{c}, R)) \cap P \neq \emptyset$, or if $bd(P)$ intersects $C(\mathbf{c}, R)$ we proceed as in Lemma 4.6.2. Suppose $C(\mathbf{c}, R)$ is totally contained in P. The solution point may be found by computing the nearest point of the boundary of P from **c**. Then, we can test every edge of P to find the point that minimizes the distance. Therefore, $O(n)$ suffices to find a point $\mathbf{X} \in bd(P)$ such that $\theta(\mathbf{X})$ is the maximum value of $\theta \mid P$. To show that the lower bound on the time complexity is $\Omega(n)$ it suffices to build a convex polygon obtained by pulling an arbitrary vertex of a regular convex polygon contained in P. Then, every edge must be visited to ensure that any point in $\{\mathbf{X} \in bd(P): \theta(\mathbf{X})$ is the maximum value of $\theta \mid P\}$ is not missed in the search. Q. E. D.

The segment ab is non-orthogonal to P

We turn now to the problem of computing a point $\mathbf{X} \in P$ such that the aperture angle function evaluated at this point brings the maximum value of $\theta \mid P$ when the given segment **ab** is not orthogonal to H. In order to characterize the solution and to find the method of solution, we use the results found for the segment-segment case.

Lemma 4.6.4: Let **ab** be a non-orthogonal segment to the *xy*-plane (and above it). Then computing $\mathbf{X} \in P$ such that $\theta(\mathbf{X})$ is the maximum value has time complexity $\Theta(n)$.

Proof: By Corollary 4.3.1 the solution for the segment-plane problem is determined by a unique point **X**. Testing whether P contains **X** is done in $O(\log n)$ time by using Lemma 4.6.1. If P contains **X** the solution for $\theta \mid P$ is **X** itself. Otherwise, the solution can be found by determining the maximum value for each edge in P and defining **X** as the maximum over all these local maxima. Finding the maximum value in a set of n values is performed in $O(n)$ time. Then, computing a point $\mathbf{X} \in P$ such that $\theta(\mathbf{X})$ is the maximum value has time complexity $O(n)$. Since we are considering as evaluation of the algorithm the time complexity in the worst case, the lower bound used when **ab** is perpendicular to P is also valid in this case. Q. E. D.

By Lemma 4.6.4 we may obtain the following algorithm whose time complexity is $\Theta(n)$.

In the instance where **x** is constrained to be in the boundary of P, if $H \cap \mathbf{ab} \neq \emptyset$, the interesting case is when the maximum is attained in the interior of P. However, this problem can be solved by finding the points where the maximum value is reached for each edge. Then, the the global maximum is attained in the point that defines the maximum value over the set of local maxima.

Algorithm SPNO: Segment-Polygon Problem Non-Orthogonal Case

Input: A convex polygon P on a plane H stored as an array. A segment **ab** that neither intersects P nor H and that is non-orthogonal to H.

Output: A point **X** in P such $\theta(\mathbf{X})$ that is the maximum value of $\theta \mid P$.

Begin

Step 1.- Solve the problem when **x** is allowed to move on H. Let **Y** be a point where the maximum is reached.

Step 2.- Test whether **Y** is in P.

If $\mathbf{Y} \in P$ then exit with $\mathbf{X} = \mathbf{Y}$.

else

Find a point $\mathbf{Y}_i$ for each edge e_i in P such that $\theta(\mathbf{Y}_i)$ is maximum and exit with $\mathbf{X} = \mathbf{Y}^*$, where $\theta(\mathbf{Y}^*) = \max\{\theta(\mathbf{Y}_i)\}$.

End

4.7 The Segment-Convex Polyhedron Problem

Let **ab** be a segment and let **K** be a convex polyhedron in E^3 not intersecting **ab**. Determine a point $\mathbf{X} \in \mathbf{K}$ such that $\theta(\mathbf{X}) = ang(\mathbf{aXb})$ is the maximum value of $\theta(\mathbf{x})$ for $\mathbf{x} \in \mathbf{K}$. In this case the aperture angle function is denoted by $\theta \mid \mathbf{K}$. We assume that the polyhedron is represented in the form of a *double-connected-edge-list* (*DCEL*). This is a data structure used to represent planar graphs, and since a convex polyhedron is isomorphic to a 3-connected and planar graph (Steinitz's Theorem, refer to [BL91]), the corresponding structure may be used to represent a convex polyhedron. There is a brief description of such a structure in [PS88]. This structure is implemented by using six linear arrays. Thus, it corresponds to the data structure used to represent convex polygons in the previous section.

