A Linear Time Algorithm for Constructing Visibility Diagram

Shou-Yan Chou
Tony C. Woo
Department of Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 91-28

October 1991

To appear in ASME Transactions, J. of Mechanical Design

A Linear Time Algorithm for Constructing a Circular Visibility Diagram

Shuo-Yan Chou
Tony C. Woo
Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 91-28

October 1991

A Linear Time Algorithm for Constructing a Circular Visibility Diagram

Shuo-Yan Chou
Tony C. Woo
Department of Industrial and Operations Engineering
University of Michigan

October 28, 1991

Abstract

Unlike linear visibility for which there exists only one line segment connecting two points, there are an infinite number of circular arcs connecting two points. To compute circular visibility: a collection of circular arcs which emanate from a given point and reach an edge without intersecting another edge in a polygon, the arcs are classified with respect to the edges of the polygon resulting in a planar partition. It is shown in this paper that such a partition can be constructed in $O(n)$ time, where n is the number of vertices in the polygon. Given such a diagram, the point visibility hull inside a simple polygon can be found easily.

1 Introduction

Among the visibility problems, a fundamental one is the computation of a point visibility polygon: the portion of a polygon which is visible to a point internal to it. ElGindy and Avis [9] and Lee [12] developed linear time algorithms for constructing a point visibility polygon inside a simple polygon. Another fundamental visibility problem is the computation of an edge-visibility polygon. Introduced by Avis and Toussaint [2], edge visibility is divided into three categories: complete, strong, and weak. Whether a polygon is completely or strongly visible to a given edge can be answered by the *kernel algorithm* developed by Lee and Preparata [11], whereas the problem of detecting if a polygon is weakly visible from an edge can be solved in linear time [2]. Chazelle et. al [4,5] show that an edge-visibility polygon inside a triangulated simple polygon can be constructed in linear time. Suri and O'Rourke [17] show that an edge visibility polygon inside a non-simple polygon can be constructed in $\Omega(n^4)$ time. These linear visibility algorithms support many applications. The *art gallery* problem [14] seeks the minimum number of points inside a polygon such that the point-visibility polygons of these points cover the entire polygon. The *minimum link path* between two points inside a simple polygon is solved optimally by constructing a sequence of visibility polygons [16]. The *shortest path* can also be solved in linear time by utilizing visibility [10].

Circular visibility is established by arcs. Since straight lines can be considered as degenerate arcs, the realm of visibility is extended by considering circularity, the notion of of which is illustrated in Figure 1, where a point q is *circularly visible* to a point p if a circular arc can be drawn from p to q without hitting an obstacle. Such a directed circular arc - clockwise or counterclockwise - is called a *visibility arc* and can be uniquely defined by its radius, center, endpoints, and direction. For

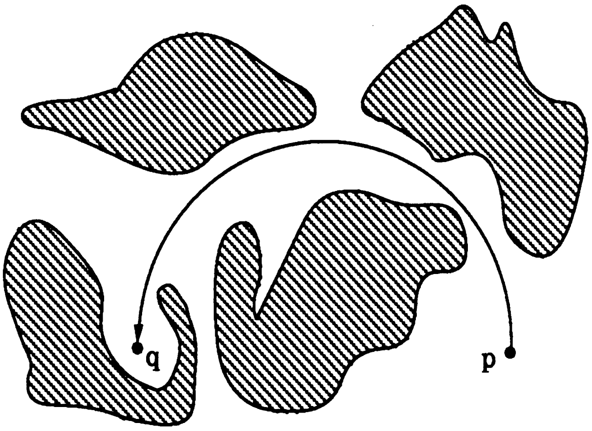


Figure 1: Circularly visible points

the simplicity of discussion, we adopt the counterclockwise direction. An $O(n \log n)$ algorithm has been reported by Agarwal and Sharir [1] for computing the portion of a simple polygon which is circularly visible to a fixed interior point.

Circular visibility can be used to characterize the motions of a mobile robot. The number of links in a minimum link path is smaller with circular arcs than with line segments as links. Circular visibility can also be used to characterize the workspace of a stationary robot with rotary joints - locating joints for assembly or disassembly.

Clearly, there can be more than one circular arc between the point in question and points on an edge of a polygon. Since an arc can be represented by its center, the visibility arcs to a particular edge of the polygon can be represented by a region for their corresponding centers. We utilize this representation to solve the following problem.

Problem CVD(p,Q): Circular Visibility Diagram of a Simple Polygon Q

Given: a simple polygon Q with edges e_0, e_1, ..., e_n, and a point p contained in Q.

Find: for each e_i, the circular arcs which emanate from p and intersect e_i before intersecting any other edge of Q.

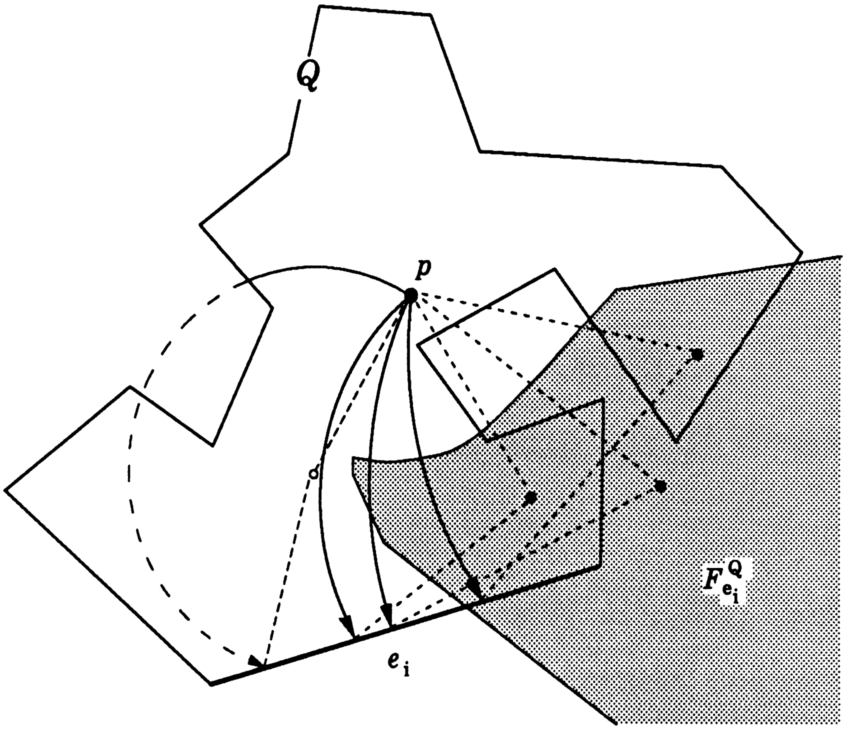


Figure 2: Some counterclockwise visibility arcs of edge e_i.

Let $F^Q_{e_i}$ denote the set of centers corresponding to the counterclockwise visibility arcs from p to e_i, as shown in Figure 2. (The notation F^B_A is used, throughout this paper, to represent the set of centers of the visibility arcs from a given point to A in the presence of B.) Since visibility arcs are represented by their centers, a solution to Problem CVD(p,Q) indicates a partition of the plane. The partition is represented as $\{F^Q_\phi, F^Q_{e_0}, F^Q_{e_1}, ..., F^Q_{e_n}\}$, where F^Q_ϕ represents the set of centers for counterclockwise arcs which emanate from p that do not hit the boundary of Q. Such a partition is called the *circular visibility diagram* (CVD) of Q with respect to p.

The data structure of the CVD is similar to the dual space data structure used by Chazelle and Guibas [5] for solving a variety of linear visibility problems. In that paper, a transform is employed in which a line $ax + by + 1 = 0$ is represented by a point (a,b) [4,13]. Points in the dual space are grouped into regions according to

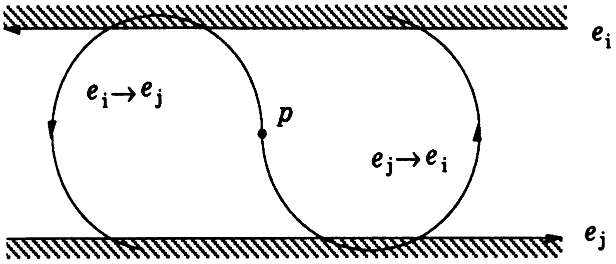


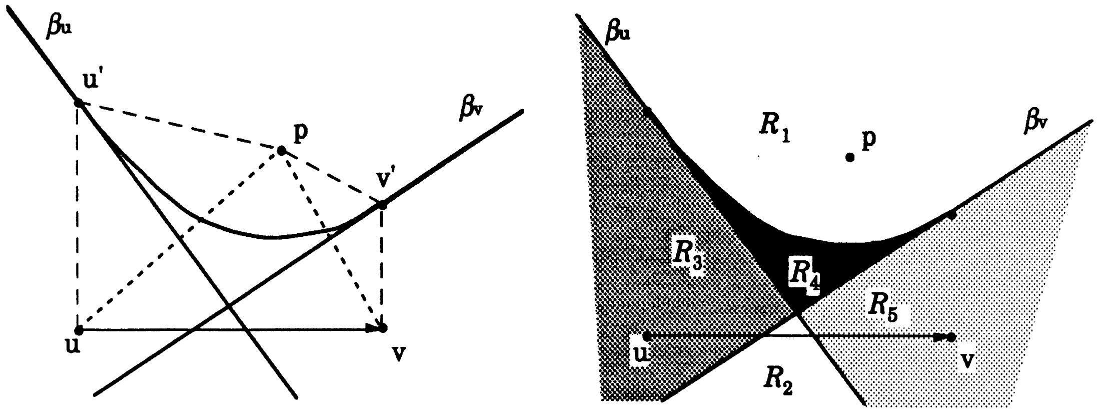
Figure 3: The possible hitting order between two edges.

the edge whose corresponding visibility rays hit, resulting in a planar partition in the dual space.

In linear visibility, a partial order in which visibility rays hit the edges of a polygon is crucial for the construction. In circular visibility, however, arcs emanating from a point can hit two edges in either order, as shown in Figure 3. To overcome the apparent lack of a partial order, a polygon is decomposed into a star-shaped polygon and a set of pockets, each of which exhibit a partial order. By constructing the CVDs for the star-shaped polygon and then for every pocket, the CVD of a simple polygon is constructed. The sections follow the development of such an idea. The CVD of a single edge is first examined. Based on the CVD for an edge, algorithms for constructing the CVDs of two types of open polygonal curves – obtuse star-shaped chains and pockets – are developed. Finally, the circular visibility diagram of the simple polygon is constructed and linearity in time for the construction is shown.

2 Circular Visibility Diagram of An Edge

In this section, the CVD for a single edge is constructed. The objective is to classify the visibility arcs into three groups – those which hit one side of an edge; those which hit the other side of the edge; and those which do not hit an edge at all – by their



(a)

(b)

Figure 4: A partition of counterclockwise visibility arcs with respect to edge $\vec{uv}$.

corresponding centers. The distinction among various visibility arcs is first examined.

