

**MAXIMUM INTERSECTION OF SPHERICAL
POLYGONS AND WORKPIECE ORIENTATION
FOR 4- AND 5-AXIS MACHINING**

Kai Tang
Tony Woo
The University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 89-34

December 1989

**Maximum Intersection Of Spherical Polygons
and
Workpiece Orientation for 4- and 5-axis Machining**

**Kai Tang
Tony Woo
The University of Michigan
Dept. of Industrial & Operations Engineering
Ann Arbor , MI 48109**

**Send correspondence to: Tony Woo
Dept. of Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117**

Abstract

Orienting the workpiece in such a way as to minimize the number of setups in a 4-axis or a 5-axis Numerical Control (NC) machine is formulated as follows: Given a set of spherical polygons (that are representations of curved surfaces visible to a 3-axis NC machine), find a great circle (the 4th axis) or a band (the 4th and the 5th axes) containing a great circle that intersects the polygons maximally.

While there are potentially infinitely many solutions to this problem, a sphere is partitioned into $O(N^2)$ regions based on the N polygons. Within each of these regions, it is shown that it requires $O(N \log N)$ time to determine maximum intersections and all the solutions are congruent. Central projection mapping is employed so as to present the algorithms in the plane.

1. Introduction

In CAD/CAM, a mechanical component is often represented by a group of surfaces. For a surface to be machined by a ball-end tool, the axis of the tool must not deviate with the normal vectors of the points on the surface by more than 90° . (See Figure 1.1.) We call this the *visibility constraint*: For a point to be visible by a tool, its orientations are delimited by the tangent plane at that point. Clearly, the visibility constraint may be less than 90° for some points on a surface. This leads to the notion of intersecting halfplanes at the point of interest, giving rise to a *cone* at a point. The intersection of all cones [6] at all points on a surface gives the set of tool orientations that meet the visibility constraint for the entire surface, which we call a *visibility cone*. The apex of a visibility cone is obtained by translating the end points of all the rays for all the points on the surface to the origin of some coordinate system. A visibility cone is unbounded as it implies that a tool may approach a surface from infinity. For computation purposes, it is more expedient to deal with bounded entities. Intersecting a unit sphere with a visibility cone (with its apex at the center of the unit sphere) gives a spherical polygon, or a *visibility map*, which is akin to the Gaussian map [2]. Varying a point in a Gaussian map by up to 90° on the unit sphere gives a cone of up to a hemisphere. The intersection of all such cones is the visibility cone of the surface, whose image on the unit sphere is the visibility map.

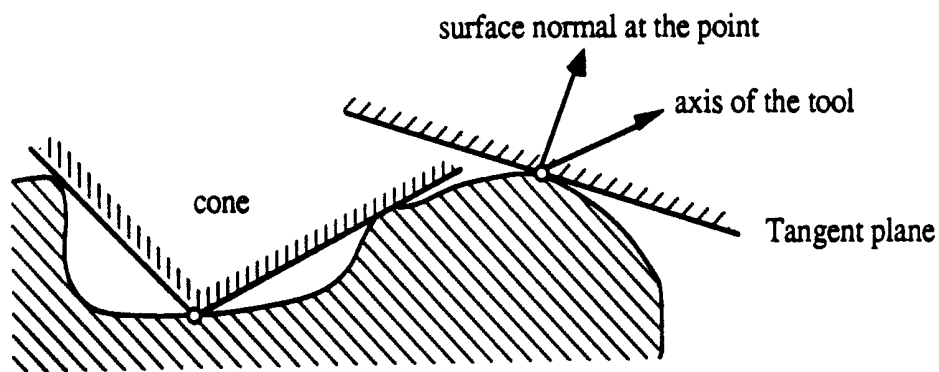


Figure 1.1 Visibility constraint of surface points

Implicit to the previous discussion on visibility is a 3-axis milling machine, that is, a tool for a 3-axis milling machine "sees" up to a hemisphere [2]. Figure 1.2 shows a sketch of a 4-axis NC machine. The tool moves in three principle directions X, Y and Z, and the platform rotates about the Y-axis. The 4th axis, because it allows the workpiece to rotate, may be represented on the unit sphere as a great circle. We call this great circle the

eigen circle. (See Figure 1.3.) Taken together, the visibility maps (3-axis) and the eigen circle (the 4th axis) give 4-axis machining.

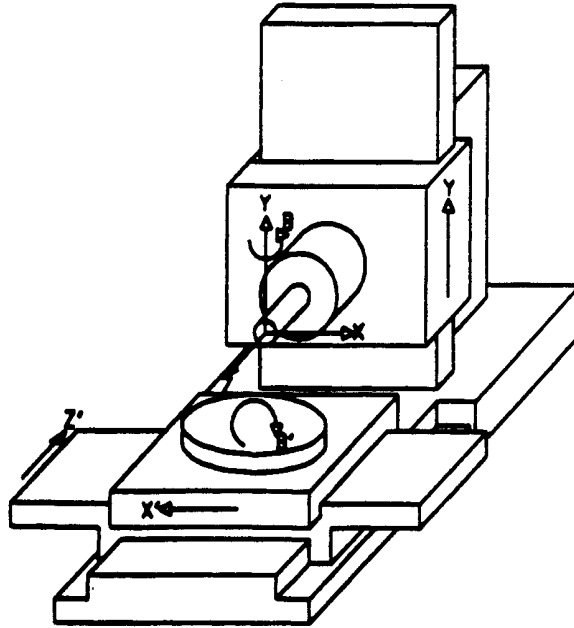


Figure 1.2 4-axis milling machine

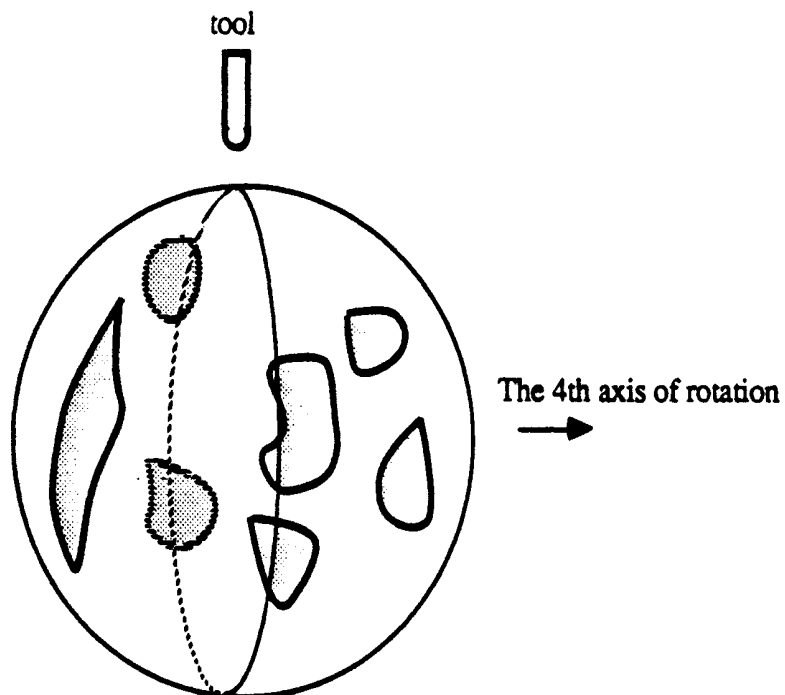


Figure 1.3 Intersecting the eigen circle with the Visibility Maps

Workpiece orientation on a 5-axis machine builds on the 4-axis case. The 5th axis, as shown in Figure 1.4, is orthogonal to the 4th axis. On the sphere, these two axes form a band, called an *eigen band*, assuming limited rotation for the 5th axis. (See Figure 1.5.) Since there always exists an eigen circle in an eigen band, we can focus on the solution for 4-axis without loss of generality.

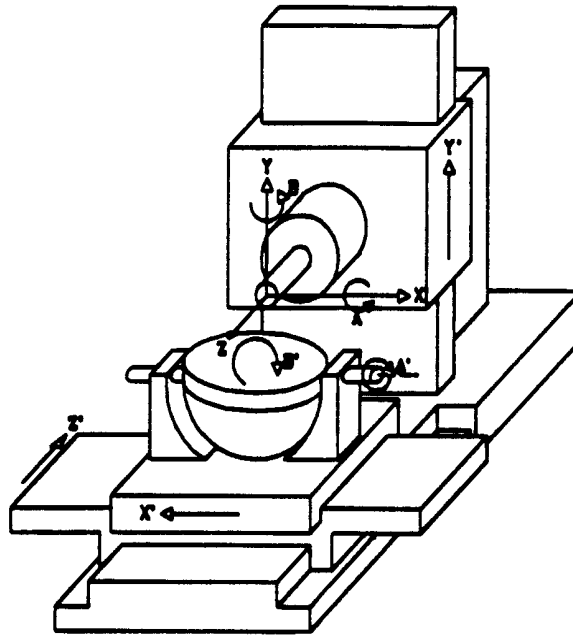


Figure 1.4 5-axis milling machine

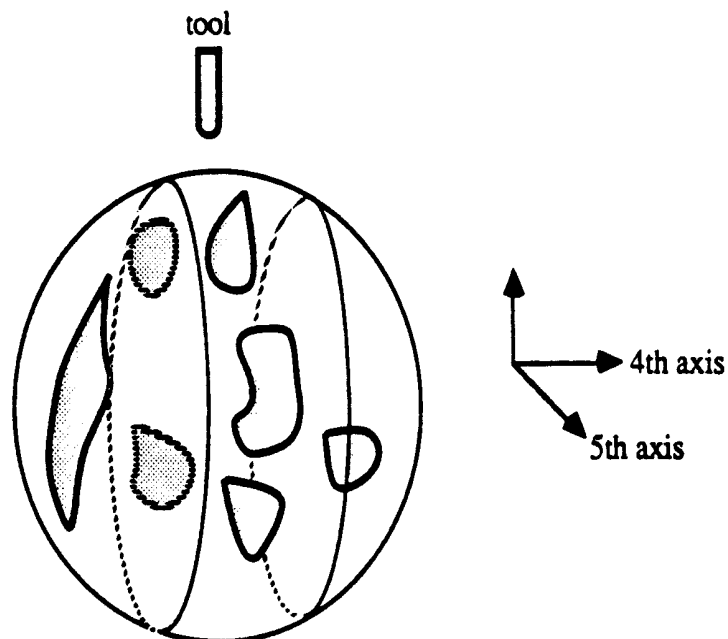


Figure 1.5 Covering visibility maps with the eigen band

Any point in a visibility map represents a tool orientation through which the entire surface can be machined. In order to minimize the number of mounting and dismounting of the work piece, which is non-productive, it is desirable that the eigen circle intersects as many visibility maps as possible, giving the maximum number of the surfaces can be machined for a single setup. This observation leads to the following problem.