First of all we state that the maximum value must be attained at some point in the boundary of **K**. For each point **y** in the interior of **K** the rays that define the aperture angle $ang(\mathbf{ayb})$ intersect the boundary of **K**. Then, it is possible to determine a point **x** in $bd(\mathbf{K})$ such that $ang(\mathbf{axb})$ is bigger than $ang(\mathbf{ayb})$.

Lemma 4.7.1: The maximum aperture angle $ang(\mathbf{aXb})$ is reached at a point **X** in $bd(\mathbf{K})$.

It turns out that we can restrict the set of points in $bd(\mathbf{K})$ to a subset. This subset is characterized as the set of points **x** in $bd(\mathbf{K})$ such that $int(\mathbf{xc})$ does not intersect the interior of **K** for each **c** in **ab**. In fact, it is the set of points from which **ab** is visible.

Lemma 4.7.2: The maximum aperture angle $ang(\mathbf{aXb})$ is reached at a point **X** in $bd(\mathbf{K})$ such that **ab** is visible from **X**.

Proof: Let **y** be a point that does not see **ab**. Then, there exists a point **c** in **ab** such that the segment **yc** intersects the interior of **K**. Thus, this segment not only intersects the interior but also the boundary of **K**. Therefore, there exists a point **x** in $bd(\mathbf{K})$ such that the segment **xb** does not intersects the interior of **K**. Then, $ang(\mathbf{axb})$ is bigger than $ang(\mathbf{ayb})$. Thus, for each point **y** that does not see **ab** there exists a point **x** which sees **ab** and whose aperture angle $ang(\mathbf{axb})$ is bigger than $ang(\mathbf{ayb})$. Q. E. D.

Let $V_{\mathbf{K}}(\mathbf{ab})$ be the set of points in **K** which can see the complete segment **ab**. To characterize this set consider for each 2-face $F_2(i)$ ($i = 1,\ldots,f_2$ where f_2 denotes the number of 2-faces) in **K** the support plane $H(F_2(i))$. If $H(F_2(i))$ separates **ab** from **K**, or if **ab** and **K** are weakly separated by $H(F_2(i))$ then $H(F_2(i))$ separates $\mathbf{K} \setminus F_2(i)$ from **ab**, or from $\mathbf{ab}\setminus\{\mathbf{c}\}$, where **c** denotes one endpoint of **ab**. Thus, in either case the interior of **K** is separated from **ab** for each one of those planes, and for each point **x** in each face $F_2(i)$ it sees **ab**. Therefore, if $H(F_2(i))$ separates **ab** from **K**, or if **ab** and **K** are weakly separated by $H(F_2(i))$, $F_2(i)$ is a subset of $V_{\mathbf{K}}(\mathbf{ab})$. In this way, $V_{\mathbf{K}}(\mathbf{ab})$ is the union of 2-faces $F_2(i)$ in **K** which are separated or weakly separated by a support plane of **K** at $F_2(i)$.

Consider now the complexity of computing for a given line segment and a convex polyhedron, which is disjoint to **ab**, the set $V_{\mathbf{K}}(\mathbf{ab})$. If we consider a more general problem where the segment **ab** is replaced by another convex polyhedron, $V_{\mathbf{K}}(\mathbf{ab})$ may be obtained by computing all the separating planes between the two convex polyhedra. Davis [Da85] used a projective transformation to convert this problem to a convex hull problem of two disjoint convex polyhedra. Then by using this transformation our problem is solved in $O(n)$ time, since finding the convex hull of two disjoint convex polyhedra is performed in $O(n)$ time. However, because we consider a segment **ab** instead of a convex polyhedron, the projective transformation is dispensed. For this special case the corresponding algorithm is a simpler one and has time complexity $O(n)$.

Lemma 4.7.3: Given a line segment **ab** and a disjoint n-convex polyhedron **K** stored in a DCEL, the region $V_{\mathbf{K}}(\mathbf{ab})$ may be computed in $O(n)$ time.