Let p be the point from which visibility arcs emanate. Let q be a point on a directed edge $\vec{uv}$. A visibility arc can reach q from the left or from the right of $\vec{uv}$. Such a distinction among visibility arcs can be made by classifying the loci of the centers to p and to $\vec{uv}$, as the partition shown in Figure 4 (a). First, the bisector β_u of p and u and the bisector β_v of p and v, are constructed respectively. Then, a parabola $\beta_{\overleftrightarrow{uv}}$ with focus p and directrix coincident with $\vec{uv}$, is constructed. By construction, $\beta_{\overleftrightarrow{uv}}$ is tangent to β_u and β_v at u' and v', and both $\bar{uu'}$ and $\bar{vv'}$ are perpendicular to $\bar{uv}$ [3]. Let $\beta'_{\overleftrightarrow{uv}}$ denote the portion of $\beta_{\overleftrightarrow{uv}}$ between u' and v'. Let β_u^+ denote the half-line of β_u which has C^1 continuity with $\beta'_{\overleftrightarrow{uv}}$ at u' and β_v^+ the half-line of β_v which has C^1 continuity with $\beta'_{\overleftrightarrow{uv}}$ at v'[1]. While β_u, β_v, and $\beta_{\overleftrightarrow{uv}}$ contain all the equidistant points between p and u, p and v, and p and $\overleftrightarrow{uv}$, respectively, the continuous curve $\beta_{\bar{uv}}$ consisting of β_u^+, β_v^+, and $\beta'_{\overleftrightarrow{uv}}$ contains all the equidistant points between p and $\bar{uv}$. This curve is also known [15] as the *Voronoi diagram* of p and $\bar{uv}$.

β_u, β_v, and $\beta'_{\overleftrightarrow{uv}}$ partition the plane into five regions, R_1, R_2, ..., and R_5, as shown

[1] The other halves of β_u and β_v are denoted as β_u^- and β_v^-.

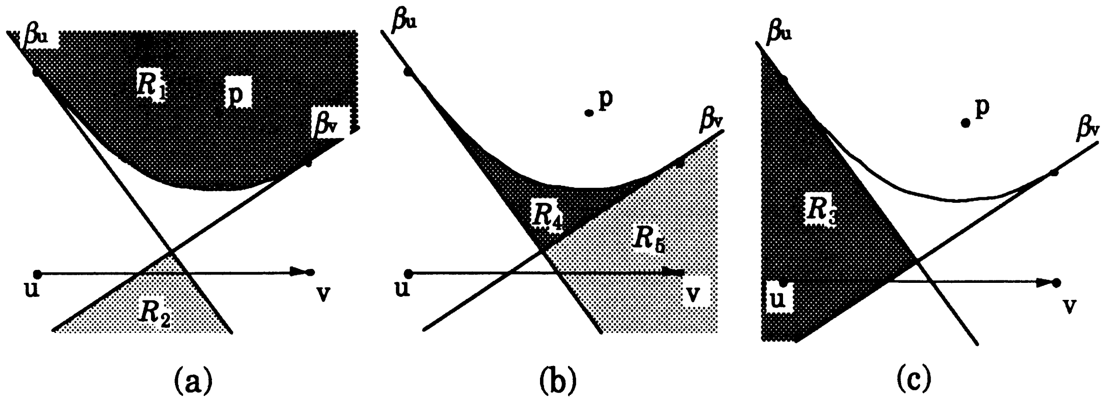


Figure 5: The centers of visibility arcs which: (a) miss $\vec{uv}$, (b) hit $\vec{uv}$ from the left, and (c) hit $\vec{uv}$ from the right.

in Figure 4 (b). It is shown in the following theorem that, arcs drawn from p about points in a region will miss $\vec{uv}$, or hit $\vec{uv}$ from the left, or hit $\vec{uv}$ from the right. These five regions are thus combined into three sets, as shown in Figure 5. We note that hitting the right side of $\vec{uv}$ is essentially the same as hitting the left side of $\vec{vu}$.

Theorem 2.1 *Let $F_{\phi}^{\vec{uv}}$, $F_{\vec{uv}}$, and $F_{\vec{vu}}$ represent, respectively, the regions containing all the points about which counterclockwise arcs drawn from p miss $\vec{uv}$, hit $\vec{uv}$ from the left, and hit $\vec{uv}$ from the right. Then,*

(i) $F_{\phi}^{\vec{uv}} = R_1 \cup R_2$

(ii) $F_{\vec{uv}} = R_4 \cup R_5$

(iii) $F_{\vec{vu}} = R_3$

Proof: As R_1 lies on the side of the Voronoi diagram which contains p, the distances $d(x,p) < d(x,\vec{uv})$, for all $x \in R_1$, which means that visibility arcs from p centered at a point in R_1 will not intersect $\vec{uv}$. Similarly, since R_2 is the intersection of the half-planes (not containing p) of the bisectors β_u and β_v, i.e. $d(x,u) < d(x,p)$ and

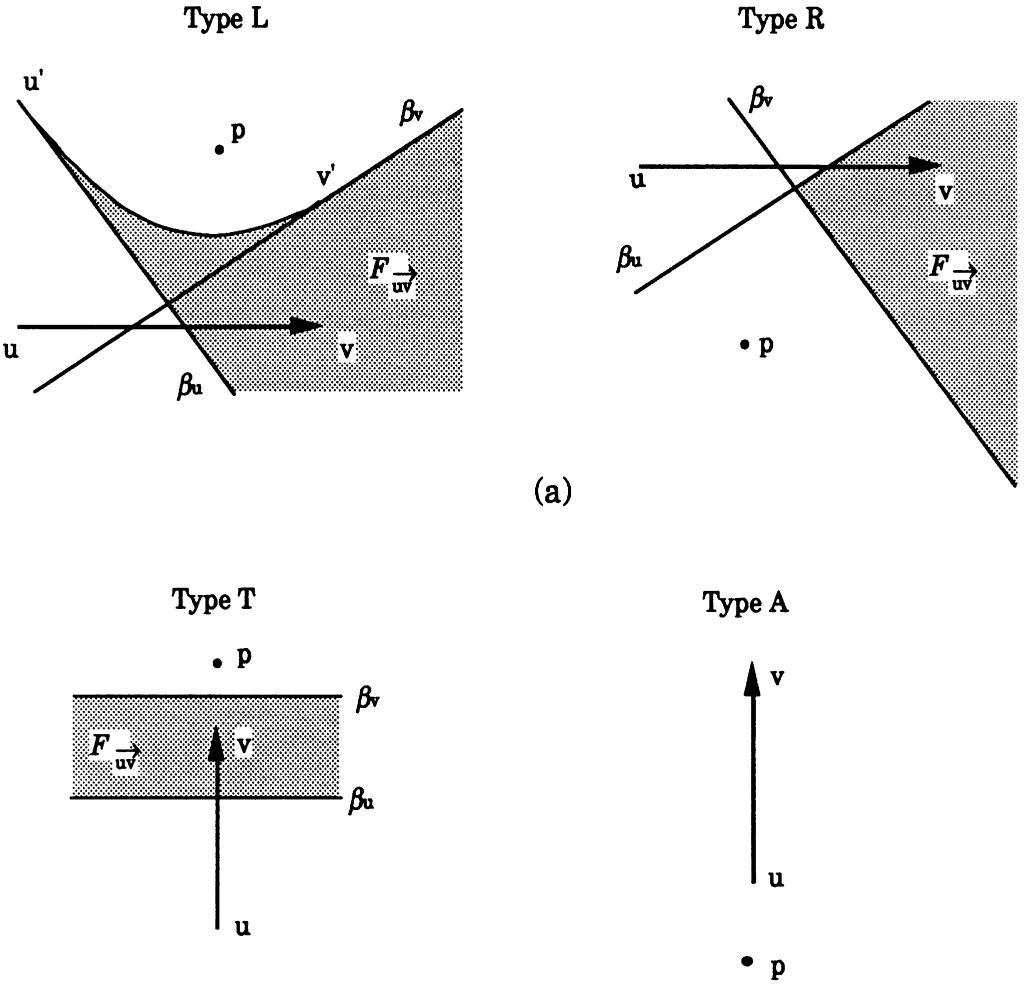
$d(x, v) < d(x, p)$, for all $x \in R_2$, visibility arcs centered at a point in R_2 will not intersect $u\vec{v}$ either. Thus, (i) is true.

To show (ii) and (iii), first consider the points in R_4. As R_4 lies on the side of the Voronoi diagram that does not contain p, $d(x, u\vec{v}) < d(x, p)$, for all $x \in R_4$; R_4 also lies in the half-planes, containing p, of the bisectors β_u and β_v, in which $d(x, p) < d(x, u)$ and $d(x, p) < d(x, v)$, for all $x \in R_4$. In other words, any circle centered at a point in R_4 will not contain the end points of $u\vec{v}$; yet, the distance between the center and $u\vec{v}$ is shorter than the radius of the circle, which means that such circles will intersect $u\vec{v}$ at two points. Since R_4 lies to the left of $u\vec{v}$, all such arcs will intersect $u\vec{v}$ from the left.

A similar reasoning shows that every arc drawn from point p about a point in R_3 or R_5 will intersect $\bar{u}v$ at one point. Suppose an arbitrary arc drawn from p intersects $u\vec{v}$ at q. The center of the arc $\widehat{pq}$, if in R_5, always lies to the right of $q\vec{p}$. This indicates that a counterclockwise $\widehat{pq}$ drawn from p and centered at a point in R_5 always hits $u\vec{v}$ from the left. On the other hand, if arc $\widehat{pq}$ is drawn from p and centered at a point in R_3, it always hits $u\vec{v}$ from the right. This completes the proof for (ii) and (iii).

□

Suppose that the edges of a polygon are given in the counterclockwise order. Since p is inside the polygon, all the first crossings of the visibility arcs will be from the left of the edges. Thus only the intersections from the left of an edge are of interest. Figure 6 gives the four possible cases of the counterclockwise CVDs of an edge $u\vec{v}$. Figure 6 (a) depicts a Type L CVD, where p is to the *left* of $u\vec{v}$ and a Type R CVD, where p is to the *right* of $u\vec{v}$. In both cases, counterclockwise arcs drawn from p about a point in $F_{u\vec{v}}$ hit the left side of $u\vec{v}$. Figure 6 (b) shows the limiting cases Type $\grave{\mathrm{T}}$ and Type A, in which p is collinear with $u\vec{v}$ and $u\vec{v}$ is directed either *toward*



(a)

(b)

Figure 6: Counterclockwise CVDs for p and $\vec{uv}$.

or *away* from p. We note that if $\vec{uv}$ is directed away from p, no counterclockwise arc emanating from p will hit $\vec{uv}$ from the left side.

3 Circular Visibility Diagram of an Obtuse Star-shaped Chain

The classification of the visibility arcs which hit a special class of open polygonal chain on the left in next examined. As noted earlier, edges of a polygonal chain may obstruct visibility arcs. The resolution of the obstruction and subsequently the construction of the CVD can be costly. However, if a partial order in which an arc hits the edges of such a chain can be established, the CVD can be then constructed efficiently.

An open polygonal chain is *star-shaped* if the polar angle θ, which is the angle rotated counterclockwise from the polar axis, of a point traversing the chain is either monotonically decreasing or increasing, with respect to some point p. A star-shaped chain is *obtuse* if the span of θ of the points on the chain is less than 180°. Given an obtuse star-shaped chain with monotonically increasing θ with respect to p (with all the edges directed counterclockwise), the visibility arcs to the individual edges of the chain can be identified. To determine efficiently the first edge that an arc hits, a partial order in which circular arcs emanating from p hit the edges of the chain is identified. That such a partial order exists is established by the following lemma.

