4. Conclusions

We have presented an optimal algorithm for determining the visibility of a polygon from a given edge. In the case where a polygon is not visible from an edge *uv*, it is natural to define a *weak visibility polygon* $V(P, uv)$ as the set of all points of *P* visible from at least one point on *uv*. An open problem which is a natural extension of our work, would be to develop a linear algorithm to find $V(P, uv)$. Another interesting open question would be to determine a minimal set of edges from which *P* is visible. It is known that in the worst case a guard may have to visit $\lfloor n/3 \rfloor$ locations in order to observe an *n*-sided polygon (Chvátal [9]). A final, more general problem than that considered here is: given a polygon does there exist an edge from which the polygon is weakly visible. The corresponding problem for strong and complete visibility can be solved in linear time by using the kernel finding algorithm of Lee and Preparata [5]. One of the motivations for this paper relates to the notion of "external visibility" of polygons (Toussaint [10]). Our algorithm may be used to determine in linear time whether a polygon is externally visible.

5. Acknowledgment

The authors would like to thank Hossam ElGindy for careful reading of the manuscript and for programming the algorithm in FORTRAN. They are also indebted to one of the referees for many helpful suggestions, including the simplification of the proofs of Lemmas 2.2 and 2.3.

6. References

[1] H. Freeman and P. P. Loutrel, "An algorithm for the two-dimensional "hidden line" problem," *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 6, pp. 784-790, 1967.

[2] L. Davis and M. Benedikt, "Computational models of space: Isovists and isovist fields," *Comput. Graphics Image Proc*., vol. 11, pp. 49-72, 1979.

[3] N. J. Nilsson, "A mobile automaton: An application of artificial intelligence techniques," in *Proc. IJCAI*, 1969, pp. 509-520.

[4] M. I. Shamos, "Problems in computational geometry," Carnegie-Mellon Univ., Pittsburg, PA, 1975, revised 1977.

[5] D. T. Lee and F. P. Preparata, "An optimal algorithm for finding the kernel of a polygon," *J. Ass. Comput. Mach*., vol. 26, no. 3, pp. 415-421, 1979.

[6] F. A. Valentine, "Minimal sets of visibility," *Proc. Amer. Math. Soc.*, vol. 4, pp. 917-921, 1953.

[7] E. Buchman and F. A. Valentine, "External visibility," *Pac. J. Math.*, vol. 64, no. 2, pp. 333-340, 1976.

[8] F. A. Valentine, *Convex Sets*. New York; McGraw-Hill, 1964.

[9] V. Chvátal, "A combinatorial theorem in plane geometry," *J. Combinatorial*, *Th. B.*, vol. 18, no. 1, pp. 39-41, 1975.

[10] G. T. Toussaint, "A hierarchy of simple polygons," to be published.

[11] H. El Gindy and D. Avis, "A linear algorithm for determining the visibility polygon from a point," *J. Algorithms*, 1981.

[12] G. T. Toussaint, "Pattern recognition and geometrical complexity," in *Proc. 5th Int Conf. Pattern Recog*., Miami Beach, FL, Dec. 1980.

```
procedure VISIBILITY
  call PREPROCESS
  call RIGHTSCAN
  call LEFTSCAN
  for i = 1 to n do if r_i left of l_i terminate "no visibility";
  r ← p_1; l ← p_n;
  for i = 2 to n - 1 do if r_i is left of r do r ← r_i;
                        if l_i is right of l do l ← l_i;
  if l = p_n and r = p_1 terminate "complete visibility";
  if l is left of r terminate "strong visibility from" l, "to," r;
  terminate "weak visibility";
end
```

It can be easily verified that VISIBILITY runs in O(n) time. Thus, we may state the main result of the paper.

Theorem 3.1: The procedure VISIBILITY determines in O(n) time whether P is weakly, strongly, or completely visible from a given edge.

As a final point of interest, we give another characterization of visibility from an edge. Recall from Lemma 2.4 that P is *weakly visible* from uv if and only if every vertex of P is weakly visible from uv.