Maximum Intersection of Polygons on the Sphere (MIPS): Given a set of polygons on the sphere, find a great circle that intersects them maximally.

MIPS is characterized by several difficulties - the least of which is the fact that the computation is performed on the sphere involving (spherical) polygons that are not necessarily convex and disjoint. Ominously, the problem domain is continuous. There are infinitely many solutions (of the eigen circle) possible. Of the ones that intersect the spherical polygons, there are again uncountably many solutions. From which, we desire one particular orientation of the eigen circle that intersects the given spherical polygons maximally.

To overcome the difficulty, we introduce the notion of a *via*. By letting a via be a point on the sphere (and insisting that the eigen circle passes through the via), we obtain an $O(V+N\log N)$ time solution, where N is the total number of polygons and V is the total number of their vertices.

But, there can still be infinitely many possible vias on the sphere. From the N spherical polygons, a partitioning is established. It is shown that there are only $O(N^2)$ partitions, from each partition the solutions are congruent. Combining the partitions with the vias leads to an $O(N^2V + N^3\log N)$ time solution to **MIPS**.

2. Approach and Structure

2.1. Central Projection

Projecting a unit sphere onto a plane offers many conveniences [2]. A great circle is mapped to a line; a point to a point. There are inconveniences as well: antipodal points (a pair of diametrically opposed points) have identical images, and certain bounded spherical polygons become unbounded when projected. In this section, these problems are dealt with, and the equivalence between the problem on the sphere and in plane is established. But first, we define some entities in spherical geometry. A mapping that transforms these geometric entities on the sphere to the plane and its various properties are discussed, which lead to the equivalence relation.

A *spherical polygon* is an area on the sphere whose boundary consists of ordered segments of great circles of the sphere. These segments are called the *edges* of the spherical polygon and any two consecutive segments intersect at a *vertex*. Figure 2.1 shows a spherical polygon with three edges. Notice the requirement that an edge of a spherical polygon must be a segment of a great circle, which implies that not any arbitrary area on the sphere can be a spherical polygon. This requirement suffices our problem, since a visibility map represented by a spherical polygon is the intersection of great circles [2].

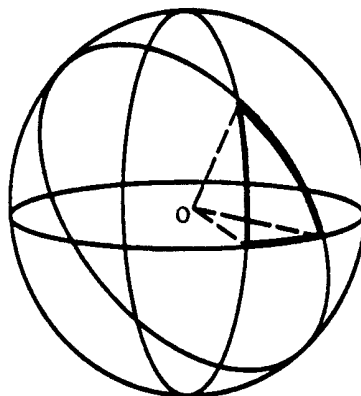


Figure 2.1. A spherical polygon

Next we examine the mapping between the unit sphere and the projection plane $Z=1$, called *central projection*. For an arbitrary point p on the sphere, let the line as defined by p and the origin intersect the plane $Z=1$, and call that intersection point the *image* p' of p . This mapping is unique but not one to one. In addition to the two-to-one mapping of antipodal points (that are diametrically opposite) as illustrated in Figure 2.2, central projection can be many-to-one. If a point p on the unit sphere is at the equator, i.e., its Z coordinate is 0, then its image is at infinity. With the caveat of the equator, two properties about central projection are noted without proof [3].

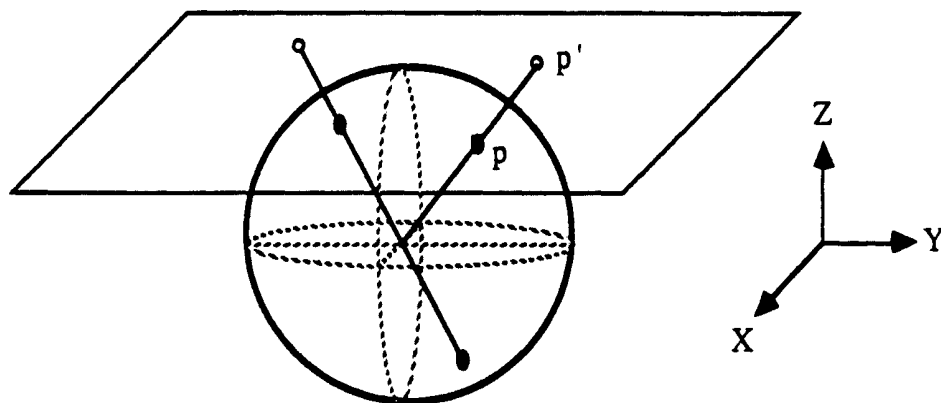


Figure 2.2. Central Projection Mapping

Lemma 2.1. The central projection between straight lines in the plane $Z=1$ and the great circles on the sphere, except its equator, is unique and one-to-one.

Lemma 2.2. The image of a spherical polygon is either a bounded polygon in the plane $Z=1$ if it does not cross the equator, or two unique unbounded polygons otherwise. (See Figure 2.3.)

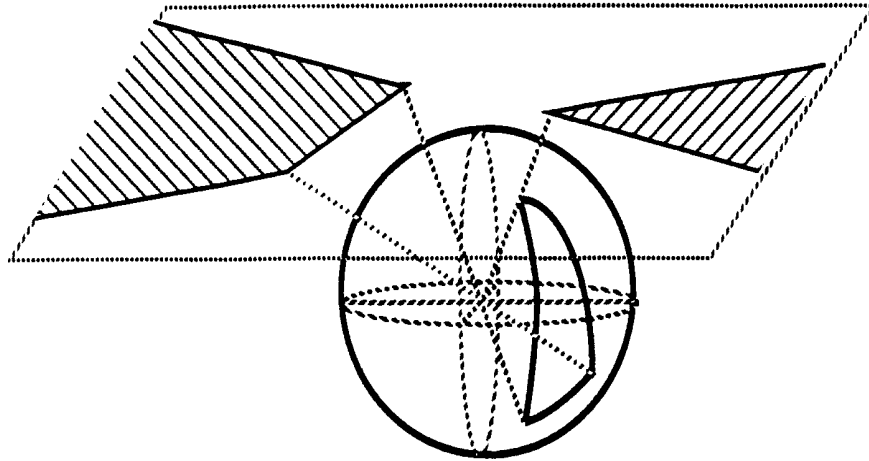


Figure 2.3. Central Projection of a spherical polygon that crosses the equator

Lemma 2.1 and Lemma 2.2 together assert that an eigen circle intersects a spherical polygon if and only if the corresponding images, a line and a polygon (polygons), intersect. We now formally define the problem of *Maximum Intersection of Polygons in the Plane* (MIPP) as:

MIPP: Given a set of simple polygons, (convex or otherwise, bounded or unbounded, overlapping or disjoint), find a line that intersects them maximally.

Now, **MIPP** is not yet a complete equivalence of **MIPS**, because the equator of the sphere has no image in the plane and some spherical polygons are mapped to two polygons in the plane whose intersection should be counted as one instead of two. The remedy for the former is straightforward. After **MIPS** is solved in the plane as **MIPP**, a special great circle - the equator - is invoked to intersect with the spherical polygons. This intersection is obviously in $O(E)$ time where E is the total number of edges of the spherical polygons. The special accounting for unbounded polygons will be described when the algorithms are developed. Therefore, **MIPS** can be completely solved on the sphere, if **MIPP** can be solved in the plane.

2.2. Intersection Via a Point and Via a Line

The main difficulty in solving **MIPP** is that the domain of the solutions is continuous; that is, there could be an infinite number of lines that intersect the same polygons maximally. We invoke the concept of a via. A via is one or more points in the plane such

that the solution (lines) to **MIPP** must pass through. Formally, the Maximum Intersection Problem in the Plane with Via (**MIPP_V**) is defined as follows.

MIPP_V: Let \mathcal{V} be a set of points in the plane. Given a set of polygons and the via \mathcal{V} in the plane, find a line through \mathcal{V} that intersects the polygons maximally.

MIPP could be viewed as a generalized **MIPP_V** with \mathcal{V} being the entire plane. Since a via limits the domain of the lines, the solution to a **MIPP_V** might not be the solution to the corresponding **MIPP**. The essence here is to find a finite set of vias such that the solutions to their **MIPP_V** cover the solutions to **MIPP**. If such a set of vias exist and their corresponding **MIPP_V** are solvable, then the solution to **MIPP** is clearly one of the solutions to the **MIPP_V**. In seeking such a set of vias, two special vias are crucial: one is when the via is a line, the problem of which will be denoted as **MIPP_L**, and the other is when the via is a single point, to be denoted as **MIPP_P**.

It is basic to geometry that two non-parallel lines in the plane always intersect. This fact implies that the solutions to **MIPP_L** of any two non-parallel lines cover the solutions to **MIPP**. Specifically, let H and H' be two non-parallel lines in the plane, L and L' be the two solutions (lines) to **MIPP_L** respectively, then the solutions to **MIPP** must be one of L and L' that intersects the greater number of polygons. Therefore, **MIPP** is readily solvable if **MIPP_L** can be solved.