Lemma 3.1 *Let $C = \{e_0, e_1, e_2, ..., e_m\}$ be an obtuse star-shaped chain which is monotonically increasing in θ. A counterclockwise arc emanating from p hits e_i before e_j only if $i < j$. Similarly, if θ is monotonically decreasing, then an arc can hits e_i before hitting e_j only if $j < i$.*

Proof: For a chain that is monotonically increasing, a counterclockwise arc drawn from p which hits e_i at a point q will always lie to the left of $\vec{qp}$, whereas e_j, for all $j > i$, always lies to the right of $\vec{qp}$ because the chain C is star-shaped about p and spans less than 180°. Since such an arc does not intersect any succeeding edges of e_i in C, it hits another edge before hitting e_i only if the edge precedes e_i.

□

Since constructing the CVD for an obtuse star-shaped chain with decreasing θ is closely analogous to that with increasing θ, we will illustrate the latter case. Based on Lemma 3.1, a recursive relationship between the CVD of an obtuse star-shaped chain and the CVDs of its constituent edges is established.

Theorem 3.2 *The regions in a counterclockwise CVD of the monotonically increasing chain C can be computed by:*

$$F_{e_i}^{C} = \begin{cases} F_{e_0} & \textit{if } i = 0 \\ F_{e_i} \cap F_{\phi}^{C_{i-1}} & \textit{if } i > 0, \end{cases}$$

and

$$F_{\phi}^{C_i} = \begin{cases} F_{\phi}^{e_0} & \textit{if } i = 0 \\ F_{\phi}^{e_i} \cap F_{\phi}^{C_{i-1}} & \textit{if } i > 0 \end{cases}$$

where $C_i = \{e_0, e_1, ..., e_i\}$ *denotes a sub-chain of* C.

Proof: Based on Lemma 3.1, circular arcs about points in F_{e_i} may hit e_j, only if $j < i$. Therefore, for points to be in $F_{e_i}^{C_i}$, their corresponding circular arcs can not hit e_j, for all $j < i$. As $F_{\phi}^{C_i}$ represents the intersection of $F_{\phi}^{e_j}$, for all $j < i$, $F_{e_i}^{C_i}$ is equal to the intersection of F_{e_i} and $F_{\phi}^{C_{i-1}}$.

□

Intersecting F_{e_i} and $F_{\phi}^{C_{i-1}}$, for all $i > 0$, takes $O(n^2)$ time, which seems to indicate that an algorithm for constructing a CVD for an obtuse star-shaped chain exceeds linear time. However, by utilizing certain properties of the consecutive $F_{e_i}^{C}$s, we can show that constructing a circular visibility diagram of an obtuse star-shaped chain is analogous to cutting a pie, piece by piece. Also, since the time complexity for the identification of the cutting points is amortized, such a pie cutting procedure can therefore be achieved in linear time. A similar linear time cutting procedure is described by Edelsbrunner and Guibas [7] for computing a "bay" formed by lines sorted in slope order.

Before investigating the properties of the consecutive $F_{e_i}^{C}s$, the portion of $F_{\phi}^{e_i}$, which has no effect on the construction of the CVD, is identified and is omitted from the subsequent analysis. Recall that $F_{\phi}^{e_i}$, which equals the union of regions R_1 and R_2 as shown in Figure 5 (a), contains all the points about which arcs emanating from p miss e_i. Let region R_2 computed with respect to e_i be denoted as $R_2^{e_i}$. As to be established in Lemma 3.3, the intersection of $R_2^{e_i}$ and $F_{e_{i+1}}$ is always empty. Also, since the star-shaped chain is obtuse, the region F_{e_j}, for $j > i$, will not go around p and come back to intersect $R_2^{e_i}$. Therefore, each individual $F_{\phi}^{e_i}$ denotes only points in R_1, which is also the region of the Voronoi diagram of p and e_i containing p.

Lemma 3.3 *Let C be an obtuse star-shaped chain, and e_i and e_{i+1} be two consecutive edges in C. Then, $F_{e_{i+1}} \cap R_2^{e_i} = \phi$.*

Proof: Let β_{v_i} represent the perpendicular bisectors of $\bar{pv}$, as shown in Figure 7. Since C is star-shaped, p is to the left of e_i and e_{i+1}, which means that F_{e_i} and $F_{e_{i+1}}$ are both Type L regions. β_{v_i}, which is the perpendicular bisector of p and the vertex v_i joining edges e_i and e_{i+1}, creates two half-planes, one of which contains p. By definition, $F_{e_{i+1}}$ is on the side of half-plane containing p while R_2 of e_i is on the other

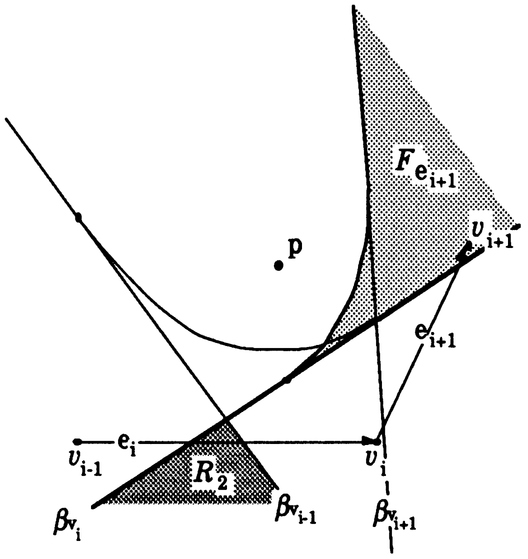


Figure 7: The superimposition of $F_{e_{i+1}}$ and R_{2,e_i}.

half-plane. Therefore, $F_{e_{i+1}} \cap R_2^{e_i} = \phi$.

□

Also, properties on the unbounded regions in the CVD are needed before establishing the properties on the consecutive $F_{e_i}^C$s. In a given CVD, it is possible to distinguish bounded regions from the unbounded ones. The existence of unbounded regions in a CVD is closely related to the linear visibility of the edges, as established in the following lemma.

Lemma 3.4 *For some edge e_i of a polygon Q, the region $F_{e_i}^Q$ is unbounded if and only if some point on e_i is linearly visible from p.*

Proof: A point q on e_i is linearly visible from p if and only if the line segment $\bar{pq}$ does not intersect any other edges of Q. A line segment is in fact a degenerate circular arc whose center lies on the perpendicular bisector of this line segment at infinity. A region corresponding to a linearly visible edge contains points to infinity and therefore is unbounded. The validity of the converse is easy to see.

□

It can also be shown that these unbounded regions of a CVD are ordered about p.

Lemma 3.5 *The counterclockwise order of the unbounded regions around p in the counterclockwise CVD is the same as that of the edges linearly visible from p.*

Proof: The order of the linearly visible edges around p follows the order of the rays from p to the edges. Also, the order of the unbounded regions around p follows the order of the centers at infinity which in turn follows the order of the rays from p to the edge because the radii of the centers at infinity are perpendicular to the right of the rays from p.

□

With the order on the unbounded regions of the CVD in hand, a property between the consecutive $F^C_{e_i}$s can now be extablished. Without loss of generality, all the CVDs of individual edges of a chain C are assumed to be of Type L. (Refer to Figure 6.) Being Type L, F_{e_i} of edge e_i in C is bounded by three curves: $\beta^-_{v_{i-1}}$, β'_{e_i} which is the portion of β_{e_i} between v'_{i-1} and v'_i, and $\beta^+_{v_i}$, as shown in Figure 8 (a). To simplify the following discussion, β'_{e_i} and $\beta^+_{v_i}$ are denoted by a single curve f_{e_i}, as indicated in Figure 8 (a). The portion of f_{e_i} remaining in the CVD of an obtuse star-shaped chain C is denoted as $f^C_{e_i}$, as shown in Figure 8 (b).

Since C is star-shaped about p, i.e. every vertex on e_i is linearly visible to p, $F^C_{e_i}$ is unbounded and the order of $F^C_{e_i}$ about p is the same as that of e_i. Such an order of $F^C_{e_i}$ indicates an order for constructing the CVD of C. In the following, $F^C_{e_{i+1}}$ is shown to be bounded by $f_{e_{i+1}}$ and the boundaries of $F^{C_i}_{\phi}$.

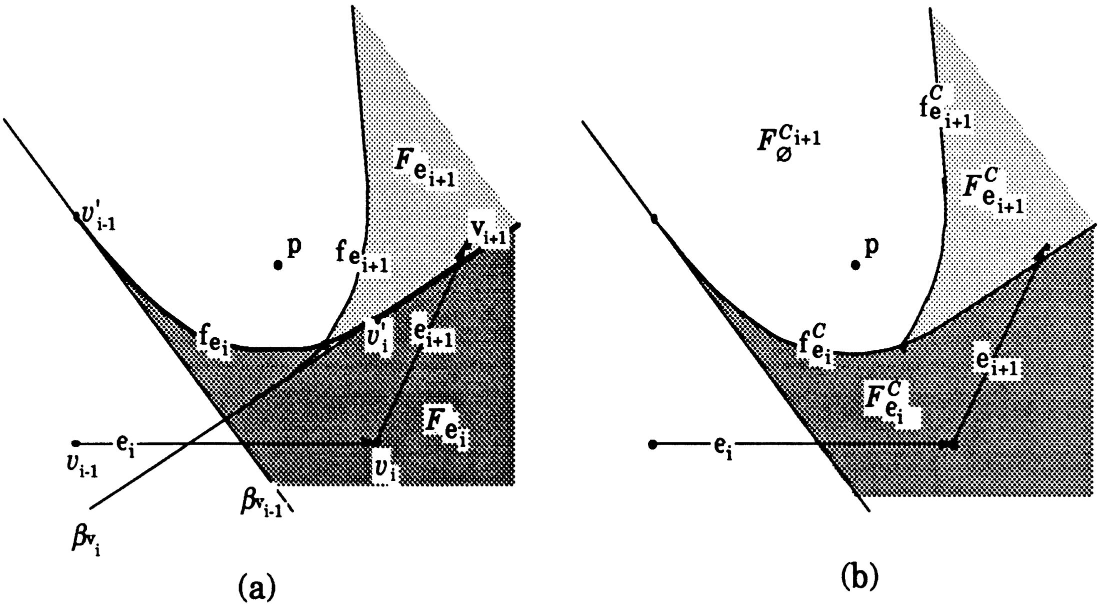


Figure 8: (a) Overlapping the CVDs of two edges. (b) The circular visibility diagram of two consecutive edges.