Theorem 3.2: P is *strongly visible* from uv if and only if for every pair of vertices in P, there is a point on uv from which they are visible.

Proof: Let y and z be two vertices of P visible from uv. We define l_y, r_y, l_z, r_z as before. If y and z are visible from some point $w \in uv$, it follows that $l_y r_y \cap l_z r_z \neq \emptyset$. Thus, if every pair of vertices y, z is visible from a point in uv, it follows that every pair of segments, $l_y r_y$, $l_z r_z$ has a non-empty intersection. From Helly's theorem [8] we have that

$$uv \cap \left\{ \bigcap_{k=1}^{n} l_k r_k \right\} \neq \emptyset$$

The theorem now follows from Lemma 2.3.

Lemma 3.1: If at some iteration, RIGHTSCAN terminates in step 2), then s is not visible from uv.

Proof: If RIGHTSCAN terminates in step 2), then rst is a left turn and xst is a right turn. The situation is illustrated in Fig. 6. Suppose that s is visible from uv, and consider any visibility line sw from s to uv. This line enters the closed polygonal region bounded by $H = rs \cup RC(s, r)$. If $v = r$ we have an immediate contradiction. Since the visibility line sw lies between st and sx, w cannot lie on uv. If $v \neq r$, then $uv \cap H = \emptyset$, and by the Jordan Curve Theorem, sw must leave the region H. Hence, sw intersects $RC(s, r)$, contradicting the fact that it is a visibility line. Hence, s is not visible from uv, proving the lemma.

Lemma 3.2: If at some iteration, RIGHTSCAN terminated in step 4), then t is not visible from uv.

Proof: Suppose RIGHTSCAN terminates in step 4) with $r_t \notin uv$ and suppose t is visible from a point $w \in uv$. Then the line segment tw intersects the internal convex path from t to v; see Fig. 7. Since xst is a left turn, it follows that vertex s lies inside the polygon $T = tw \cup LC(t, w)$. Thus, $w \neq v$, $v \notin T$, and the Jordan Curve Theorem implies that the chain $RC(s, v)$ intersects the line segment tw, contradicting the fact that t is visible from w. Thus, t is not visible from uv.

Lemma 3.3: If both RIGHTSCAN and LEFTSCAN terminate normally, and for every vertex t of P, r_t is to the right of l_t, then r_t and l_t are, respectively, the right and left intercepts of t.

Proof: Consider any vertex t of P, and assume that the conditions of the lemma hold. Let w be any point in the interval $l_t r_t$. We will show that wt lies inside P. Suppose that the chain $RC(t, v)$ crosses wt. Then the internal convex chain from t to v must cross tw. But, by construction tr_t lies to the right of tw and therefore the convex chain from t to v must cross tr_t. This is a contradiction, thus $RC(t, v)$ does not cross tw. Similarly, the left chain $LC(t, u)$ cannot cross tw, and hence tw lies inside P.

On the other hand, consider any point $w \in r_t v$ with $w \neq r_t$. Then by the argument of Lemma 3.2, tw crosses the convex chain from t to v and hence $RC(t, v)$. Thus, w is not visible from t. Similarly, if $w \in ul_t$ with $w \neq l_t$, then tw intersects $LC(t, u)$ and w is not visible from t.

Lemma 3.4: If both RIGHTSCAN and LEFTSCAN terminate normally and for some vertex t of P, r_t is to the left of l_t, then t is not visible from uv.

Proof: Consider any point $w \in uv$. If w lies to the right of r_t, then by the argument of Lemma 3.2, tw intersects the chain $RC(t, v)$. On the other hand, if w is to the left l_t, then tw intersects the chain $LC(t, u)$. But w must either lie to the right of r_t or to the left of l_t under the conditions of the lemma. Thus, tw intersects P and since w was arbitrary, t in not visible from uv.