Our goal here is to find *one* line that intersects the maximum number of polygons in the plane. Suppose we can find a partitioning that discretizes a line H into a finite set of intervals, and for each of these intervals the solutions to the **MIPP_P** for all the points in the interval are the same (in the sense that they intersect the same number of polygons). Then any *one* of them can be the solution to the **MIPP_V** with the via being that interval. In other words, the **MIPP_L** can be solved by picking an arbitrary point from each interval, solving their **MIPP_P**, and selecting one of the solutions that intersects the most number of polygons. This structural relation is described by Figure 2.4.

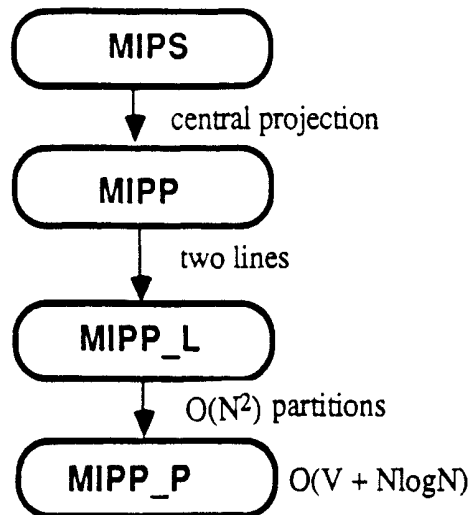


Figure 2.4 Structure of the solutions to **MIPS**

In the next section, the **MIPP_P** problem is solved and an $O(V + N \log N)$ time algorithm is given, where N is the number of polygons and V is the total number of their vertices. The problem of partitioning a line is studied in section 4, where we prove that only $O(N^2)$ intervals exist in such a partitioning. This leads to an $O(VN^2 + N^3 \log N)$ time algorithm to solve the **MIPP_L** problem. By the structural relation as shown in Figure 2.4, both **MIPP** and **MIPS** then are solved in $O(VN^2 + N^3 \log N)$ time.

3. Solution to MIPP_P

We now state the problem of maximum intersection of polygons via a point formally as **MIPP_P**: *Given a finite number of polygons in the plane, convex or otherwise, bounded or unbounded, overlapping or disjoint, and a point p , find a line that passes through point p intersecting as many polygons as possible.*

It is assumed that the point p is not within or on the boundary of the convex hull of any polygons. (Suppose it is inside or on the convex hull of a polygon P . Then any line passing through p will always intersect P which can be then removed from consideration.) Consider the condition under which a line intersects a polygon P . From a point p , there are two unique supporting lines [4] to P . When P is unbounded, either one of the lines is made parallel to an edge that extends to infinity (as in Figure 3.1(b)), or both lines are made parallel (as in Figure 3.1(c)). Clearly, any line passing through p and bounded by these two supporting lines is guaranteed to intersect P . We call such a pair of supporting lines the *butterfly* (centered at point p) of P .

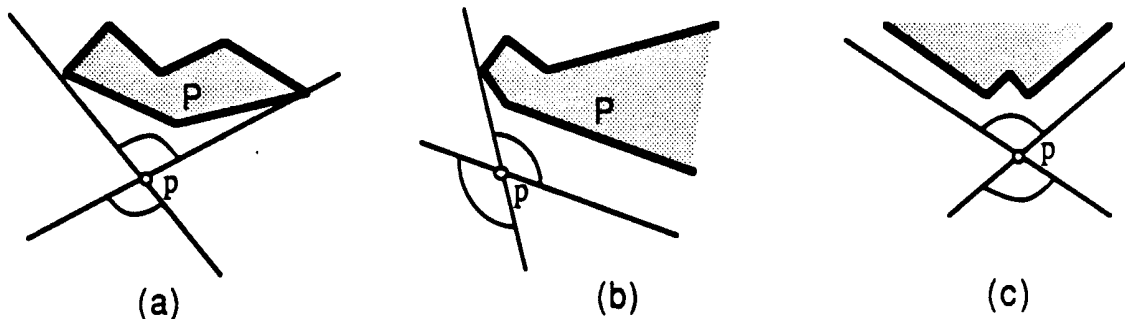


Figure 3.1 Supporting lines to a polygon from a point

To characterize the angle of the butterfly of P , let U^p be the halfspace that contains all the points on or above the horizontal line passing through a point p . The *bounding angle* (centered at p) of P characterize the butterfly (centered at p) of P in U^p . This bounding angle can be one or two disjoint angular intervals. As shown in Figure 3.2, if polygon P does not cross the horizontal line, its bounding angle is a single angular interval; otherwise its bounding angle becomes two angular intervals. Figure 3.2 also shows that *a line passing through a point p intersects polygon P if and only if it is within the bounding angle (centered at p) of P* . Thus, the bounding angle of P embodies all the lines that pass through p and intersect P . If a bounding angle is a single angular interval, its two rays, in

the counter-clockwise order, are marked by "IN" and "OUT". Similarly, the four rays of the two angular intervals are marked by "IN", "OUT", "IN", "OUT", in the counter-clockwise order.

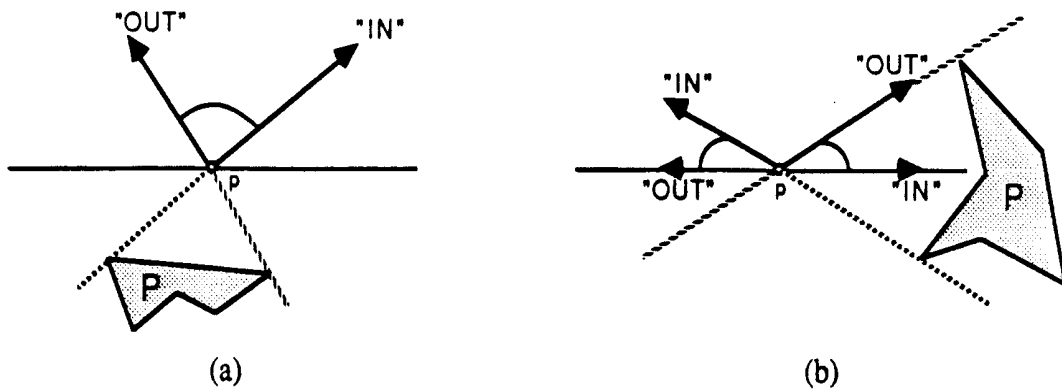


Figure 3.2 Bounding angle

The "IN" and "OUT" rays of all the polygons in the plane constitute a unique angular partition of UP , as illustrated in Figure 3.3. (For clarity, an "IN" ray of polygon P_i in Figure 3.3 is associated with the subscript "i" as IN_i ; similarly for the "OUT" rays.) The partition consists of a number of angular intervals whose two delimiting rays are the two angularly adjacent rays. These angular intervals are called the *homogeneous intervals*. For example, there are total of 7 homogeneous intervals in Figure 3.3: (IN_2, IN_1) , (IN_1, OUT_2) , (OUT_2, IN_3) , (IN_3, OUT_3) , (OUT_3, OUT_1) , (OUT_1, IN_2) , and (IN_2, OUT_2) . By the way in which the bounding angles are defined, we state the following lemma (the proof of which is omitted because of its obviousness).

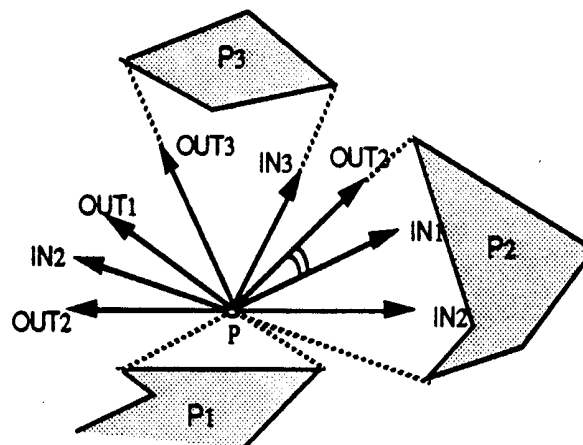


Figure 3.3 Angular partition and homogeneous intervals

Lemma 3.1. Any line within a homogeneous interval passing through a point p intersects the same set of polygons.

Lemma 3.1 assures that associated with each homogeneous interval there is a definite number of polygons. This number will be denoted as the NUM of that homogeneous interval (R_i, R_{i+1}) , with R_i and R_{i+1} being its two delimiting rays. The calculation of the NUM is expedited through the observation that it only depends on the NUM of the immediately previous homogeneous interval and the labeling of the rays. For instance, in Figure 3.3, $\text{NUM}(\text{IN}_1, \text{OUT}_2)$ is 2 as any line in this homogeneous interval must intersect P_1 and P_2 . $\text{NUM}(\text{OUT}_2, \text{IN}_3)$ can be updated as follows. The first delimiting ray of $(\text{OUT}_2, \text{IN}_3)$ is an "OUT" ray, which means "leaving" the homogeneous interval for polygon P_2 . Therefore $\text{NUM}(\text{OUT}_2, \text{IN}_3)$ is 1 by decrementing $\text{NUM}(\text{IN}_1, \text{OUT}_2)$.

The algorithm **MIPP_POINT** given below finds a homogeneous interval with maximum NUM. The algorithm takes as input the point p and the set of polygons P_1, P_2, \dots, P_N . The output is an index M , a sequence of R rays $\mathfrak{R}[1:R]$, emanating from p in the counter-clockwise order, and an integer array $\text{NUM}[1:R-1]$. Each angular interval $(\mathfrak{R}[i], \mathfrak{R}[i+1])$, $(i=1, 2, \dots, R-1)$, represents a homogeneous interval and the number $\text{NUM}[i]$ is the NUM of $(\mathfrak{R}[i], \mathfrak{R}[i+1])$, $(i=1, 2, \dots, R-1)$, with $\text{NUM}(M)$ being the maximum. For simplicity, it is assumed that no two rays of $\mathfrak{R}[1:R]$ are coincident.