Finding the point where the maximum value is reached for $\theta \mid V_{\mathbf{K}}(\mathbf{ab})$ is done in linear time with respect to the number of edges for each face in $V_{\mathbf{K}}(\mathbf{ab})$. Then, the maximum value over $V_{\mathbf{K}}(\mathbf{ab})$ is computed in $O(n)$ time.

Lemma 4.7.4: Given a line segment **ab** and a disjoint n-convex polyhedron **K** stored in a DCEL, $\theta \mid$ **K** may have $\Omega(n)$ local maxima.

Proof: Let the line segment **ab** be positioned vertically over the origin of the xy-plane. Let $C(0, R)$ be the circle centered at the origin which is the solution for the segment-plane problem (see Fig. 4.7.1). Such a circle will contain a convex polygon P in its interior which is constructed as follows. Consider the concentric circle $C(0, \varepsilon)$ to $C(0, R)$ which radius is ε, for an ε-small positive value. Let $C(0, R - \varepsilon)$ be the concentric circle to $C(0, R)$ which radius is smaller than R (by an ε-small positive value). Let $p_1 p_n$ be a line segment parallel to the y-axis such that p_1 lies on the circle $C(0, R - \varepsilon)$ and in the first quadrant. The other point, p_n, also lies on $C(0, R - \varepsilon)$ but it is in the fourth quadrant. Place the set of points $p_2, \ldots, p_{n-1}$ in clockwise order on the portion of $C(0, R - \varepsilon)$ which lies to the right of the line $y = \varepsilon$ on the xy-plane. Then the polygon P mentioned earlier lies in the interior of $C(0, R)$ and is defined by $[p_1, p_2, \ldots, p_{n-1}, p_n]$. Because each edge $p_i p_{i+1}$ (subindex addition is done modulo n) lies on the $Disk(C(0, R))$ defined by $C(0, R)$, the line $L(p_i, p_{i+1})$ leaves the remaining vertices in one of the semiplanes defined by $L(p_i, p_{i+1})$, being P a convex polygon. The value of ε has been chosen in such a way that for the vertex p_i the *cone* defined by the semiplanes $H^+(p_{i-2}, p_{i-1})$ and $H^+(p_{i+1}\ p_{i+2})$, which are used to define P as a finite intersection of semiplanes, minus the $Disk(C(0, R))$, is non-empty. Let s be a point in the lower open half-plane defined by the xy-plane and whose orthogonal projection on the xy-plane is in the interior of the polygon defined by P. This point is placed in an ε-distance from the xy-plane. Then, the convex polyhedron is built by joining s to each vertex in $[p_1, p_2, \ldots, p_{n-1}, p_n]$. Since s lies in the exterior of a convex polygon (which is a polyhedral set) when s is connected to P, the new polyhedral set is a convex polyhedron. Observe that P is the set V_K(**ab**). By Lemma 4.7.2 the maximum value is attained at some point lying in P, i.e., finding the maximum value of θ(**x**) when **x** is allowed to move on **K** is reduced to computing the maximum value of $\theta \mid P$. This is equivalent to finding the maximum distance on the xy-plane from the origin to P. In this particular case such a distance is determined by each vertex of P, and any vertex of this polygon can be moved in the orthogonal direction by an ε-distance in order to obtain one global maximum. Because the vertex to be moved has been chosen arbitrary the algorithm to determine the maximum distance from a point to a convex polygon has to visit each vertex of P. Therefore, $\theta \mid$ **K** may have $\Omega(n)$ local maxima. Q.E.D.