Lemma 3.6 *Let C be an obtuse star-shaped chain, and let e_i and e_{i+1} be two consecutive edges in C. Then, $F^C_{e_{i+1}}$ is bounded by $f_{e_{i+1}}$ and the boundaries of $F^{C_i}_{\phi}$.*

Proof: As shown in Theorem 3.2, $F^C_{e_{i+1}}$ is equal to the intersection of $F_{e_{i+1}}$ and $F^{C_i}_{\phi}$. As $F_{e_{i+1}}$ is a Type L region, by definition, it is bounded by $f_{e_{i+1}}$ and β_{v_i}. See Figure 8 (a). Also, $F^{C_i}_{\phi}$ is bounded by β_{v_i} as $F^{e_i}_{\phi}$ is bounded by β_{v_i} and $F^{C_i}_{\phi}$ is a subset of $F^{e_i}_{\phi}$. Moreover, since both $F_{e_{i+1}}$ and $F^{C_i}_{\phi}$ lie in the half-plane of β_{v_i} which contains p, $F^C_{e_{i+1}}$ is therefore bounded by $f_{e_{i+1}}$ and the boundary of $F^{C_i}_{\phi}$.

□

It is also essential for the algorithm that $f_{e_{i+1}}$ intersects the boundaries of $F^{C_i}_{\phi}$ at only one point, which is equivalent to showing that $f^C_{e_i}$ is continuous.

Lemma 3.7 *$f^C_{e_i}$ is continuous, for all i.*

Proof: Given two consecutive edges, we show that $f_{e_{i+1}}$ will intersect f_{e_i} only once. This is because that the tangents of the points on f_{e_i} are non-decreasing as f_{e_i} goes to infinity and are bounded by the two perpendicular bisectors between p and the two end points of e_i. Thus, the tangents of the points on $f_{e_{i+1}}$ are greater than those on f_{e_i}. Also, since C is obtuse, $\beta_{v_{i+1}}$ can not go around p and hit f_{e_i}. Therefore, $f_{e_{i+1}}$ will only intersect f_{e_i} once. By using the same argument, we can show that $f_{e_{i+1}}$ can intersect the boundaries of $F_{\phi}^{C_i}$ only once, which means that the portion of $f_{e_{i+1}}$ becoming $f_{e_i}^{C}$ is continuous.

□

Lemma 3.6 indicates that $F_{e_{i+1}}^{C}$ can be constructed by intersecting $f_{e_{i+1}}$ with the boundaries of $F_{\phi}^{C_i}$, and Lemma 3.7 that they intersect at only one point. Subsequently, in each iteration of the algorithm, $f_{e_{i+1}}^{C}$ partitions $F_{\phi}^{C_i}$ into two regions: $F_{e_{i+1}}^{C}$ and $F_{\phi}^{C_{i+1}}$, as depicted in Figure 8 (b).

In the algorithm for constructing a CVD, F_{e0}^{C} and $F_{\phi}^{C_0}$ are constructed first. This can be done in constant time. Then, in each iteration, f_{e_i} is employed to intersect the boundaries of $F_{\phi}^{C_{i-1}}$, which subsequently results in $F_{\phi}^{C_i}$ and $F_{e_i}^{C}$. After $f_{e_i}^{C}$ is computed, it replaces those boundaries[2] of $F_{\phi}^{C_{i-1}}$ which lie in $F_{e_i}^{C}$ and becomes a boundary of $F_{\phi}^{C_i}$. A stack is used to record $f_{e_j}^{C}$ that comprise the boundaries of $F_{\phi}^{C_i}$; every time an f_{e_i} is introduced to partition $F_{\phi}^{C_{i-1}}$, those $f_{e_j}^{C}$ that do not intersect f_{e_i} are popped out until one does. Then $f_{e_i}^{C}$ is computed and pushed to the top of the stack. Those $f_{e_j}^{C}$ that were popped, along with $f_{e_i}^{C}$, comprise the boundaries of $F_{e_i}^{C}$. At the end of the algorithm, the $f_{e_j}^{C}$s remaining in the stack are identified as the boundaries of F_{ϕ}^{C}.

[2] It is noted that the particular $f_{e_j}^{C}$ bounding $F_{\phi}^{C_{i-1}}$ that f_{e_i} intersects may still contribute to the boundaries of $F_{\phi}^{C_i}$.

The time complexity of this algorithm is of the same order as the number of $f^C_{e_j}$ popped out of the stack, which is of the same order as the number of edges in the chain. Therefore, this algorithm takes $O(n)$ time, where n is the total number of vertices in C. The counterclockwise CVD of an obtuse star-shaped chain is depicted in Figure 9.

4 Circular Visibility Diagram of a Pocket

In this section, the CVDs of another class of open polygonal chain is discussed. This open polygonal chain is called a *pocket.* It has its two end points collinear with p, and p does not lie on the *lid*, the line segment connecting the two end points. As adopted earlier, the edges of a pocket are oriented counterclockwise; only visibility arcs hitting the left side (the inside) of the pocket are of interest. In the example depicted in Figure 10 (a), only counterclockwise arcs emanating from p can hit the inside of the pocket without first hitting the pocket from the outside. Such pockets are called *counterclockwise pockets.* Pockets on the opposite side of a lid, on the other hand, can only have clockwise visibility arcs hitting the inside of them and are thus called *clockwise pockets*, as illustrated by the shaded area in Figure 10 (b). The collinearity of p with the lid exhibits an essential property, which will be explained shortly and employed in the construction of the CVD for a pocket.

Lemma 4.1 *Let $\bar{uv}$ be a line segment collinear with point p, where $u \in \bar{pv}$. Then every arc emanating from p will intersect $\bar{uv}$ at most once.*

Proof: Suppose an arc emanating from p passes through $\bar{uv}$ at q. Since there can be at most two intersections between an arc and a line segment, and since the arc has

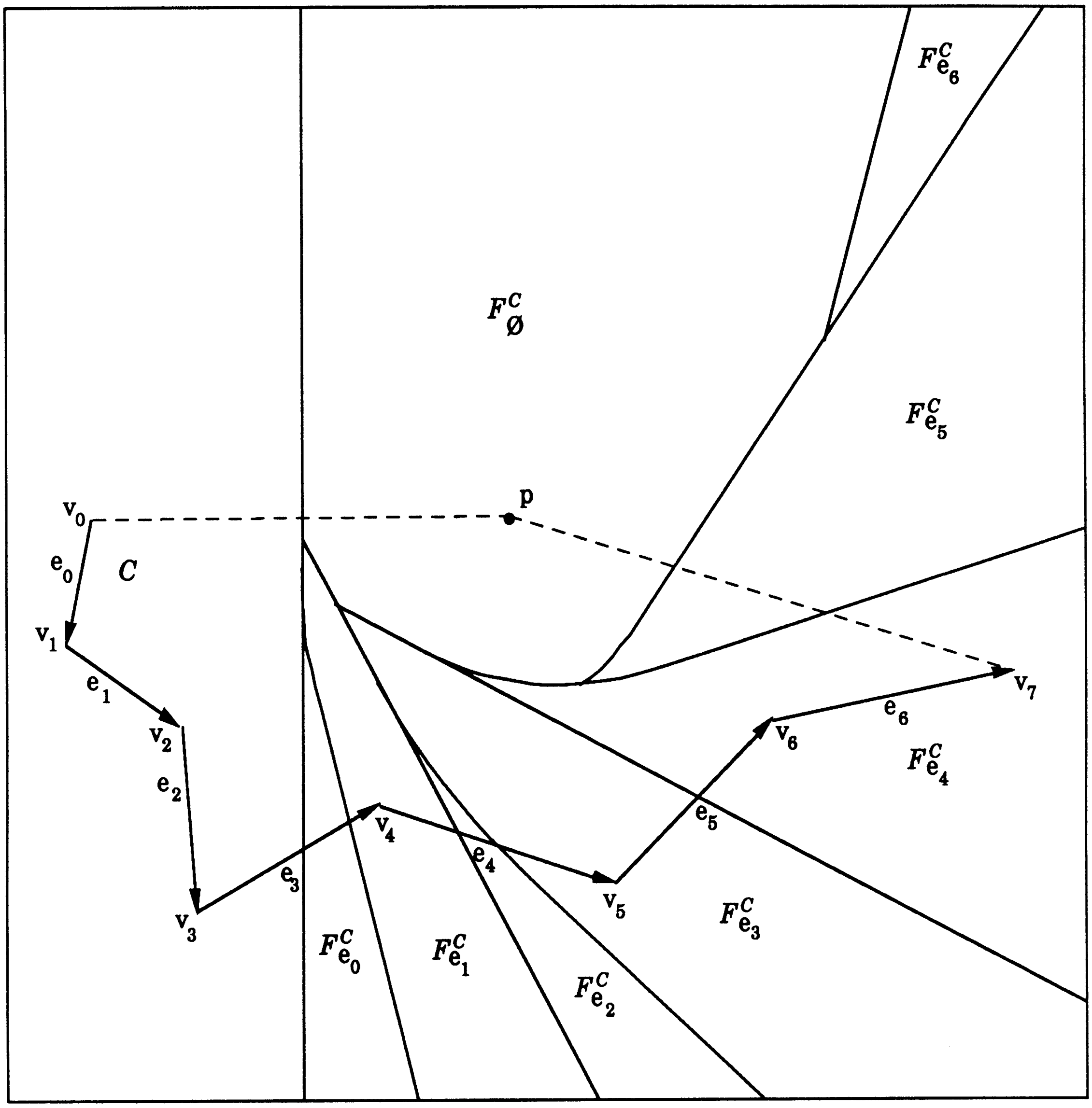


Figure 9: The CVD of an obtuse star-shaped chain.

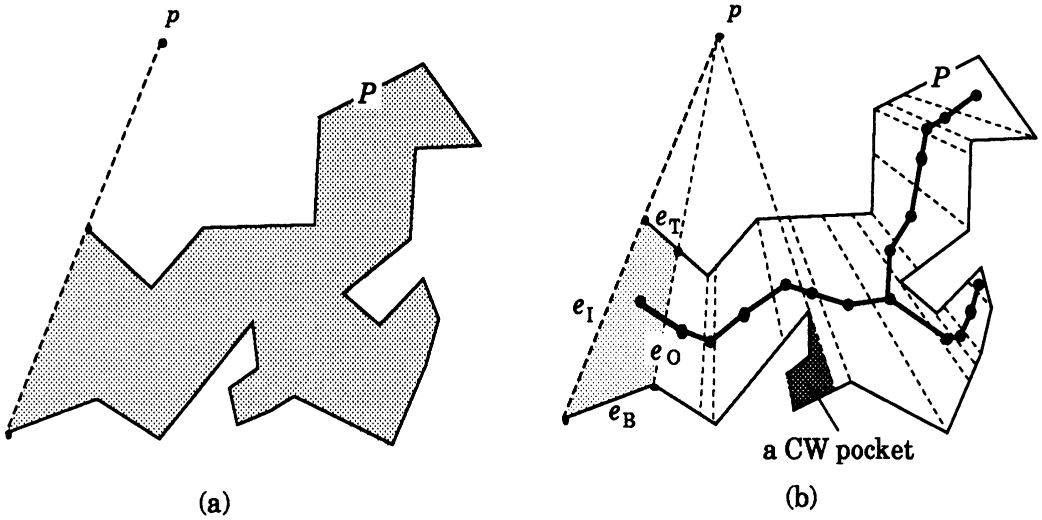


Figure 10: (a) A counterclockwise pocket. (b) The trapezoidization of a pocket.

already passed through $\bar{pv}$ at q and p, it can not have another intersection with $\bar{pv}$. As $p \notin \bar{uv}$, $\bar{uv}$ contains exactly one intersection with the arc, namely q.