Algorithm **MIPP_POINT** (p, P_1, P_2, \dots, P_N)

%

Compute the homogeneous intervals and their NUMs of the angular partition of a point p and a set of polygons P_1, P_2, \dots, P_N .

Output the homogeneous interval whose NUM is the maximum.

%

Begin

step 1. $P_1, P_2, \dots, P_K \leftarrow$ those of P_1, P_2, \dots, P_N which do not cover p
($K \leq N$);

step 2. $\mathfrak{R}[1:R] \leftarrow$ the "IN" and "OUT" rays of P_1, P_2, \dots, P_K ;

step 3. Sorts $\mathfrak{R}[1:R]$ in the CCW order and stores the sorted rays back in $\mathfrak{R}[1:R]$;

step 4. $\text{NUM}[1] \leftarrow 1 + (N-K)$;

$M \leftarrow 1$;

```

        NUMmax <-- NUM[1];
step 5.   for i=2 to R-1 do
            if (R[i] is "IN") then
                NUM[i] = NUM[i-1] + 1
            else
                NUM[i] = NUM[i-1] - 1
            end if;
            if (NUM[i] > NUMmax) then
                M <-- i
                NUMmax <-- NUM[i]
            end if;
        end do; {step 5}
step 6.   return {M, R[1:R], NUM[1:R-1]}
end {Algorithm MIPP_POINT}

```

Theorem 3.1. **MIPP_P** can be solved in $O(V + N \log N)$ time and $O(V + N)$ space, where V is the total number of the vertices of the polygons.

Proof. Constructing the convex hull of a polygon, either bounded or unbounded, takes linear time in the number of the vertices of the polygon. (Refer to [4] and the Appendix). Thus step 1 in **MIPP_POINT** takes $O(V)$ time since detecting whether a point is in the intersection of m halfplanes is certainly $O(m)$. Notice that if a point p is outside the convex hull of a polygon P then the two supporting rays from p to P are the same as the supporting rays from p to the convex hull of P . Since it takes linear time to find the two supporting lines from a point to a convex polygon [4], Step 2 requires at most $O(V)$ time. Step 3 is in time $O(R \log R)$ [4], where R is the number of rays. Step 5 requires only $O(R)$ time. Since R is at most $4N$, **MIPP_POINT** requires $O(V + N \log N)$ time. The linearity requirement in space is due to the fact that there are only $O(V + R)$ entities. Q.E.D.

Before leaving this section, consider the special case of unbounded polygons - that the intersection between a line and two associated unbounded polygons should be counted as one and not two, if they are the images of a spherical polygon crossing the equator. Let P_i and P_j be such a pair and α_i and α_j be their bounding angles (centered at p). If the

bounding angles of P_i and P_j are disjoint, then no remedial action need be taken since there is no line passing through p can intersect both P_i and P_j . (See Figure 3.4(a).) When the two bounding angles α_i and α_j overlap, we take the union $\alpha_i \cup \alpha_j$ of these two angles. Any line that passes through p and falls into this union is guaranteed to intersect either P_i or P_j . That is, this union is equivalent to the bounding angle of the union of P_i and P_j . (See Figure 3.4(b).) A remedy is therefore only needed in computing the "IN" and "OUT" rays of P_i and P_j at step 2 of the algorithm **MIPP_POINT**. That is, if the bounding angles of P_i and P_j overlap, then replace them by their union. Since this checking and replacing take constant time for a pair of polygons, the time and space requirements of algorithm **MIPP_POINT** after the modification will not be altered. (In the extreme case where all N spherical polygons are at the equator yielding $2N$ unbounded polygons in the plane, the equator will be the solution to **MIPS**.)

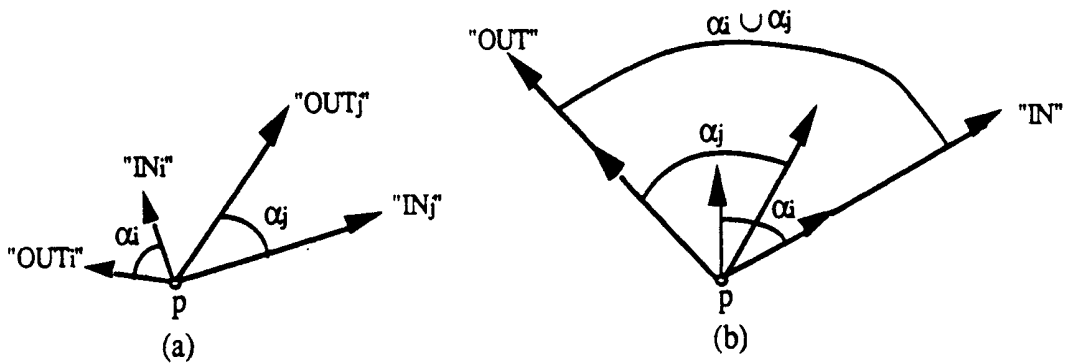


Figure 3.4 Treatment for unbounded polygons

4. Solution to MIPP_L and MIPP

Our strategy for solving MIPP is as follows: Given two arbitrary lines H and H' that are not parallel, we solve MIPP_L for H and H' . These two solutions cover the entire plane and hence one of them, which intersects a greater number of polygons, becomes the solution to MIPP. Having solved MIPP_P, we now move on to solving MIPP_L. This requires finding a partitioning of a line H into a finite set of intervals such that within each of these intervals the solutions to MIPP_P for all points in the interval are the same. For notation, a line L passing through a point p will be denoted as L_p . First, a definition of the congruence relation among points is useful.

Definition 4.1. Two points p and q are *congruent* to each other if the solutions to their MIPP_P intersect the same number of polygons. A set of points is a *congruent set* if all its points are congruent to one another.

In search of a partitioning of a line into a finite number of congruent sets, we start by considering only two polygons, say P and Q , in the plane. Let $CH(P)$ and $CH(Q)$ be the convex hulls of P and Q respectively, and $CH(P,Q)$ be the convex hull of P and Q . (If a polygon P is unbounded, then $CH(P)$ consists of an open chain of edges with the two end edges being two half lines.) The definition of a *homogeneous zone* is in order, which will directly lead to the computation of congruent sets.

Definition 4.2. A homogeneous zone of two polygons P and Q is a set of points such that for any two points p and q in the zone, there exists a line L_p intersects both P and Q if and only if there exists a line L_q intersecting both P and Q .

A homogeneous zone is derived from the observation that a line intersects a polygon if and only if it intersects its convex hull. If $CH(P)$ and $CH(Q)$ overlap, then for any point p in the plane, a line L_p that intersects the overlapping part of $CH(P)$ and $CH(Q)$ intersects both P and Q . Thus the entire plane is the homogeneous zone if $CH(P)$ and $CH(Q)$ are not disjoint.

Next, consider the special case when the convex hulls of two unbounded polygons P and Q are disjoint and a halfline edge of P is parallel to a halfline edge of Q . (See Figure 4.1.) Such two polygons are *parallel* to each other. If halflines E_p and E_q point in the same direction, under the central projection mapping, their inverse images are two circular

segments C_p and C_q which share the same vertex v on the equator of the unit sphere. (See Figure 4.2(a).) This shows that E_p and E_q should be counted as intersecting each other, only at infinity. When E_p and E_q have opposite directions, as shown in Figure 4.2(b), their inverse images C_p and C_q have a vertex each on the equator that are antipodal. Any great circle that passes through these two vertices clearly intersects C_p and C_q , and its image under central projection is a line parallel to E_p and E_q . As the result, if P and Q are parallel, then any line parallel to the two parallel halfline edges of P and Q intersects both P and Q , since the inverse image of this line intersects both the inverse images of P and Q . Therefore, the entire plane should be the homogeneous zone of two parallel polygons P and Q .