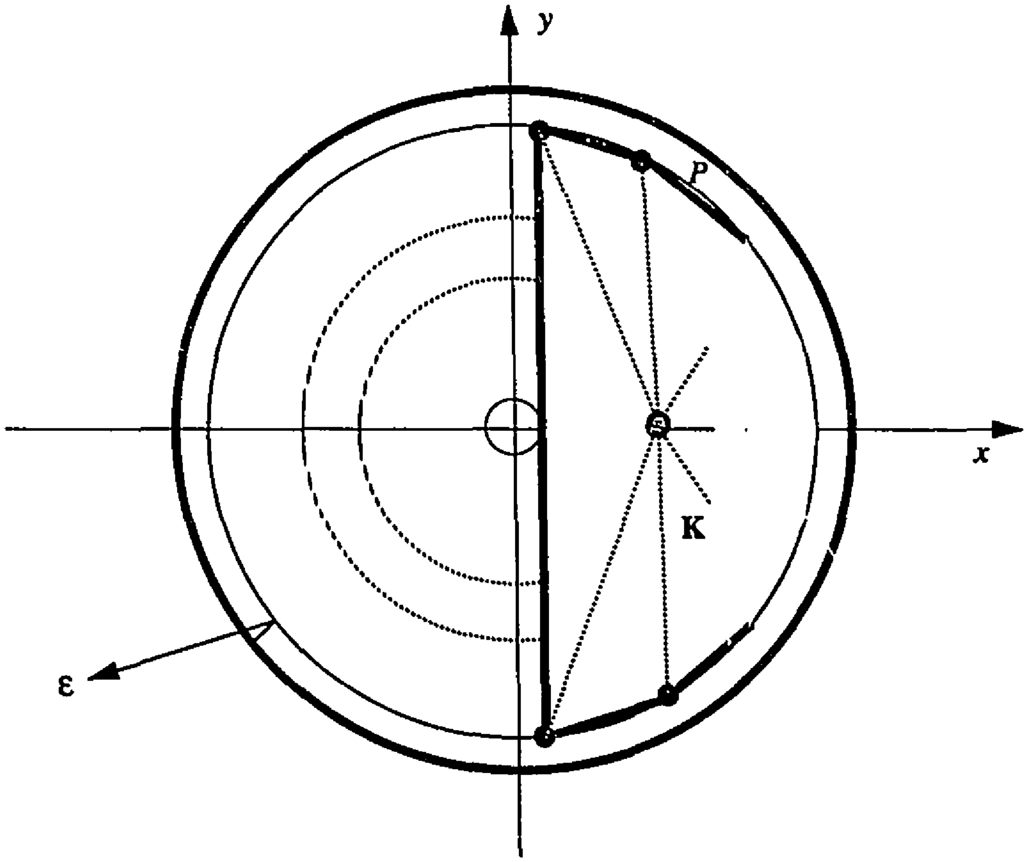

Fig. 4.7.1

As a consequence of lemmas 4.7.3 and 4.7.4 we state the following theorem.

Theorem 4.7.1: Given a line segment **ab** and a disjoint n-convex polyhedron **K** stored in a DCEL, $\theta(\mathbf{X}) = ang(\mathbf{aXb})$ may be computed in optimal $\Theta(n)$ time.

CHAPTER 5

CONCLUSIONS

In this thesis 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 present 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 a convex polygon P (n and m are the number of vertices, respectively). We also present algorithms whose complexities are $O(n + m)$ and $O(n \log m)$ for computing the *maximum* aperture angle with respect to Q. Thus, when $m = o(n \log n)$ the first algorithm is faster than the second one. However, if $m = \Omega(n \log^{1+\varepsilon} n)$, for any $\varepsilon > 0$, the second one is faster. Finally, we prove an $\Omega(n)$ time lower bound for the maximization and minimization problems, and an $\Omega(m)$ time lower bound for the minimization problem. Thus, the corresponding algorithm for the minimization problem is optimum.

In three dimensions we find the solution to the following problem: Given a convex polyhedron $\mathbf{K}$ and given a segment $\mathbf{ab}$ that does not intersect $\mathbf{K}$, find a point $\mathbf{X} \in \mathbf{K}$ such that the aperture angle defined as $ang(\mathbf{aXb})$ is maximum. Note that the corresponding aperture angle is defined as in the two dimensional case. We present a solution whose time complexity is linear with respect to the number of edges of $\mathbf{K}$, i. e., it is $O(n)$ if the corresponding convex polyhedron $\mathbf{K}$ has n vertices. We prove an $\Omega(n)$ time lower bound in order to prove the optimality of our algorithm.

To solve the problems defined in this thesis there exist many optimization techniques, then the reader may be interested to know why we use geometric properties to solve these problems instead of using one of these techniques.

First of all, the methods developed in this thesis do not need a specialized knowledge in the optimization field to understand them, they use basic tools from geometry.

Some optimization methods, such as Descendent methods or Lagrangian methods [GM74], [Po82] and [Be82], require an objective function which is twice differentiable. However, we proved that the function is non-differentiable at some point x_0 (Sec. 4.4).