□

To efficiently construct the CVD for a pocket, a partial order in which arcs emanating from p hit the edges of the pocket is required. However, the edges of a pocket do not inherently possess such an order. In the following, the edges of a pocket are decomposed into edges exhibiting partial ordering: a CVD is first computed with respect to the decomposed edges; and regions in such a CVD are then merged into the CVD of the pocket with respect to the original (undecomposed) edges.

The decomposition of the edges of the pocket is done by utilizing *vertex-edge visible pairs* joined by dotted line segments, as shown in the pocket of Figure 10 (b). A vertex-edge visible pair is a vertex and an edge which can be connected by a line segment lying entirely inside the pocket. By employing line segments whose extensions pass through p to connect all the vertex-edge pairs of the pocket, the interior of the pocket is decomposed into *trapezoids*[3], as shown in Figure 10 (b). Such a decomposition is also known as a trapezoidization[6,15,18]. Tarjan and van Wyk [18] show that a trapezoidization of a simple polygon can be done in $O(n \log\log n)$ time, where n is the total number of vertices in the simple polygon. As a recent development, the time complexity of the trapezoidization algorithm is advanced by Chazelle [6] to $O(n)$.

Each trapezoid in the pocket consists of four sides. The *in-edge* and the *out-edge* are connecting line segments of the vertex-edge visible pair, where the *in-edge* is the side with the smaller θ. Both of the *in-edge* and the *out-edge* are considered to be transparent. The *top-edge* and the *bottom-edge* are portions of the edges of the pocket, where the *top-edge* is closer to p than the *bottom-edge* is. Both of the *top-edge* and the *bottom-edge* are considered to be opaque.

The circular arcs are now classified with respect to the side of a trapezoid they hit. Let e_I, e_O, e_T, and e_B denote the *in-edge*, *out-edge*, *top-edge*, and *bottom-edge*, respectively, of a trapezoid T_i. Let $F^{T_i}_{e_I^-}$ denote the region containing all the centers about which arcs emanating from p hit e_I from the outside (right side)[4]. Since e_T and e_B are opaque and since only counterclockwise visibility arcs are employed, any arc which hits e_T, e_B, or e_O from the inside must first pass through e_I. After crossing e_I, an arc will then first hit e_T, e_B, or e_O. Those arcs that hit e_T or e_B are blocked. Those arcs that hit e_O, on the other hand, may pass through e_O (as e_O is transparent), come back through it, and hit either e_T or e_B. However, since p is collinear with e_O, by Lemma 4.1, those arcs that pass through e_O can not come back into T_i through

[3] lines originating from the same point can be viewed as in parallel.

[4] As adopted earlier, e_I is a directed edge such that the inside of the trapezoid is to the left of it. e_I^- is e_i directed in the opposite direction. In other word, visibility arcs about points in $F^{T_i}_{e_I^-}$ will hit e_I^- from the left, which is consistent with the notation of F^B_A.

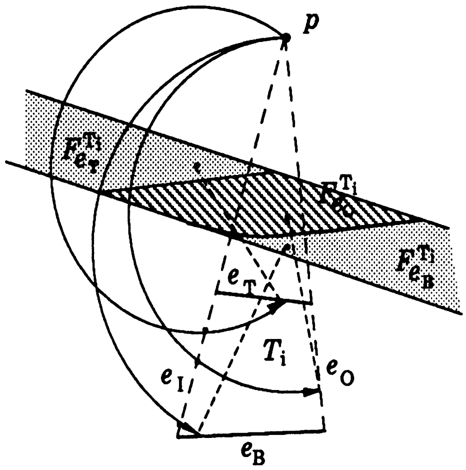


Figure 11: $F^{T_i}_{\bar{e}_I}$ is divided into $F^{T_i}_{e_T}$, $F^{T_i}_{e_B}$, and $F^{T_i}_{e_O}$.

e_O. Therefore, $F^{T_i}_{\bar{e}_I}$ can be partitioned into three mutually exclusive regions, $F^{T_i}_{e_T}$, $F^{T_i}_{e_B}$, and $F^{T_i}_{e_O}$, corresponding to arcs that pass through e_I and hit e_T, e_B, and e_O, respectively. Such a partition is illustrated in Figure 11.

A partial order in which arcs about points in $F^{T_i}_{\bar{e}_I}$ intersect e_T, e_O, and e_B is utilized to compute $F^{T_i}_{e_T}$, $F^{T_i}_{e_B}$, and $F^{T_i}_{e_O}$.

Theorem 4.2 *Let F_{e_T}, F_{e_B} and F_{e_O} be the regions containing all the points about which circular arcs hit e_T, e_B and e_O, respectively. Then,*

$$\begin{cases} F^{T_i}_{e_B} & = F^{T_i}_{\bar{e}_I} \cap F_{e_B} \\ F^{T_i}_{e_T} & = F^{T_i}_{\bar{e}_I} \cap (F_{e_T} - F_{e_B}) \\ F^{T_i}_{e_O} & = F^{T_i}_{\bar{e}_I} \cap (F_{e_O} - F_{e_B}) \end{cases}$$

Proof: Since e_T is facing away from the emanating point p, arcs which hit e_T immediately after crossing e_I will not intersect e_O or e_B afterwards. Also, by Lemma 4.1, an arc which hits e_O immediately after crossing e_I will not intersect e_B or e_T afterwards. However, since e_B is facing toward p, an arc which hits e_B immediately after crossing e_I may intersect e_O or e_T afterwards. These three relations indicate that only e_B may obstruct arcs in hitting e_O or e_T.

Since e_O and e_T do not obstruct circular arcs passing through e_I in hitting e_B, $F^{T_i}_{e_B} = F^{T_i}_{e_I^-} \cap F_{e_B}$. On the other hand, since e_B may obstruct arcs in hitting e_T or e_O, $F^{T_i}_{e_T} = (F^{T_i}_{e_I^-} \cap F_{e_T}) - F^{T_i}_{e_B}$ and $F^{T_i}_{e_O} = (F^{T_i}_{e_I^-} \cap F_{e_O}) - F^{T_i}_{e_B}$. By substituting $F^{T_i}_{e_B}$ with $(F^{T_i}_{e_I^-} \cap F_{e_B})$,

$$\begin{aligned} F^{T_i}_{e_T} &= (F^{T_i}_{e_I^-} \cap F_{e_T}) - (F^{T_i}_{e_I^-} \cap F_{e_B}) \\ &= F^{T_i}_{e_I^-} \cap (F_{e_T} - F_{e_B}). \end{aligned}$$

Similarly, $F^{T_i}_{e_O} = F^{T_i}_{e_I^-} \cap (F_{e_O} - F_{e_B})$, which completes the proof.

□

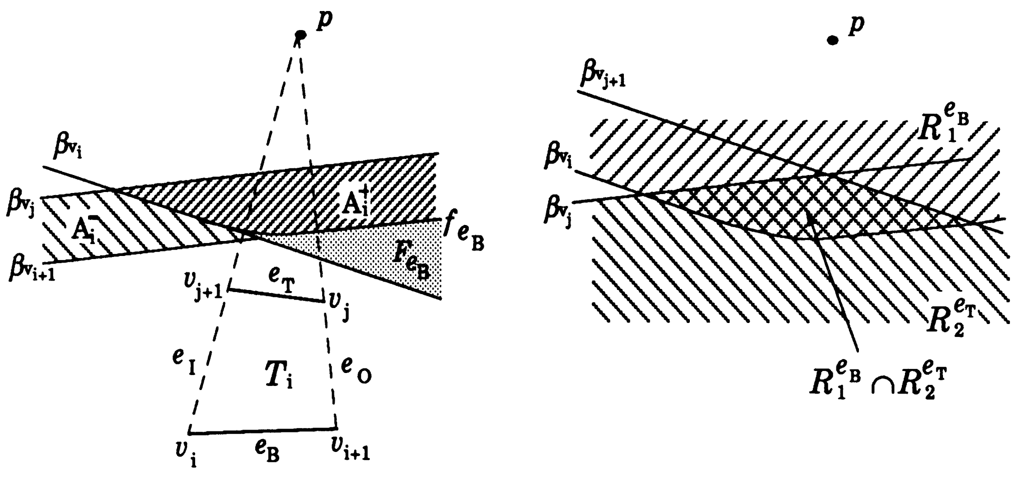
With the visibility arcs classified according to the side of a trapezoid they hit, the transition of the visibility arcs from one trapezoid to the next is now examined. Since no counterclockwise visibility arc can reach the inside of a clockwise pocket, the trapezoids in CW pockets as depicted in Figure 10 (b) are discarded from further consideration. Also, since counterclockwise visibility arcs can not reach portions of the pocket on the other side of the line coincident with the lid, such portions are discarded as well. Consequently, the in-edgs of the trapezoids to be considered span less than 180°. Without the CW pockets and portions of the pocket requiring visibility arcs to travel more than 360° to reach them, it can be safely assumed that the trapezoid containing the lid of a pocket has the *smallest* θ with respect to p.

The θs of *in-edges* exhibit a partial order in which the visibility arcs pass through these trapezoids. The same order also gives rise to the order in which arcs hit the *top-edges* and the *bottom-edges* of the trapezoids in a pocket. Such a partial order can be uniquely represented by the dual graph of the trapezoidized pocket, in which each node is associated with a trapezoid and each link with any two trapezoids sharing a side. This dual graph is clearly a partial-order tree with the root node representing the trapezoid containing the lid of the pocket[5], as shown in Figure 10 (b).

The construction of the CVD of a pocket starts from T_R, the trapezoid at the root of the partial-order tree. $F^{T_R}_{e^-_I}$ is subsequently partitioned into three regions: $F^{T_R}_{e_T}$, $F^{T_R}_{e_B}$, and $T^{T_R}_{e_O}$, by utilizing Theorem 4.2. $F^{T_R}_{e_O}$ then becomes $F^{T_1}_{e^-_I}$ of T_1, the immediate descendant trapezoid of T_R. The CVD of a pocket is constructed by orderly partitioning $F^{T_i}_{e^-_I}$ (or $F^{T_{i-1}}_{e_O}$) into $F^{T_i}_{e_T}$, $F^{T_i}_{e_B}$, and $F^{T_i}_{e_O}$ throughout the partial-order tree. It is noted that arcs which hit e_T and e_B of T_i hit the same two edges in the pocket P where T_i resides, which means $F^P_{e_T} = F^{T_i}_{e_T}$ and $F^P_{e_B} = F^{T_i}_{e_B}$. Such a partition seems to require $O(n \log n)$ time to complete[8], where n is the number of vertices of the pocket. However, by showing that $F^{T_i}_{e_O}$ is convex and utilizing the counterclockwise order of the in-edges about p, the partition can be completed in linear time.