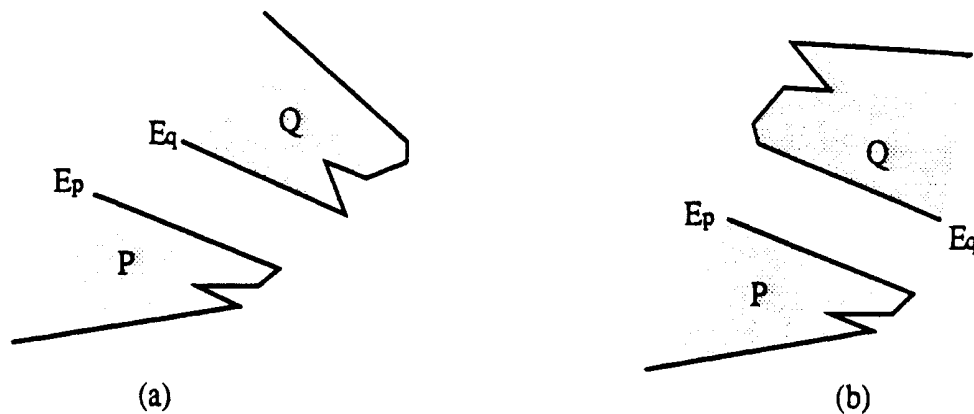


Figure 4.1 Parallel polygons

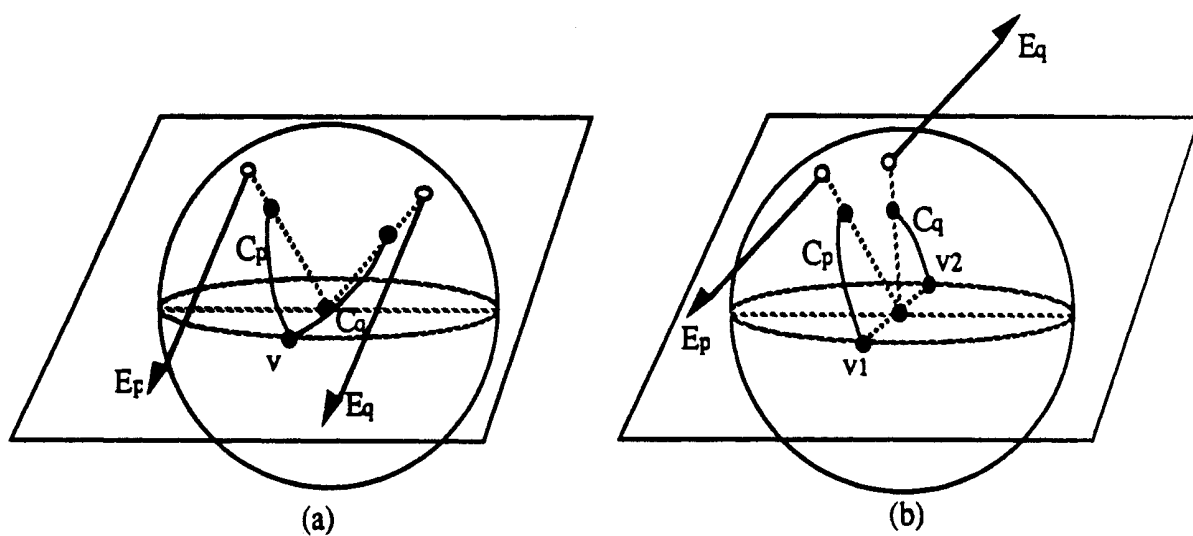


Figure 4.2 Intersection of two parallel halflines

In general, $CH(P)$ and $CH(Q)$ are disjoint and are not parallel. There exist up to four supporting lines between them, two internal and two external. For example, P and Q in Figure 4.3(a) have all four supporting lines. Figure 4.3(b) shows the supporting lines when one of P and Q is unbounded. Two unbounded polygons may have no external supporting lines as shown in Figure 4.3(c). It is known that if both P and Q are bounded and their convex hulls are disjoint, they always have four supporting lines, their two internal supporting lines are not parallel, and all the four supporting lines can be computed in linear time [5]. When only one of P and Q is bounded, it is shown in the Appendix that their four supporting lines always exist, the two internal common supporting lines are non-parallel, and all of them can be obtained in linear time. It is also proven in the Appendix that if two unbounded polygons P and Q are not parallel to each other, then there exist two internal non-parallel supporting lines and at most *one* external supporting line. Furthermore, the supporting lines of P and Q partition the plane into at most 16 zones. (As an example, the two polygons in Figure 4.3(a) form 15 such zones.) We next prove that each of these zones is a homogeneous zone.

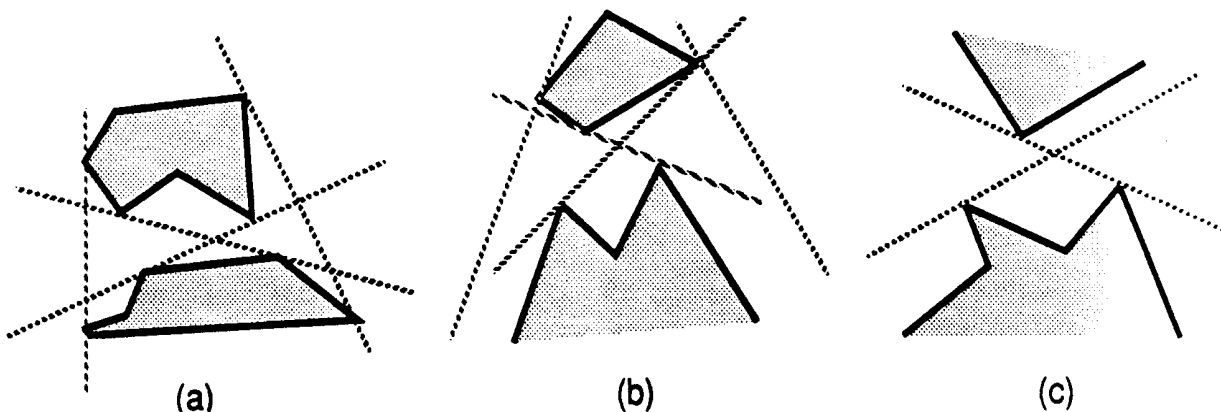


Figure 4.3. Supporting lines between two polygons

Lemma 4.1. Let P and Q be two non-parallel polygons whose convex hulls are disjoint. Then, the supporting lines of P and Q partition the plane into homogeneous zones of P and Q .

Proof. Let c be the intersection point between the two internal supporting lines of P and Q (which always exists by the proof in the Appendix and [5]). First, consider the case when both P and Q are bounded as in Figure 4.4(a). For any point p in $CH(P, Q)$,

there always exists a line L_p that intersects both P and Q . For a point q outside $CH(P,Q)$, in the zones ζ_3 or ζ_4 , the line passing through q and point c is seen to intersect both P and Q . For a point r outside $CH(P,Q)$ in zones ζ_1 or ζ_2 , no line through r will intersect both P and Q . (This is because an internal supporting line separates P from Q .)

Next, consider the case when one polygon is bounded but the other is not, as shown in Figure 4.4(b). For a point p in $CH(P,Q)$, there always exists a line L_p that intersects both P and Q . (Recall that two parallel lines intersect at infinity.) For a point q outside $CH(P,Q)$, in the zone ζ_3 , ζ_4 or ζ_5 , the line passing through q and point c intersects both P and Q . For a point r outside $CH(P,Q)$ in zone ζ_1 or ζ_2 , no line through r will intersect both P and Q .

Finally, suppose both P and Q are unbounded. There are two cases. First, consider the case when no external supporting line exists, as shown in Figure 4.4(c). Let R_1 , R_2 , R_1' , and R_2' be the four half lines (or rays) from the two internal supporting lines. Since P is unbounded, a line intersects P if it intersects either R_1 or R_1' . A similar observation holds for Q and half lines R_2 and R_2' . Thus, for any point p in the plane, a line L_p that intersects both R_1 and R_2 or both R_1' and R_2' is guaranteed to intersect both P and Q . Next, suppose only one external supporting line E exists. (See Figure 4.4(d).) Due to the unboundedness of P and Q , lines that are parallel to E and lie on the same side of E as P and Q are guaranteed to intersect both P and Q . For an arbitrary point p in the zone ζ_1 or ζ_3 , the line L_p passing through point c intersects both P and Q . Since the internal supporting lines separate P and Q from each other, there does not exist a line that intersects P, Q and ζ_2 . The proof is completed by noticing that all the zones $\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5$ in Figure 4.4 are supersets of the zones induced by the supporting lines of P and Q .

Q.E.D.

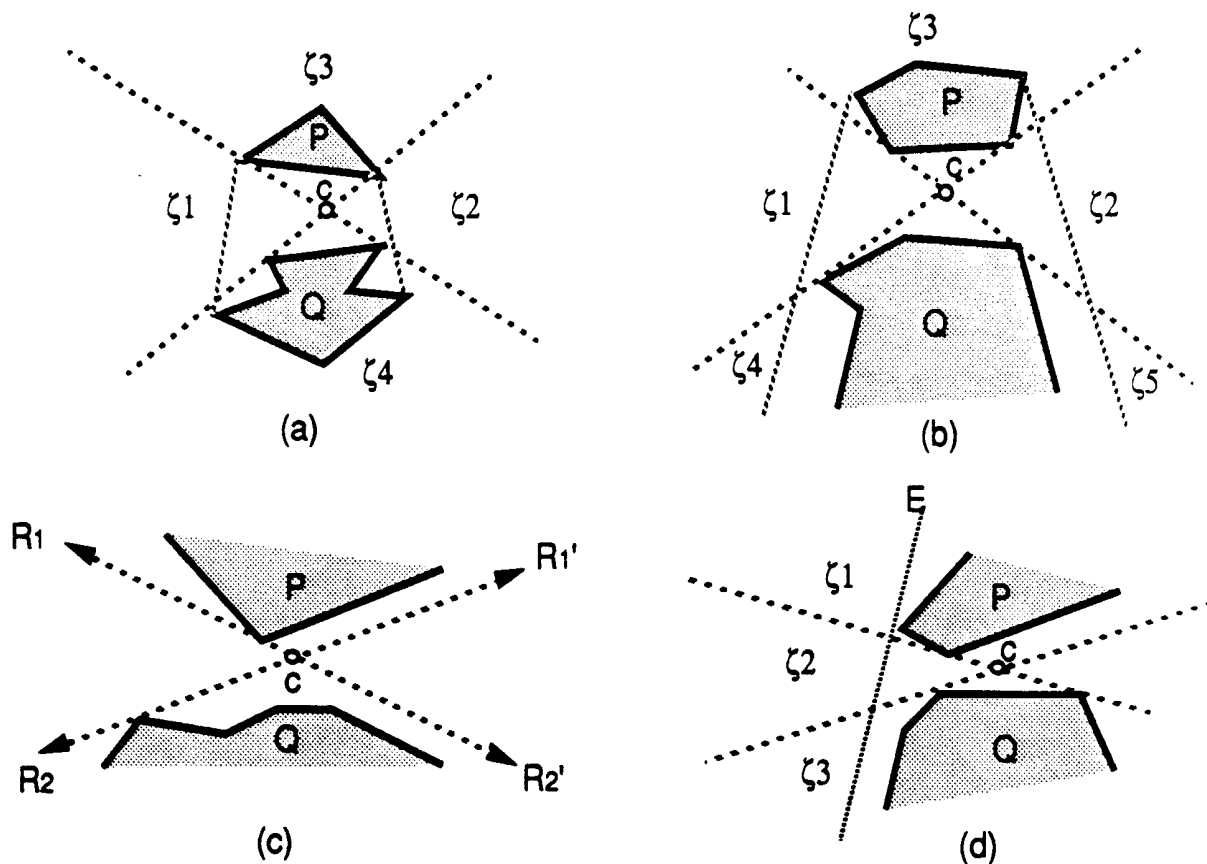


Figure 4.4 Proof of Lemma 4.1

With the consideration of parallelism and possible overlap, the following lemma summarizes our understanding of homogeneous zones for two polygons.