Thus, alternative optimization methods of solution are those called *heuristic methods* [GMSW]. These methods provide an approximation to the solution and some times they do not find the global maximum (or minimum). Because we used an idealized model of computation the solution obtained in this thesis is exact.

Finally, since the heuristic methods are iterative methods and they use an approximation to the Hessian matrix at each iteration then the methods are unstable. Because the algebraic expression for our solution uses the basic operations (defined for our model of computation) which are well understood, our method of solution is stable.

The efficiency of our algorithm versus the corresponding efficiency of the heuristic methods is an open problem if we use clock-time to measure such efficiency.

APPENDIX

A cubic equation $(Ax^3 + 3Bx^2 + Cx + D) = 0$ can be reduced to $y^3 + py + q = 0$ by replacing $x = y + \frac{B}{2A}$, where p and q are defined by:

$$p = \frac{4AC - 3B^2}{4A^2} \text{ and } q = \frac{4A^2D + 2ABC - B^3}{4A^3}.$$

The solutions of an equation of degree three are given by Cardan's formulas [Us48]. Since the solution of the reduced equation, whose variable is y, involves a square root of a function $\delta(p, q)$, it is important to test whether it is positive, negative or zero. Let $\delta(p,q) = \frac{q^2}{4} + \frac{p^3}{27} = \frac{4p^3 + 27q^2}{108}$ be such a function.

If $\delta = 0$ then $y_1 = 2\left[\frac{-q}{2}\right]^{1/3}$, $y_2 = \left[\frac{q}{2}\right]^{1/3}$ and $y_3 = \left[\frac{q}{2}\right]^{1/3}$.

If $\delta > 0$ there is a pair of conjugate complex roots.

$$y_1 = \left[-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}\right]^{1/3} + \left[-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}\right]^{1/3}$$

$$y_2 = -\frac{\left(\left[-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}\right]^{1/3} + \left[-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}\right]^{1/3}\right)}{2} +$$

$$i\frac{\left(\left[-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}\right]^{1/3} - \left[-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}\right]^{1/3}\right)}{2}\sqrt{3} \text{ and}$$

$$y_3 = -\frac{\left(\left[-\frac{q}{2}+\sqrt{\frac{q^2}{4}+\frac{p^3}{27}}\right]^{1/3}+\left[-\frac{q}{2}-\sqrt{\frac{q^2}{4}+\frac{p^3}{27}}\right]^{1/3}\right)}{2} -$$

$$i\frac{\left(\left[-\frac{q}{2}+\sqrt{\frac{q^2}{4}+\frac{p^3}{27}}\right]^{1/3}-\left[-\frac{q}{2}-\sqrt{\frac{q^2}{4}+\frac{p^3}{27}}\right]^{1/3}\right)}{2}\sqrt{3}.$$

If $\delta < 0$ the solution is expressed by extracting cubic roots of $\frac{-q}{2}+i\sqrt{\frac{-q^2}{4}-\frac{p^3}{27}}$ trigonometrically.

The corresponding modulus is $\rho = \frac{-p\sqrt{-p}}{27}$ and the corresponding argument is determined by $\varphi = \operatorname{acos}\left(\frac{27q}{2p\sqrt{-p}}\right)$. Thus, having found ρ and φ the solution in this case is given by:

$$y_1 = 2\sqrt{\frac{-p}{3}}\cos\frac{\varphi}{3},\ y_2 = 2\sqrt{\frac{-p}{3}}\cos\left(\frac{\varphi}{3}+120°\right) \text{ and } y_2 = 2\sqrt{\frac{-p}{3}}\cos\left(\frac{\varphi}{3}+240°\right).$$

Thus, $x_1 = y_1+\frac{B}{2A}$, $x_2 = y_2+\frac{B}{2A}$ and $x_3 = y_3+\frac{B}{2A}$.

References

[Aal93] 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.

[AT81] Avis, D., and Toussaint, G. T., "An Efficient Algorithm for Descomposing a Polygon into Star-Shaped Pieces," *Patter Recognition* 13, 1981, pp. 295-298.

[Ag84] Aggarwal, A., "The Art Gallery Theorem: its Variations, Applications and Algorithmic Aspects. PhD Thesis," The John Hopkins Univerity, Baltimore, 1984.