We can now state an algorithm for determining edge to polygon visibility.

while top ≠ 1 **and** *rst* is a right turn

do top ← top - 1; *s* ← STACK(top);

if top ≠ 1 **then** *r* ← STACK(top-1); **end**;

4) (Compute right intercept and test whether it lies on *uv*)

Compute the intercept r_t of the half-line from *t* through *s* with the line through *uv*;

If $r_t \notin uv$ **then terminate** “no visibility”;

5) (Store *t* and move to next vertex)

top ← top+1; STACK(top) ← *t*; *r* ← *s*;

s ← *t*; *t* ← *t*+1;

if *t* ≠ *n* **go to** 2.

The procedure LEFTSCAN is similar. The correctness of the algorithm follows from the following four lemmas.

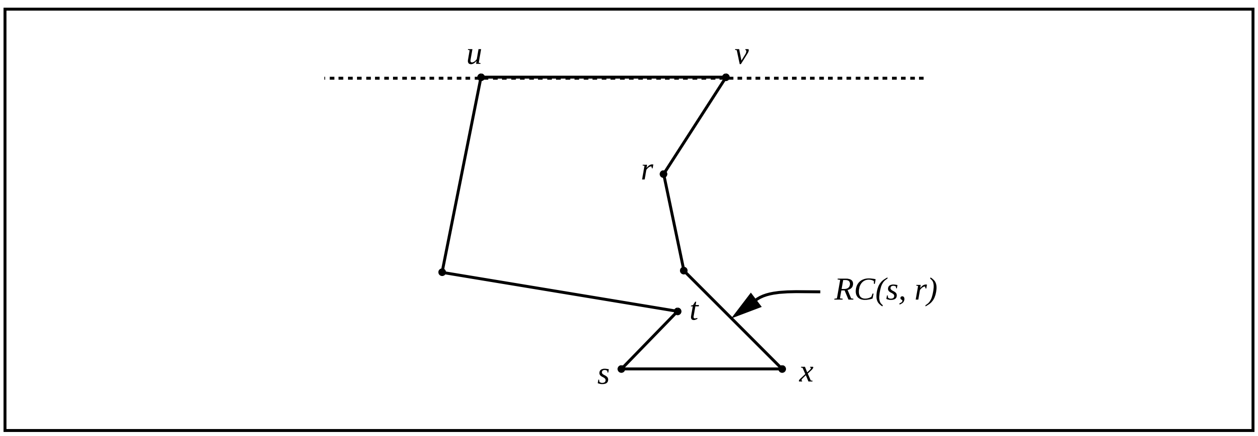

Fig. 6.

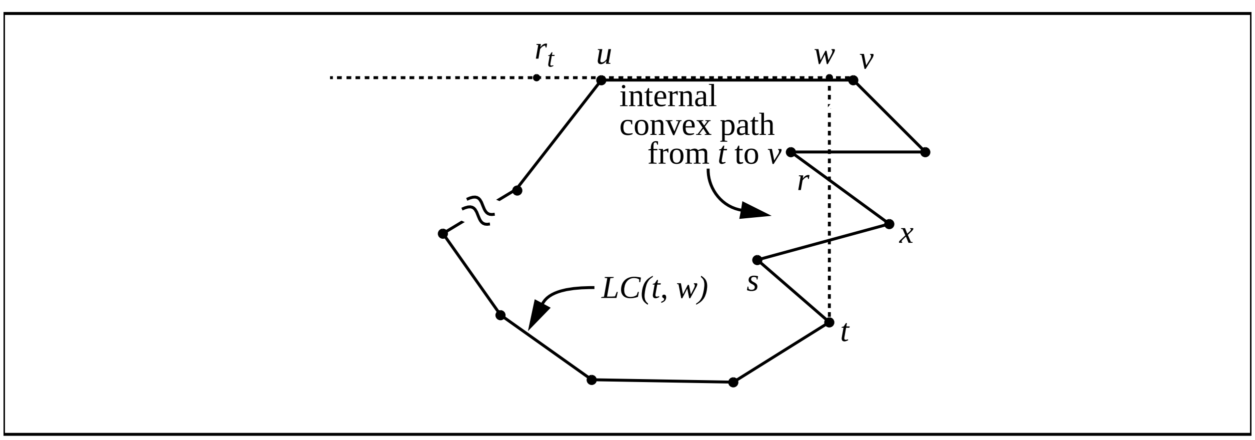

Fig. 7.