Lemma 4.2. The homogeneous zones of two polygons P and Q is the entire plane when $CH(P)$ and $CH(Q)$ overlap or when P and Q are parallel, or the zones as induced by the supporting lines of P and Q .

Lemma 4.2 implies that when there are only two polygons in the plane, their homogeneous zones are congruent sets. In order for this result to be useful in the general case when the number of polygons is greater than two, a broader definition of partitioning is necessary.

Definition 4.3. Let P_1, P_2, \dots, P_n be the n polygons in the plane, and HZ_{ij} be the set of homogeneous zones of P_i and P_j ($i \neq j$ and $i, j = 1, 2, \dots, n$). The intersection

$HZ_{1,2,\dots,n} = \bigcap HZ_{ij}$ ($i \neq j$ and $i, j = 1, 2, \dots, n$) are the homogeneous zones of P_1, P_2, \dots, P_n .

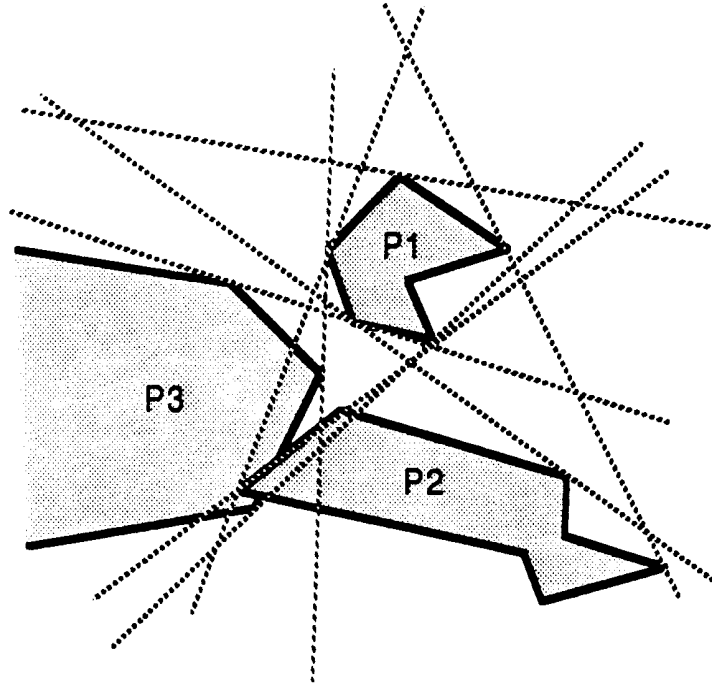


Figure 4.5 Homogeneous zones of three polygons

Figure 4.5 illustrates the homogeneous zones of three polygons. The upper bound of the number of the homogeneous zones of n polygons is $O(n^2)$ since there are at most four supporting lines between each pair of the polygons. By Definition 4.3, any homogeneous zone of P_1, P_2, \dots, P_n is also a subset of a homogeneous zone of P_i and P_j ($i \neq j$ and $i, j = 1, 2, \dots, n$). This *inclusion property* leads to the following theorem.

Theorem 4.1. Let P_1, P_2, \dots, P_n be the n polygons in the plane. Each homogeneous zone of P_1, P_2, \dots, P_n is a congruent set.

Proof. Let p and q be two arbitrary points in a homogeneous zone ζ . Without loss of generality, suppose one can find a line L_p that intersects P_1, P_2, \dots, P_k ($k \leq n$). We show there exists another line L'_q that intersects P_1, P_2, \dots, P_k as well. Let $\beta_1, \beta_2, \dots, \beta_k$ be the butterflies (centered at point q) of P_1, P_2, \dots, P_k respectively.

First we prove that for any three polygons P_i, P_j, P_k ($i \neq j \neq k$) there exists a line L'_q intersecting all three. Refer to Figure 4.6, where β is the intersection between two

butterflies β_i and β_j which must not be empty due to the inclusion property. Now, the intersection of β and a third butterfly β_k must not be empty either. For, if it is, either the intersection of β_k and β_i or the intersection of β_k and β_j is empty, which violates the inclusion property. Hence, any line L'_q bounded by the intersection of β and β_k intersects all three polygons P_i, P_j, P_k .

Now, suppose there exists a line L'_q that intersects P_1, P_2, \dots, P_{k-1} but no line passing through q can be found to intersect all P_1, P_2, \dots, P_k . Referring to Figure 4.6 again, let β be the intersection between $\beta_1, \beta_2, \dots, \beta_{k-1}$, which is not empty by assumption. Also, let P_i and P_j ($i, j \leq k-1$) be the two polygons that contribute to the delimiting lines of β . Because no line can be found to intersect P_1, P_2, \dots, P_k , the intersection of β and β_k must be empty. This implies, however, that no line passing through q can be found that intersects P_i, P_j, P_k all together. This is not possible by the first part of this proof. Therefore, the intersection of all the butterflies $\beta_1, \beta_2, \dots, \beta_k$ is not empty and as a result any line L'_q bounded by that intersection intersects all the polygons P_1, P_2, \dots, P_k . Q.E.D.

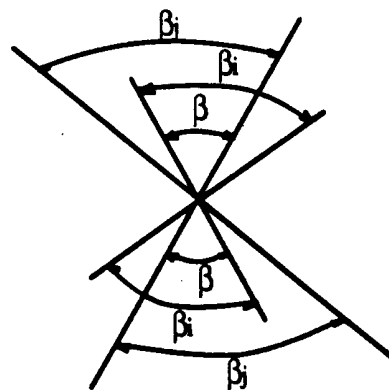


Figure 4.6. Proof of Theorem 4.1

Theorem 4.1 ensures that each homogeneous zone is a congruent set. Since a homogeneous zone is formed by the intersection of supporting lines (which are convex sets), it is convex. The intersections between a line H and these convex homogeneous zones form a finite set of congruent intervals on that line; see Figure 4.7. Capturing this congruence property, we now present the algorithm `MIPP_LINE` which finds a line that passes through a given line H and intersects the polygons P_1, P_2, \dots, P_n maximally.

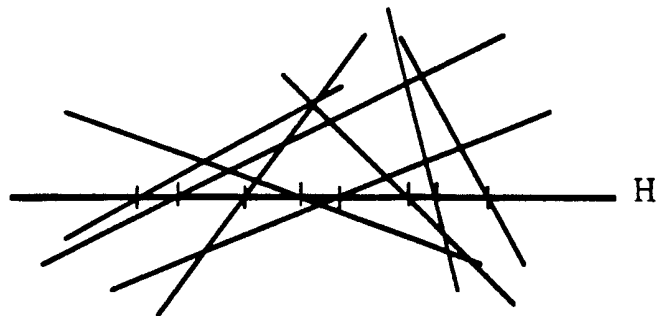


Figure 4.7 Congruent intervals on a line L

Procedure **MIPP_LINE** (H, P_1, P_2, \dots, P_n)

/* find a line that passes through a given line H and intersects the polygons P_1, P_2, \dots, P_n in the plane maximally.

*/

begin

step 1. For each pair of polygons P_i and P_j ($i \neq j$ and $i, j \leq n$), if they are not parallel and $CH(P_i)$ and $CH(P_j)$ are disjoint, compute their supporting lines.

step 2. Intersect line H with the supporting lines (obtained at step 1).

step 3. Sort the intersection points on H (as obtained from step 2), resulting in a finite number of congruent intervals in H .

step 4. From each congruent interval (obtained at step 3), pick an arbitrary point and call **MIPP_POINT** to solve its **MIPP_P**.

step 5. From the solutions obtained at step 4, choose one that intersects the most number of polygons; return it as the solution to **MIPP_L**.

end {**MIPP_LINE**}

Theorem 4.2. The **MIPP_L** can be solved in $O(VN^2+N^3\log N)$ time and $O(V+N^2)$ space, where N is the number of polygons in the plane, and V is the total number of their vertices.

Proof. Three preprocessing steps are needed prior to the computation of the supporting lines of two polygons P_i and P_j . They are: (a) check if P_i and P_j are parallel to each other, (b) construct the convex hull of P_i and P_j , and (c) detect the possible overlap between $CH(P_i)$ and $CH(P_j)$. Step (a) takes constant time. Steps (b) and (c) requires the time linear with the number of vertices of P_i and P_j . (Refer to [4] and the Appendix). Since for each convex polygon we need to compute at most the $4(N-1)$ external and internal supporting lines between it and the rest $N-1$ convex polygons which requires $O(V)$ time [4] and there are at most N such polygons, step 1 takes at most $O(V*N)$ time. Because there are at most $4N^2$ supporting lines, the intersection at step 2 requires only $O(N^2)$ time. These intersection points are then sorted at step 3 to obtain $O(N^2)$ congruent intervals, which requires $O(N^2\log N)$ time. At step 4, one execution of procedure **MIPP_POINT** is performed for each of the congruent intervals, resulting in a total time complexity of $O(N^2*(V+N\log N))$. Step 5 is certainly $O(N^2)$. Overall, the time requirement of the algorithm **MIPP_LINE** is $O(VN^2+N^3\log N)$. The proof of the space requirement is achieved by the fact that there are only $O(N^2)$ congruent intervals on L . Q.E.D.

Finally, by referring to the structural relation between **MIPS** and **MIPP** as shown in Figure 2.4 and the time and space linearity of the central projection mapping, we state the following theorem.

Theorem 4.3. Both **MIPP** and **MIPS** can be solved in $O(VN^2+N^3\log N)$ time and $O(V+N^2)$ space, where N is the total number of the polygons and V is the number of their vertices.

5. Conclusion Remarks

By introducing $O(N^2)$ homogeneous zones and finding the maximum intersection of N polygons via a point in each homogeneous zone in $O(N \log N)$ time, we solve the maximum intersection of polygons in the plane (**MIPP**). Its equivalent problem on the sphere **MIPS** is hence effectively solved.