[AHU74]Aho, A. V., Hopcroft, J. E., and Ullman, J. D., *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[AHU83]Aho, A. V., Hopcroft, J. E., and Ullman, J. D., *Data Structures and Algorithms*, AddisonWesley, 1983.

[Be82] Bertsekas, D., *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, Computer Science and Applied MathematicsG, 1982.

[Bal93] Bose, P., Guibas, L., Lubiw, A., Overmars, M., Souvaine, D. and Urrutia, J., "The floodlight problem," *Proceedings of the CCCG*, 1993.

[BL91] Bayer, M. M. and Lee C. W., "Convex Polytopes," in *Handbook of Convex Geometry*, Ed., Gruber, P. and Wills J., Editorial?, pp. ?, 1991.

[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.

[Ch91] Chazelle, B., "Triangulating a Simple Polygon in Linear Time," *Discrite and*

Computational Geometry", Vol. 6, No. 5, 1975, pp 485-524.

[Ch75] Chvatal, V., "A Combinatorial Theorem in Plane Geometry," *Journal of Combinatorial Theory Series*, B 18, 1975, pp 39-41.

[CLR90] Cormen, T., Leiserson, C., and Rivest R., *Introduction to Algorithms*, McGraw Hill, 1990.

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

[CRU93] Czyzowicz, J., Rivera-Campo, E. and Urrutia, J.,"Optimal Floodlight Illumination of Stages," *Proceedings of the CCCG*, 1993.

[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.

[Da85] Davis, G., "Computing separating planes for a pair of disjoint polytopes," *Proc. ACM Symposium on Computational Geometry*, Baltimore, Maryland, 5-7, June 1985, pp. 8-14.

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

[Ed85] Edelsbrunner, H., "Computing the minimum distance between two convex polygons," " *Journal of Algorithms*, vol. 6, 1985, pp. 213-224.

[EOW84] Edelsbrunner H., O'Rourke and Welzl E., "Stationing Guards in Rectilinear Art Galleries," *Computer Vision Graphics and Image* 27, 1984, pp. 167-176.

[Fi78] Fisk, S., "A short Proof of Chvatal's Watchman," *Journal of Combinatorial Theory Series*, B 24, 1978, p. 374.

[GM74] Gill, P. E. and Murray W., *Numerical Methods for Constrained Optimization*, Academic Press, 1974.

[GMSW] Gill, P. E., Murray W., Saunders, M. A. and Wright, M. H., "Linearly constrained optimization," in *Numerical Optimization 1981*, Ed., M. J. D. Powell, Academic Press, 1982, pp. 123-139.

[Hu93] Hurtado, F. A., *Problemas Geométricos de Visibilidad*, Doctoral Thesis, Universidad Politécnica de Catalunya, Barcelona, 1993.

[KKK83] Kahan J., Klawe M. and Kleitman D., "Traditional Galleries Requiere Fewer Watchmen," *SIAM J.*, Algebraic Dicrete Methods 4, 1984, pp. 194-206.

[Kn73] Knuth, D. E., *The Art of Computer Programming*, Addison-Wesley, 1973.

[LL86] Lee D. and Lin A., "Computational Complexityof Art Gallery Problems," *IEEE Trans. on Information Theory*, IT-32, 1986, pp. 276-282.

[Me84] Melhorn, K., *Data Structures and Algorithms*, Springer-Verlag, 1984.

[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.

[Po82] Powell, M. J., D., *Nonlinear Optimization*, Academic Press, Nato Conference Series, 1982.

[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," *Inform. Procees. Lett.*, vol. 23, No. 9, 1986, pp. 71-76.

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

[Sh77] Shamos, M.I., *Computational Geometry*, Ph. D. Dissertation, Yale University, 1977.

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

[Sl91] Slomson A., *An Introduction to Combinatorics*, Chapman and Hall Mathematics, 1991.

[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 "Rotating Calipers", *Proc. of IEEE MELCON*, Athens, Grece, May 1983.

[To92] Toussaint G. T., "What is Computational Geometry?", *Special Number of Proc. of IEEE* (G. T. Toussaint ed.), September 1992.

[Tu84] Tucker A., *Applied Combinatorics*, John Wiley and sons, 1984.

[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.

[Us48] Uspensky, J. V., *Theory of Equations*, McGraw-Hill, 1985, pp. 84-93.

[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, N0. 1, January 1994, pp. 1-16.