Before verifying the two properties indicated above, the detailed construction of $F^{T_i}_{e_O}$ is first examined. Let the trapezoid of interest, T_i, be depicted as in Figure 12 (a). By Theorem 4.2, $F^{T_i}_{e_O}$ equals the intersection of $F^{T_i}_{e^-_I}$ and $(F_{e_O} - F_{e_B})$. Since F_{e_O} is a type T region bounded by β_{v_j} and $\beta_{v_{i+1}}$ and F_{e_B} is a type L region bounded by f_{e_B} and β_{v_i}, subtracting F_{e_B} from F_{e_O} always results in two separate regions, denoted as A^+_i and A^-_i as shown in Figure 12 (a). It is noted that $A^-_i \subset R^{e_B}_3$, i.e., all the arcs centered at points in A^-_i will hit e_B from the outside. Since an arc can not hit both e_I and e_B from the outside, the intersection of A^-_i and $F^{T_i}_{e^-_I}$ is therefore always empty. On the other hand, A^+_i may contribute to $F^{T_i}_{e_O}$ since $(R^{e_B}_1 \cap R^{e_T}_2) \subset A^+_i$, where

[5] It is noted that there are trapezoids with three opaque sides and only one transparent side. However, since such trapezoids only appear at the leaves of the partial order tree, they will not affect the algorithm, and therefore will not be discussed.



(a) (b)

Figure 12: (a) $(F_{e_O} - F_{e_B})$ is divided into A_i^+ and A_i^-. (b) $(R_1^{e_B} \cap R_2^{e_T})$

$(R_1^{e_B} \cap R_2^{e_T})$, as illustrated in Figure 12 (b), contains all the centers about which arcs emanating from p do not hit either e_T or e_B from the inside. Therefore, the intersection of $F^{T_i}_{e_I^-}$ and $(F_{e_O} - F_{e_B})$ can be subsituted with the intersection of $F^{T_i}_{e_I^-}$ and A_i^+. That $F^{T_i}_{e_O}$ is convex can be shown by examining the intersection of $F^{T_i}_{e_I^-}$ and A_i^+.

Theorem 4.3 *$F^{T_i}_{e_O}$, for all $T_i \in P$, is convex.*

Proof: The initial $F^{T_R}_{e_I^-}$ is a stripe bounded by two parallel lines and is therefore convex. Subsequent $F^{T_i}_{e_O}$, which is equal to the intersection of A_i^+ and $F^{T_i}_{e_I^-}$, is convex since both A_i^+ and $F^{T_i}_{e_I^-}$ are convex. Therefore, by induction, all the $F^{T_i}_{e_O}$s are convex.

□

The computing of the intersection of $F^{T_i}_{e_I^-}$ and A_i^+, for all $T_i \in P$, is dominated by the computing of the intersection points bewteen the boundary of $F^{T_i}_{e_I^-}$ and the boundary of A_i^+. It is shown in the following that, by dividing the boundary of $F^{T_i}_{e_I^-}$

into two pieces and maintaining them with two stacks, completing all the intersections can be achieved in linear time.

Since $F^{T_i}_{e_O}$ results from the intersection of $F^{T_i}_{e_I^-}$ and A_i^+, the boundary of $F^{T_i}_{e_O}$ consists of portions of the β_{v_j}s and the f_{e_i}s contributing to the boundary of $F^{T_i}_{e_I^-}$, where v_j is a vertex of the *top-edge* of a trapezoid and e_i is the *bottom-edge* of a trapezoid. Since the in-edges of the pockets are ordered about p, the perpendicular bisectors of line segments between p and the vertices of the top-edges and the bottom-edges are also ordered, respectively, by their normals (pointing toward p); consequently, the β_{v_j}s and the f_{e_i}s[6] of the A_i^+s are in slope order, respectively, following the partial ordering of the pockets. The construction of the CVD for a pocket by intersecting $F^{T_i}_{e_I^-}$ with A_i^+ is thus analoguous to constructing an upper bay and a lower bay, as described in Section 3, simultaneously.

The upper bay, maintained by a stack S_U, consists of only the β_{v_j}s whereas the lower bay, maintained by stack S_L, consists of only the f_{e_i}s. Let μ and v denote the intersections between the upper bay and the lower bay, as shown in Figure 13 (a). As the construction of the CVD proceeds, the intersection of $F^{T_i}_{e_I^-}$ and A_i^+ with boundary β_{v_t} and f_{e_B} is computed, as shown in Figure 13 (b). The intersection points between β_{v_t} and the boundary of $F^{T_i}_{e_I^-}$ are sought by checking through S_U and S_L, respectively, starting from the end of the stacks containing μ. The β_{v_j}s in S_U which do not intersect β_{v_t} are popped out sequentially until one does. Similarly, S_L is updated by popping out f_{e_i}s which do not intersect β_{v_t} until one does. The portion of β_{v_t} lying inside $F^{T_i}_{e_I^-}$ contributes a boundary curve to $F^{T_i+1}_{e_I^-}$, and is pushed into S_U. The region encompassed by this portions of β_{v_t} and the boundary curves of $F^{T_i}_{e_I^-}$ between μ and the two intersections with β_{v_t} yields $F^{T_i}_{e_T}$. (The boundary curves of $F^{T_i}_{e_I^-}$ between

[6] By construction, the slope on f_{uv} changes monotonically and is bounded by the slopes of β_u and β_v.

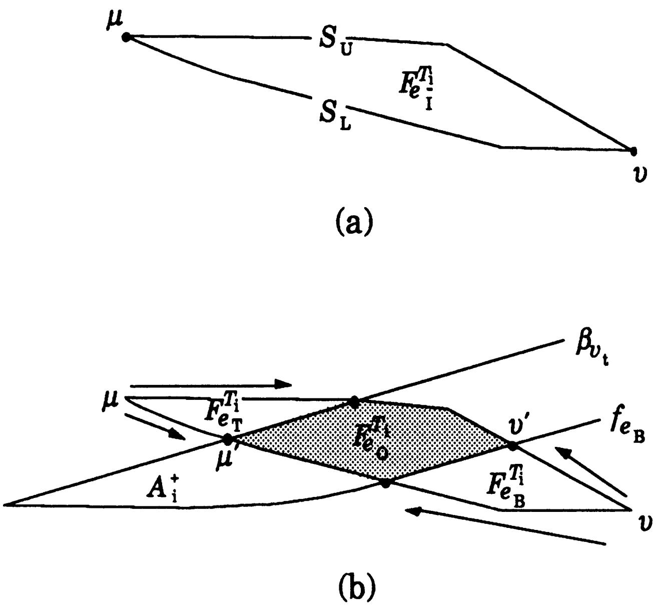


Figure 13: The partition of $F_{\bar{e}_I}^{T_i}$.

μ and the two intersections with β_{v_t} are the β_{v_j}s and f_{e_i}s being popped out of S_U and S_L when computing the intersections with β_{v_t}.) Likewise, the intersection points between f_{e_B} and the boundary of $F_{\bar{e}_I}^{T_i}$ can be identified by checking through S_U and S_L from v, the other end of the stacks. The portion of f_{e_B} lying inside $F_{\bar{e}_I}^{T_i}$ contributes a boundary curve to $F_{\bar{e}_I}^{T_i+1}$, and is pushed into S_L. The region encompassed by this portions of f_{e_B} and the boundary curves of $F_{\bar{e}_I}^{T_i}$ between v and the two intersections with f_{e_B} yields $F_{e_B}^{T_i}$. (The boundary curves of $F_{\bar{e}_I}^{T_i}$ between v and the two intersections with f_{e_B} are the β_{v_j}s and f_{e_i}s being popped out of S_U and S_L when computing the intersections with f_{e_B}.) The curves remain in S_U and S_L yeild $F_{e_O}^{T_i}$. If neither β_{v_t} or f_{e_B} intersects $F_{\bar{e}_I}^{T_i+1}$, S_U and S_L remain the same. The partition continues until all of the trapezoids are examined or when $F_{\bar{e}_I}^{T_i} \cap A_i^+$ is empty, which means that all the visibility arcs are blocked.

The time required for finishing all the partitioning is shown to be linear to the number of vertices in the pocket. Let n_U and n_L be the number of β_{v_j}s and f_{e_i}s in S_U and S_L, respectively, and let $C(n_U, n_L)$ denote the total time required for partitioning the $F^{T_i}_{\bar{e}_I}$with S_U and S_L of size n_U and n_L, respectively. Suppose i_U, i_L, j_U, and j_L curved are checked, respectively, for identifying the intersections between β_{v_t} and S_U, β_{v_t} and S_L, f_{e_B} and S_U, and f_{e_B} and S_L. Then,

$$C(n_U, n_L) = O(i_U) + O(i_L) + O(j_U) + O(j_L) + C(n_U - i_U - j_U, n_L - i_L - j_L).$$

By replacing $C(n_U, n_L)$ with $C(n_U + n_L)$,

$$C(n_U + n_L) = O(i_U + i_L + j_U + j_L) + C(n_U + n_L - i_U - j_U - i_L - j_L).$$

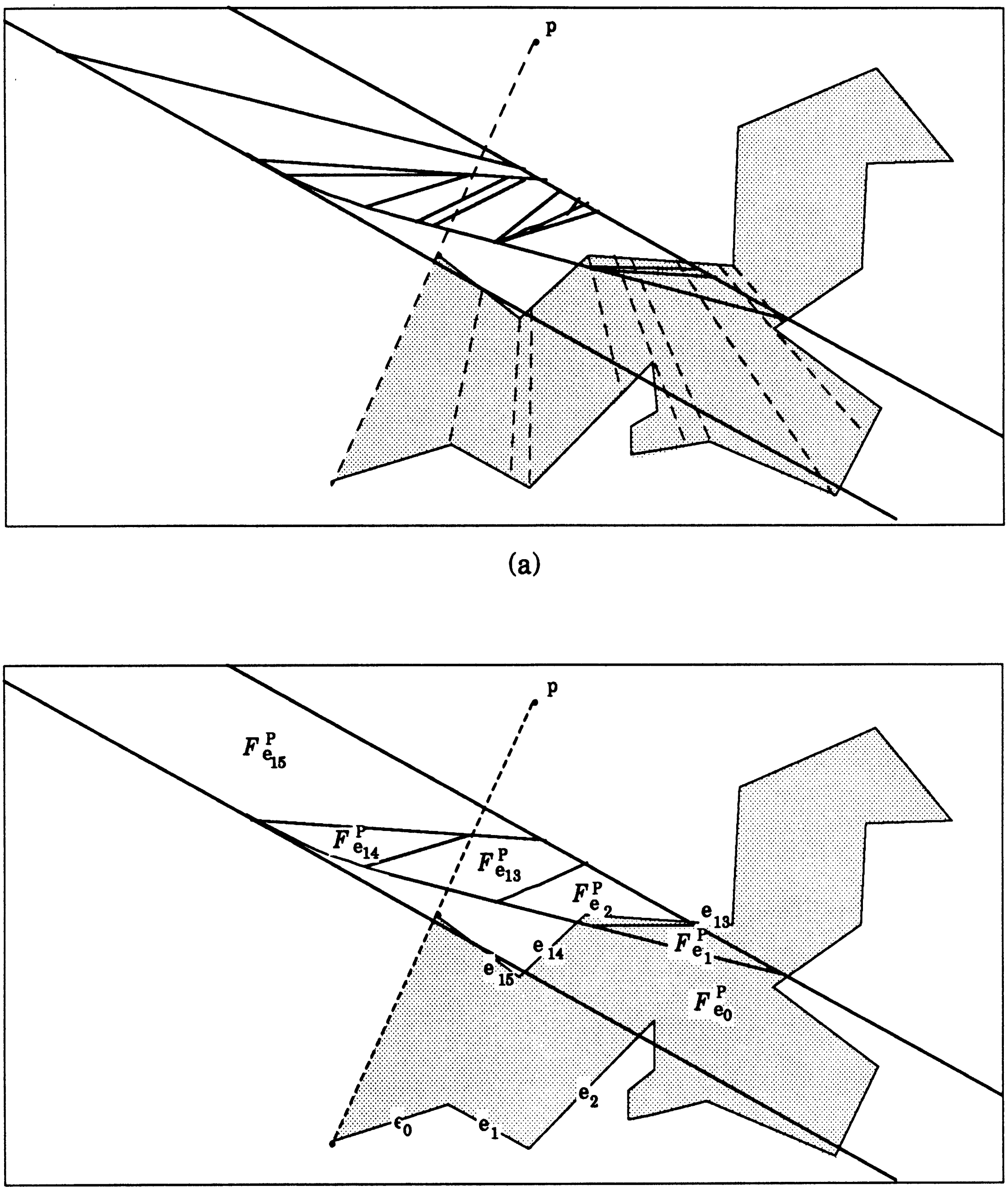
Let $(n_U + n_L)$ equal n and $(i_U + j_U + i_L + j_L)$ equal k. By substituting n and k into the formula, a recurrence formula

$$C(n) = O(k) + C(n - k)$$

is obtained. The solution to this recurrence formula is clearly $O(n)$. In other words, the plane partition, which yields the CVD of a pocket, can be constructed in $O(n)$ time, where n is the number of vertices in the trapezoidized pocket.