We note that unbounded polygons give rise to additional processing. If all polygons are bounded, as enabled by rotating the visibility maps away from the “equator” [2], which may not always be possible, the overall time for **MIPS** improves from $O(VN^2 + N^3 \log N)$ to $O(VN^2 + N^2 \log N)$. We also note that by approximating the polygons with their largest inscribing circles, the time further improves to $O(N^2)$.

While the stated application is for 4-axis and 5-axis machining: orienting the workpiece such that the eigen circle intersects the visibility maps maximally so as to minimize the number of setups, the complement problem - Minimum Intersection of Polygons on Sphere - has interesting applications too. Consider die casting or mold injection. It may be desirable to avoid having the parting line (that separates the two halves of the die) on too many surfaces. This objective may be stated as the determination of minimum intersection of the parting line with spherical polygons (or small circles). Furthermore, the location of the parting line determines the number of cones (or inserts), the minimization of which is also desirable. Conceivably, the structure and the solution we achieved for **MIPS** may have direct contribution to solving this complemented problem. We will report these findings as soon as they are finalized.

Acknowledgement

This work is supported in part by a grant from the Science Research Laboratory, Ford Motor Company, Dearborn, Michigan. The authors have also benefits from the many helpful discussions they have had with their colleagues: L.L. Chen, S.Y. Chou, D. Dutta, J. Gan, D.S. Kim, W.J. Lee, J. Park, S.Y. Shin and J. Wolter, as well as encouragement from the research community.

Appendix.

In this appendix, we prove that the convex hull of an unbounded polygon can be computed in time linear in the number of vertices. Secondly, it is shown that the convex hull of two convex polygons, when at least one of them is unbounded, can be found in linear time as well. Thirdly, if both polygons are unbounded then they have at most one external supporting line. Finally, we prove that two disjoint non-parallel unbounded convex polygons always have two non-parallel internal supporting lines which can be obtained in linear time. In the discussion, a polygon P is represented as $P = (p_1, p_2, \dots, p_n)$ where p_1, p_2, \dots, p_n are the vertices of P in clockwise order. When P is unbounded, p_1 and p_n are at infinity and the two halfline edges of P will be denoted as $[p_2, p_1>$ and $[p_{n-1}, p_n>$.

Lemma A.1. The convex hull of an unbounded polygon $P = (p_1, p_2, \dots, p_n)$ can be computed in $O(n)$ time.

Proof. Let u be a point on the halfline $[p_2, p_1>$ that is outside the convex hull of the set of point p_2, \dots, p_{n-1} , and v be a similar point on the halfline $[p_{n-1}, p_n>$. Such two points can be easily obtained in $O(n)$ time. For example, u can be the intersection point between $[p_2, p_1>$ and a line orthogonal to $[p_2, p_1>$ such that all the vertices are at one side of this line (see Figure A.1). The list of vertices $P' = (u, p_2, p_3, \dots, p_{n-1}, v)$ becomes a closed simple polygon whose convex hull $CH(P')$ can be found in $O(n)$ time [4]. By the way u and v are defined, they are the two vertices on $CH(P')$. Let u' be the vertex of $CH(P')$ that is to the right of $[p_2, p_1>$ and farthest from $[p_2, p_1>$, and v' be the vertex that is to the left of $[p_{n-1}, p_n>$ and farthest from $[p_{n-1}, p_n>$. (See Figure A.2.) u' and v' partition the vertices of $CH(P)$ into two chains, one of which $[u', \dots, v']$ is concave to both p_1 and p_n , shown darkened in Figure A.2. By making a halfline R emanating from u' and parallel to $[p_2, p_1>$ and another halfline R' emanating from v' and parallel $[p_{n-1}, p_n>$, the resultant unbounded polygons as defined by these two halflines and that concave chain of $CH(P')$ forms the convex hull of P . (See Figure A.2.)

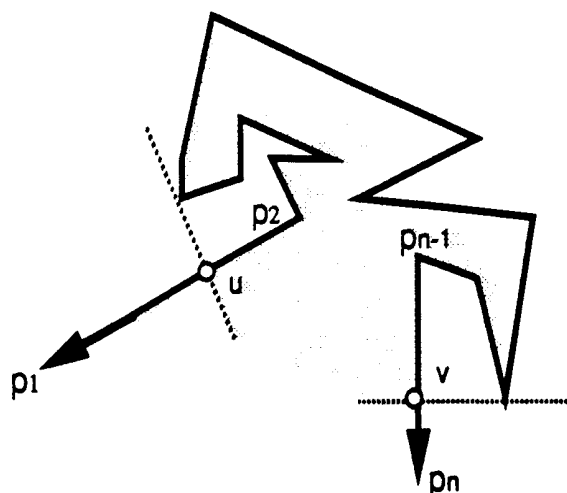


Figure A.1 Finding the extreme points u, v of an unbounded polygon

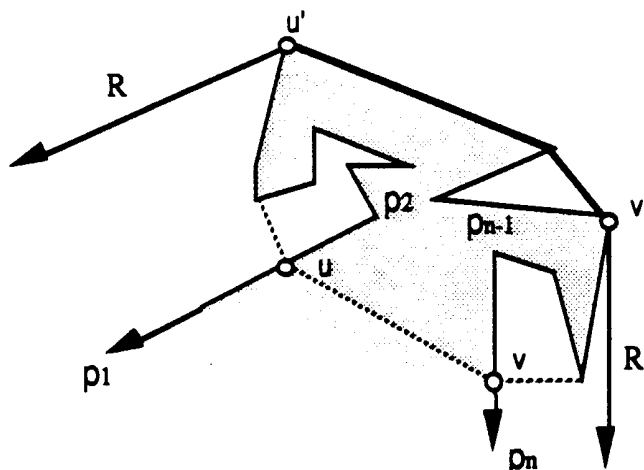


Figure A.2 Constructing the convex hull of an unbounded polygon

Lemma A.2. The detection of the overlap between two unbounded convex polygons $P=(p_1, p_2, \dots, p_m)$ and $Q=(q_1, q_2, \dots, q_n)$ can be done in $O(m+n)$ time.

Proof. If the halfline $[p_2, p_1>$ or $[p_{m-1}, p_m>$ intersects the edges of Q or the halfline $[q_2, q_1>$ or $[q_{n-1}, q_n>$ intersects the edges of P , then P and Q overlap. This can certainly be done in $O(2m+2n)$ time. Suppose they do not intersect. First we check for possible intersection between the two bounded convex polygons $P'=(p_2, p_3, \dots, p_{m-1})$ and $Q'=(q_2, q_3, \dots, q_{n-1})$ which can be carried out in $O(n+m)$ time [4]. If they overlap then P and Q overlap. If they do not overlap, either P and Q are disjoint or one of them is completely inside the other. These two cases can be distinguished by

checking whether an edge of P , say $[p_2, p_1]$ is inside Q or an edge of Q , say $[q_2, q_1]$ is inside P , which can be done in $O(m+n)$ time because of the convexity of P and Q .

Q.E.D.

Lemma A.3. The convex hull for a bounded convex polygon $P=(p_1, p_2, \dots, p_m)$ and an unbounded convex polygon $Q=(q_1, q_2, \dots, q_n)$ can be found in $O(m+n)$ time.

Proof. First we construct the convex hull of two bounded convex polygons P and $Q'=(q_2, q_3, \dots, q_{n-1})$ which takes $O(n+m)$ time [4]. (See Figure A.3(a).) Then the two extreme vertices u' and v' on $CH(P, Q')$ are found and the two halflines emanating from them are made, taking $O(n+m)$ time. (See Figure A.3(b).) Referring to the proof in Lemma A.1, these two halflines together with $CH(P, Q')$ uniquely determine $CH(P, Q)$. Q.E.D.

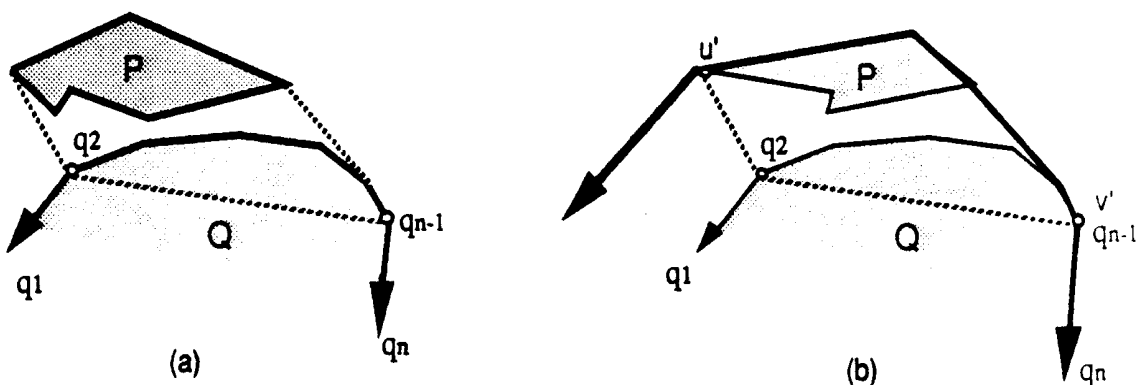


Figure A.3 Construction of the convex hull between a bounded and an unbounded polygons

Lemma A.4. Two disjoint unbounded convex polygons $P=(p_1, p_2, \dots, p_m)$ and $Q=(q_1, q_2, \dots, q_n)$ have at most one external supporting line that can be obtained in $O(n+m)$ time.

Proof. We first describe how the convex hull $CH(P, Q)$ is computed. Let $C = (c_1, c_2, \dots, c_k)$ be the convex hull between P and the bounded convex polygon $Q'=(q_2, q_3, \dots, q_{n-1})$, which is $O(n+m)$ obtainable by Lemma A.3. Consider the halflines $[q_2, q_1]$ and $[q_{n-1}, q_n]$ of Q . If both of them are inside C , C is obviously

equal to $CH(P,Q)$. Suppose $[q_2, q_1>$ intersect the boundary of C . If C has a supporting line that is parallel to $[q_2, q_1>$, say at vertex c_i , then $CH(P,Q)$ must cover the halfline emanating at c_i and with the same direction of $[q_2, q_1>$. Therefore, the convex polygon C' as defined by the vertices (c_1, c_2, \dots, c_i) and that halfline is $CH(C, [q_2, q_1>$). (See Figure A.4(a).) In the case that such a supporting line does not exist, i.e., when the halfline $[q_2, q_1>$ intersects both the opposite halflines R and R' of $[c_2, c_1>$ and $[c_{k-1}, c_k>$, $CH(P,Q)$ is the entire plane (see Figure A.4(b)). We can now perform the similar operation on the other halfline $[q_{n-1}, q_n>$ on the updated convex hull C' , and the resultant unbounded convex polygon, if it exists, is $CH(P,Q)$.

To prove that P and Q have at most one external supporting line, suppose $CH(P,Q)$ exists and its vertices are (c_1, c_2, \dots, c_k) . Since P and Q are disjoint, there exists a line L that separates P from Q [4]. L intersects the boundary of $CH(P,Q)$ at only one point, for otherwise it would mean one of P and Q is bounded. (See Figure A.4(c).) Therefore, only the line collinear with the edge $[c_i, c_{i+1}]$ which L intersects is the external supporting line between P and Q since c_i and c_{i+1} belong to different polygons. (See Figure A.4(d).) Q.E.D.

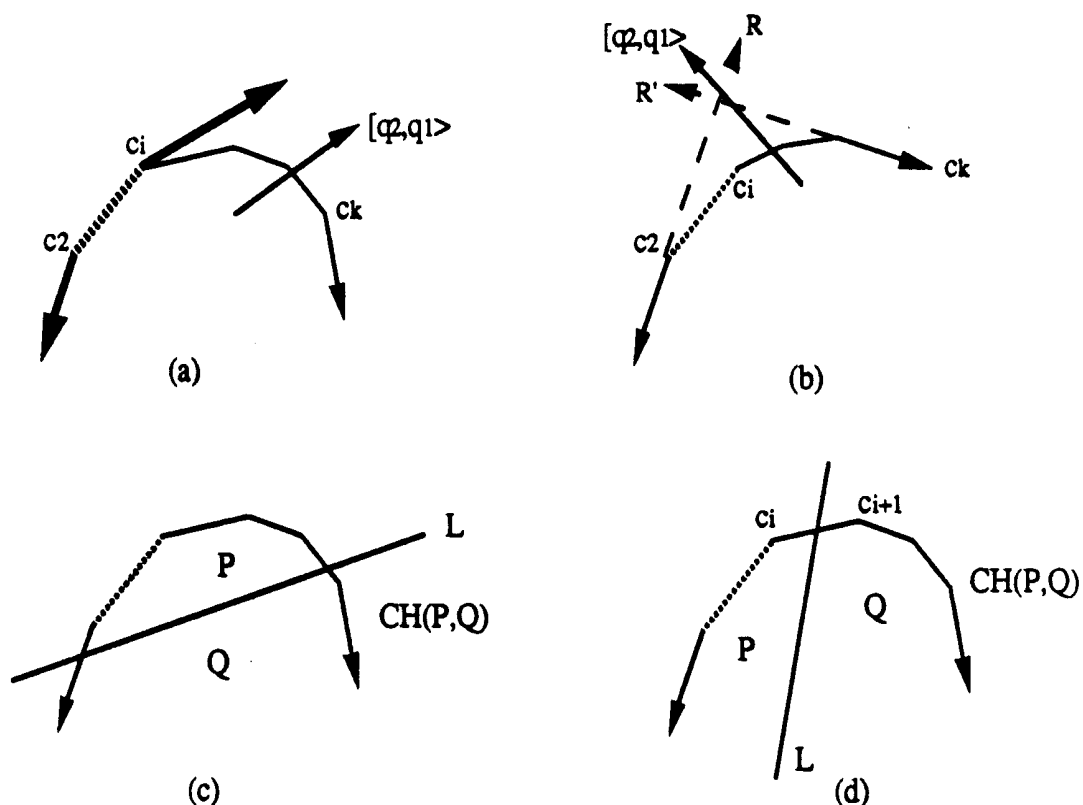


Figure A.4 Proof of Lemma A.4

Lemma A.5. Two disjoint non-parallel unbounded convex polygons $P=(p_1, p_2, \dots, p_m)$ and $Q=(q_1, q_2, \dots, q_n)$ have two non-parallel internal supporting lines which can be computed in $O(n+m)$ time.

Proof. Let L be a line separating P from Q which can be obtained in $O(m+n)$ time [4]. Without loss of generality, assume that L is horizontal and touches a vertex p_i of P which sits above L . Consider rotating L counter-clockwise about p_i before penetrating either P or Q . Only four cases are possible for L to penetrate P or Q :

- (a) L touches vertex p_{i+1} ,
- (b) i is $m-1$ and L becomes parallel to $[p_{m-1}, p_m]$,
- (c) L touches a vertex q_j , or
- (d) L becomes parallel to $[q_{n-1}, p_n]$.

For case (a), we continue to rotate L counter-clockwise, but about vertex p_{i+1} . For case (b), we move L parallel toward Q until it touches Q . (See Figure A.5(a).) For case (c), L continues to rotate counter-clockwise, but about the vertex q_j of Q . The treatment for case (d) is similar to case (b) except this time L moves parallel toward P until it touches P . (See Figure A.5(b).) This rotation of L stops either when case (b) or case (d) happens or when case (a) and case (c) occur simultaneously. (See Figure A.5(c).) Clearly the resultant internal supporting line L_u through such a series of rotations can be obtained in $O(n+m)$ time since each vertex of P and Q could be the hinge point at most once.

By rotating the horizontal line L clockwise, another internal supporting line L_c is found. We now claim L_u and L_c are not parallel. The only possibility for L_u and L_c to be parallel is when they are both horizontal. Suppose L_u is horizontal. Figure A.6 shows all three possible positions of L_u relative to P and Q . The horizontal line L_u can be rotated clockwise by a non-zero angle $\text{Min}\{\theta_1, \theta_2\}$ before penetrating either P or Q . That is, L_c can never be horizontal. Q.E.D.

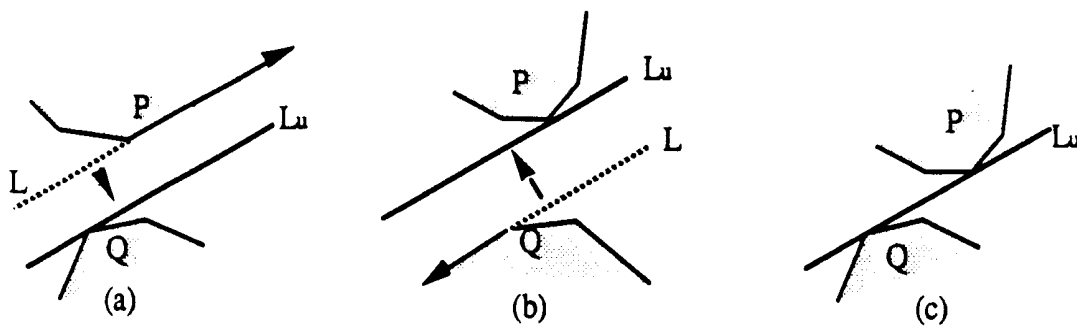


Figure A.5 Determination of an internal supporting line

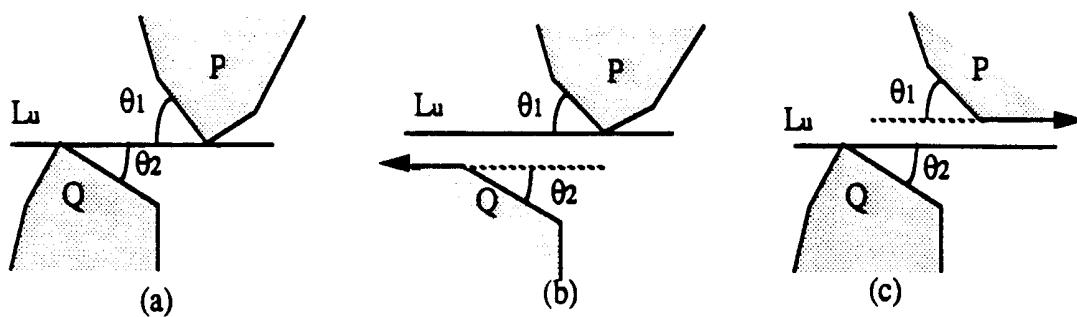


Figure A.6 Proof of the non-parallelism of internal supporting lines

Reference

- [1] K. Q. Brown, *Geometric Transformation for Fast Geometric Algorithms*, Ph.D. thesis, Dept. of Computer Science, Carnegie Mellon Univ., Dec. 1979
- [2] Lin-Lin Chen and T.C. Woo, "*Computational Geometry for Automated Machining*", submitted for publication, currently available as IOE Tech Report 89-30, Dept. of IOE, the University of Michigan, Ann Arbor, MI 48109.
- [3] D. Hilbert and S. C. Vossen, *Geometry and Imagination*, New York: Chelsea, 1952.
- [4] F.P. Preparata and M.L. Shamos, *Computational Geometry*, Springer-Verlag, 1985.
- [5] H. Rohnert, "Shortest Paths in the Plane with Convex Polygonal Obstacles", *Information Processing Letters* 23, pp71-76, 1986
- [6] K. Tang, "On the Intersection of a Set of Direction Cones", *Computer Vision, Graphics and Image Processing*, Vol. 45, pp. 357-361, March 1989