It is clear that the number of vertices in the original pocket and the number of vertices in the trapezoidized pocket are of the same order. Therefore, given a pocket with n vertices, the CVD of the pocket with decomposed edges can be constructed in $O(n)$ time. The CVD of the pocket with the original edges can be then obtained by merging the regions corresponding to the decomposed edges of original edges, which can be computed easily in $O(n)$ time. Therefore, the total time complexity of the algorithm is still bounded by $O(n)$.

The CVD of the pocket P in Figure 10 is depicted in Figure 14. Figure 14 (a)



(a)

(b)

Figure 14: (a) CVD of a pocket with decomposed edges. (b) CVD of a pocket with the original edges.

shows the CVD of P with the decomposed edges, and Figure 14 (b) shows the CVD of P with the original edges.

5 Circular Visibility Diagram of a Star-shaped Polygon

The notion of a *star-shaped* polygon is defined as a closed polygonal curve whose boundary is monotonic in θ about p. The CVD for such a polygon can be constructed by using the algorithm for obtuse star-shaped chains discussed in Section 3.

Given a polygon Q, star-shaped about p, it is observed that cutting Q by a horizontal line passing through p yields two obtuse star-shaped chains – the upper chain C_U and the lower chain C_L – each of which spans 180° about p. The CVDs for C_U and C_L can be constructed individually, and then merged. Clearly, if an arc does not intersect either C_U or C_L, it will not intersect Q. If an arc intersects only one of the chains, it intersects the corresponding edge in Q. If an arc intersects both chains, the center of such an arc appears in both of the CVDs of the two chains. Yet, the arc can only hit one edge in Q. This means that the overlapping of the two CVDs needs to be resolved so as to merge the two CVDs properly. To resolve the overlapping, the order in which the arc intersects the two chains needs to be determined.

Let L_V be the perpendicular line passing through p. Let F_{C_U} and F_{C_L} denote the regions containing the centers about which arcs emanate from p and hit C_U and C_L, respectively. Let $F^Q_{C_U}$ and $F^Q_{C_L}$ be the regions containing the centers about which arcs emanate from p and intersect C_U and C_L, respectively, in the presence of Q. The following lemma resolves the overlappings between the CVDs of the two chains.

Lemma 5.1 *Let q be a point in $F_{C_U} \cap F_{C_L}$. Then, if q lies to the left of L_V, $q \in F^Q_{C_U}$.*

If q lies to the right of L_V, $q \in F^Q_{C_L}$.

Proof: Arcs centered at a point to the right of L_V always go downward first from p. Therefore, if such arcs intersect both C_U and C_L, they must hit C_L first. Likewise, arcs centered at a point to the left of L_V must hit C_U first if they intersect both C_U and C_L.

□

Lemma 5.1 indicates that, in the presence of Q, arcs emanating from p and centered at points to the right of L_V can hit C_U without being blocked by C_L only if these points also lie in $F^{C_L}_{\phi}$. Similarly, arcs emanating from p and centered at points to the left of L_V can hit C_L without being blocked by C_U only if these points also lie in $F^{C_U}_{\phi}$.

Let $F^Q_{e_j}$, for all $e_j \in C_L$, be constructed first. The feasible area where $F^{C_L}_{e_j}$ can lie is the union of $F^{C_U}_{\phi}$ and the half-plane to the right of L_V. $F^{C_U}_{\phi}$ can be obtain by computing the CVD of C_U, as shown in Figure 15 (a). The result of the union of $F^{C_U}_{\phi}$ and the half-plane to the right of L_V is shown in Figure 15 (b). $F^Q_{e_j}$ can therefore be constructed along the boundary of $F^{C_U}_{\phi}$ with the procedure used for obtuse star-shaped chains. After completing constructing the $F^Q_{e_j}$s, $F^{C_L}_{\phi}$ is obtained. $F^Q_{e_i}$, for all $e_i \in C_U$, can then be constructed similarly along the boundary of $F^{C_L}_{\phi}$. Figure 16 shows the final result of the CVD of the star-shaped polygon Q.

It is noted that the construction of the CVD for Q is equivalent to constructing the CVDs for three obtuse star-shaped chains (one and a half rounds of the star-shaped polygon). Since the construction of $F^Q_{e_i}$, for all $e_i \in C_U$, and $F^Q_{e_j}$, for all $e_j \in C_L$, takes $O(n)$ time each, where n is the number of vertices of Q, the time required for constructing the CVD for a star-shaped polygon is bounded by $O(n)$.

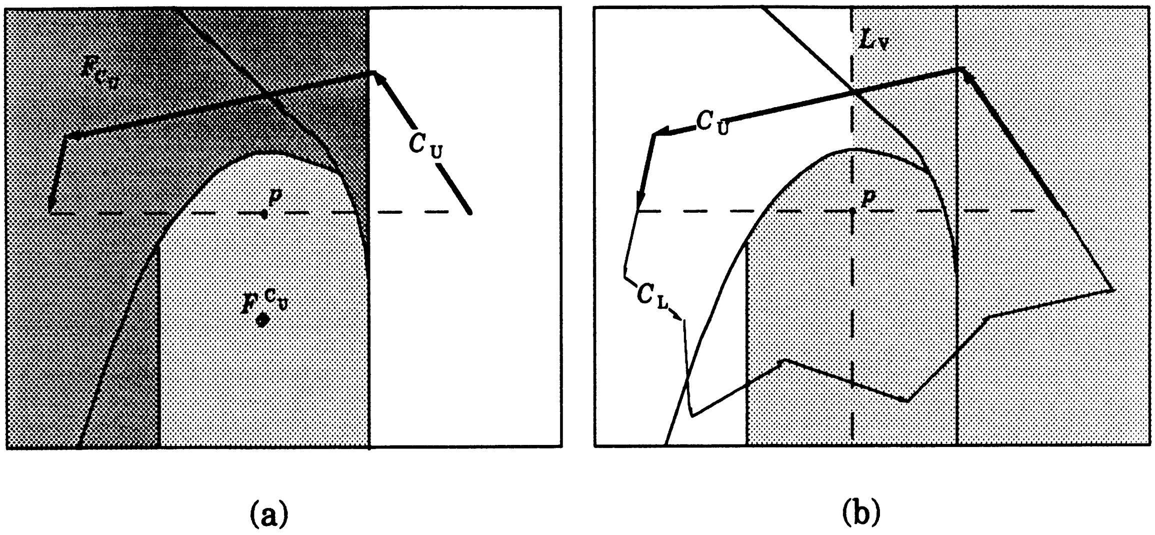


Figure 15: Construction of the CVD for the upper chain of a star-shaped polygon.

6 Circular Visibility Diagram of a Simple Polygon

Now that the linear time algorithms that compute the CVDs for a star-shaped polygon and for a pocket have been established, the utilization of these algorithms for constructing the CVD of a simple polygon is to be shown. The outline of the CVD construction procedure is illustrated as follows.

```
Algorithm CVD($p$,$Q$)
    Decompose $Q$: $Q = Q^* + P_1 + ... + P_m$
    Construct CVD($p$,$Q^*$)
    for $i = 1$ to $m$ do
        Construct CVD($p$,$P_i$)
    Merge the CVDs
end.
```

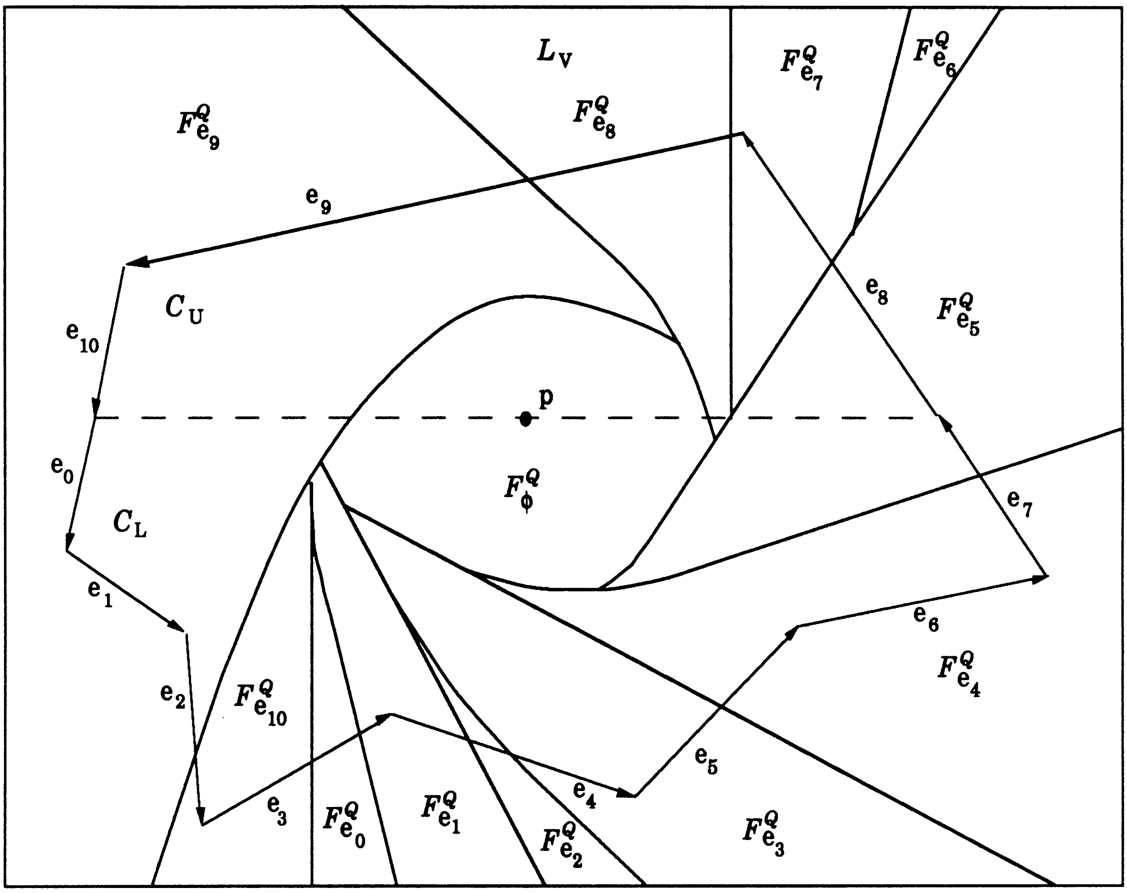


Figure 16: The CVDs of the star-shaped polygon.

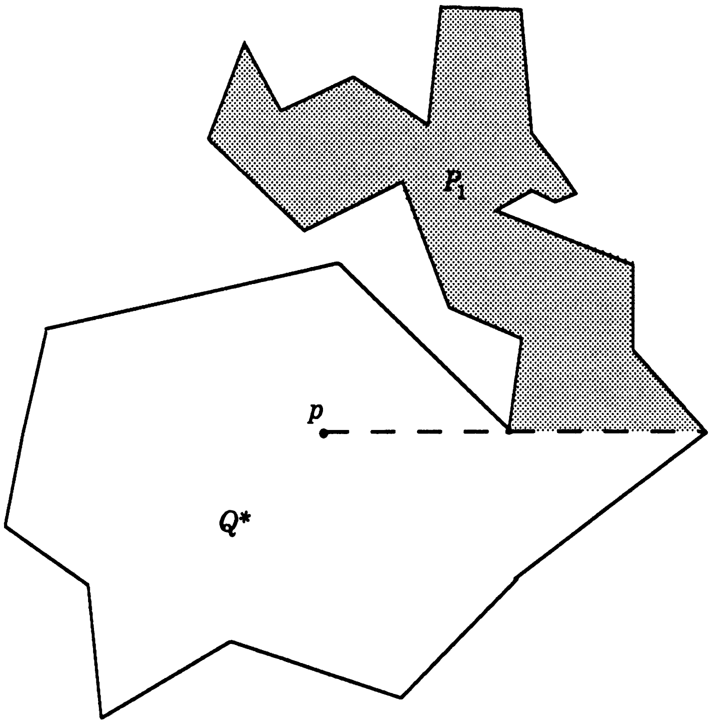


Figure 17: The decomposition of a simple polygon Q.

It is observed that any simple polygon can be decomposed into a star-shaped polygon and a number of pockets, with respect to a point inside the polygon, as shown Figure 17. Let Q^* be the star-shaped polygon obtained by computing the linear visibility polygon with respect to p. It can be constructed optimally by the algorithm developed by Lee [12] in $O(n)$ time, where n is the number of vertices of Q. Then, taking the boolean difference between Q^* and Q results in (if exists) a set of pockets P_1, P_2, ..., P_m. Since $Q^* \subset Q$, the computing of the boolean difference between them can be achieved in $O(n)$ time.

That this algorithm constructs a correct CVD of a simple polygon is easy to see. By Lemma 4.1, arcs crossing the lid of a pocket can not come back into Q^* (p is be definition in Q^*) through the lid and hit other edges of Q^*. Thus, the collection of visibility arcs that hit the edges of Q which are also the edges of Q^* are the same as those computed with respect to Q^*. On the other hand, visibility arcs that hit the edge of Q^* which is the lid of a pocket contribute to all the visibility arcs going into this pocket. The CVD for edges of Q which are in the pockets can thus be computed by further decomposing the region containing all the points about which arcs hit the lid, as described in Section 4.

It is shown in the following that the construction of the CVDs of Q^* and the pockets is bounded by $O(n)$, where n is the total number of vertices in Q. Where the CVD of Q^* can be constructed in $O(n)$ time using the algorithm described in Section 5, the computing of the CVDs of the pockets in Q requires extra effort. It is because as some of the visibility arcs to the lids may be obstructed by other edges of Q^*, the CVD region of the lid of a pocket will not be a simple Type T region. By construction, such a region is the intersection of a stripe, a region bounded by two parallel lines, and a $F_{\phi}^{C_i}$, as described in Section 3. Since both the stripe and $F_{\phi}^{C_i}$ are convex, the region containing all the points about which visibility arcs crossing the lid is convex. The algorithm for computing the CVDs for pockets is therefore applicable with starting regions being convex. The time required for constructing the CVD for a pocket becomes $O(m_i + n_i)$, where m_i is the number of vertices in pocket P_i and n_i is the number of edges of the starting region $F_{e_i}^{Q}$, where ei is the lid of P_i. Note that the sum of n_i is bounded by the total number of vertices in Q^*. Therefore, the total time required for constructing the CVDs for all the pockets is still bounded by the total number of vertices in Q.

The total time complexity for constructing the CVD of a simple polygon is the sum of the time complexity for the individual processes. Since the time complexity of all of the processes are bounded by $O(n)$, this algorithm computes the CVD of a simple polygon in linear time. The CVD of the polygon in Figure 17 is shown in Figure 18.

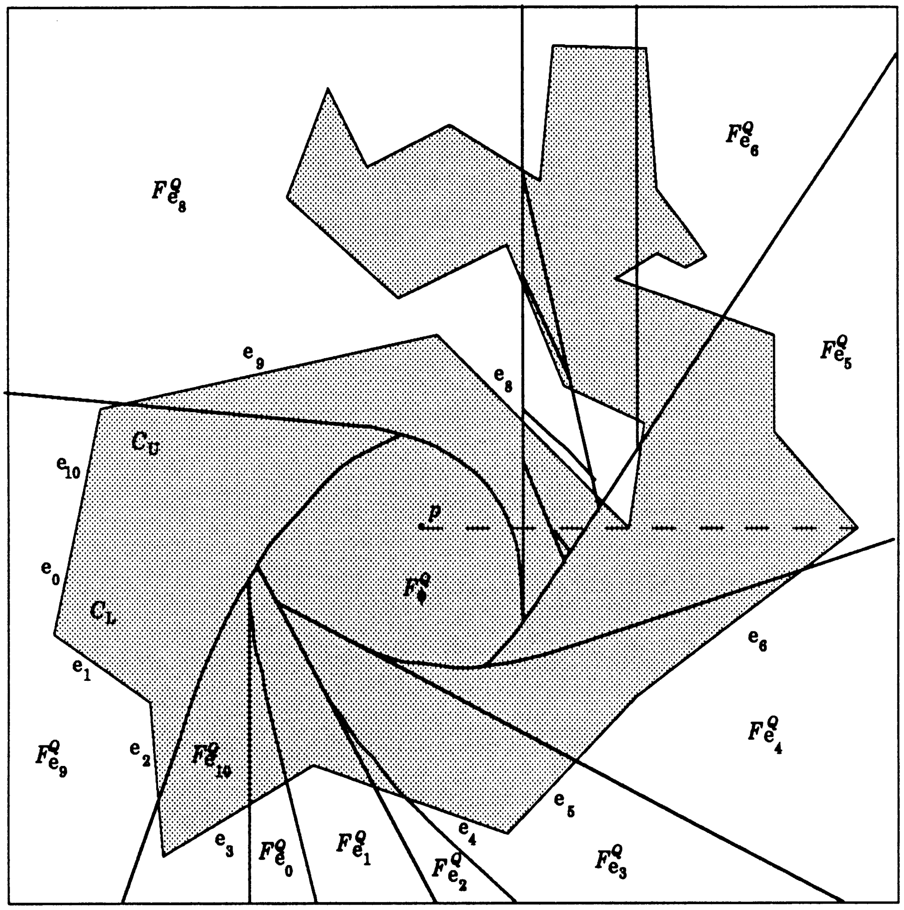


Figure 18: The CVD of a simple polygon Q.

7 Conclusion

This paper utilized the notion of circular visibility and developed an algorithm for the classification of circular visibility arcs. Such a classification is achieved by computing a plane partition of the centers about which arcs emanating from a fixed point hit a particular edge of a simple polygon. A linear time algorithm is developed to compute such a plane partition.

Acknowledgment

The authors would like to thank Jan Wolter who initiated the work.

References

[1] Agarwal P.K. and M. Sharir, "*Circular Visibility in a Simple Polygon from a Point*", Technical Report No. 527, Department of Computer Science, New York University, August 1990.

[2] Avis D. and G.T. Toussaint, "*An Optimal Algorithm for Determining the Visibility of a Polygon from an Edge*", IEEE Trans. Comput., **C-30**, 910-914, 1981.

[3] Baker, W.M., Algebric Geometry: A New Treatise on Analytical Conic Sections, George Bell and Sons, London, 1906.

[4] Chazelle, B.M., L.T. Guibas, and D.T. Lee, "*The Power of Geometric Duality*", BIT, **25**(1), 76-90, 1985.

[5] Chazelle, B.M. and L.T. Guibas, "*Visibility and Intersection Problems in Plane Geometry*", Discrete & Computational Geometry, **4**, 551-581, 1989.

[6] Chazelle, B.M., "*Triangulating a Simple Polygon in Linear Time*", Discrete & Computational Geometry, **6**, 485-524, 1991.

[7] Edelsbrunner, H. and L.J. Guibas, "*Topologically Sweeping an Arrangement*", Proc. 18th Ann. ACM Sympos. Theory Comput., 389-403, 1986.

[8] Edelsbrunner, H., Algorithms in Combinatorial Geometry, Springer-Verlag, 1987.

[9] ElGindy, H. and D. Avis, "*A Linear Time Algorithm for Computing the Visibility Polygon from a Point*", J. Algorithms, **2**, 186-197, 1981.

[10] Guibas, L., J. Hershberger, D. Leven, M. Sharir, and R. Tarjan, "*Linear-Time Visibility and Shortest Path Problems Inside Triangulated Simple Polygons*", Algorithmica, **2**, 209-233, 1987.

[11] Lee, D.T. and F.P. Preparata, "*An Optimal Algorithm for Finding the Kernel of a Polygon*", J. ACM, **26**, 415-421, 1979.

[12] Lee, D.T., "*Visibility of a Simple Polygon*", Computer Vision, Graphics, and Image Processing, **22**, 207-221, 1983.

[13] Lee, D.T. and Y.T. Ching, "*The Power of Geometric Duality Revisited*", Information Processing Letters, **21**, 117-122, 1985.

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

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

[16] Suri, S., "*A Linear Time Algorithm for Minimum Link Paths inside a Simple Polygon*", Computer Vision, Graphics, and Image Processing, **35**, 99-110, 1986.

[17] Suri, S. and J. O'Rourke, "*Worst-Case Optimal Algorithms for Constructing Visibility Polygons with Holes*", Proc. 2nd ACM Symp. Comp. Geom., Yorktown Heights, 14-23, 1986.

[18] Tarjan, R.E. and C. van Wyk, "*An $O(n \log\log n)$-time Algorithm for Triangulating a Simple Polygon*", SIAM J. Computing, **17**(5), 143-178, October 1985.