is easily seen that the vertices in region A are visible from v if and only if they are in sorted angular order about u. The same applies to region B, with vertex replacing vertex u. Let t be the intersection, if any, of the left extension of uv, and the boundary of P. Similarly, let w be the intersection, if any, of the right extension of uv and the boundary of P. Define a new polygon C by $C = \{t, u, v, w, LC(w,t)\}$. Referring to Fig. 5, $C = \{t, u, v, 4, 5, 6, 7\}$. C has the property that all of its vertices lie on the same side of the line through uv. A polygon with such a property is said to be in *standard form*, and our main algorithm will be designed to work on such polygons. It is easy to construct a linear routine PREPROCESS that: 1) puts P in standard form, and 2) for each vertex x in regions A and B, either computes $r_x = l_x = u$ or v, or determines that x is not visible from uv. The details of such a routine will be omitted.

The main algorithm consists of two scans of the vertices of P, which is in standard form. In the first scan we traverse the polygon from v to u in a clockwise orientation, successfully computing right intercepts. If we find a vertex x whose right intercept does not lie in the segment uv, then we terminate with “no visibility.” The scan procedure uses a stack to keep track of what may be considered an “internal” convex hull of vertices of P between v and the current vertex x. Given the convex path between x and v we may readily find the right intercept r_x by: 1) finding the vertex x’ adjacent to x on the convex path to v, and 2) extending the line through xx’ to intersect the line through uv. If r_x lies in the segment uv we proceed to the next vertex; otherwise we terminate the “no visibility.” The second scan is from u to v in counterclockwise orientation, in which we compute the left intercepts l_x.

We make the simplifying assumption that the vertices of P are numbered 1 to n in clockwise order around P, and the edge uv in question is the edge joining vertex n to vertex 1. The only data structure required is a stack called STACK which can hold up to n elements. Given three points $r = (x_i, y_i)$, $s = (x_j, y_j)$, and $t = (x_k, y_k)$, let

$$S = x_k(y_i - y_j) + y_k(x_j - x_i) + y_j x_i - y_i x_j.$$

We say that rst is a *right turn* if S is negative, and that rst is a *left turn* if S is positive. The three points are *collinear* whenever S is zero. We can now present the algorithm RIGHTSCAN.

procedure RIGHTSCAN

1) (Initialize)

$r \leftarrow$ STACK(1) $\leftarrow 1$;

$s \leftarrow$ STACK(2) $\leftarrow 2$;

$t \leftarrow 3$; $v \leftarrow 1$; $u \leftarrow n$; top $\leftarrow 2$;

2) (See if t is contained in the convex path determined so far)

$x \leftarrow s$-1;

if rst is a left turn **and** xst is a right turn **then terminate** “no visibility”;

3) (If rst is a right turn, backtrack the stack to make the path convex)

The above propositions suggest an algorithmic approach for determining polygonal visibility. For each vertex try to compute the right and left intercepts. If each vertex is visible, we can use the intercepts to test for strong and/or complete visibility. These ideas are formulated in the next section.

3. An Algorithm for Edge-Polygon Visibility

As we have seen in Section 2, we can determine visibility from a given edge *uv* from a knowledge of the right and left intercepts of each visible vertex. In this section we show how to compute these intercepts in O(*n*) time.

The first step is a preprocessing step that we use to put the polygon in "standard form." This step simplifies the main algorithm, yielding an easier proof of correctness. Consider the polygon *P* in Fig.5. It is clear that the vertices in region *A* are visible from edge *uv* if and only if they are visible from vertex *u*. Furthermore, the right and left intercepts of such vertices are the vertex *u*. We observe that if *P* is visible from *uv* in any sense, then the boundary of *P* can cross the line through *uv* at most once to the right of *v*, and at most once to the left of *u*. When this is the case, it

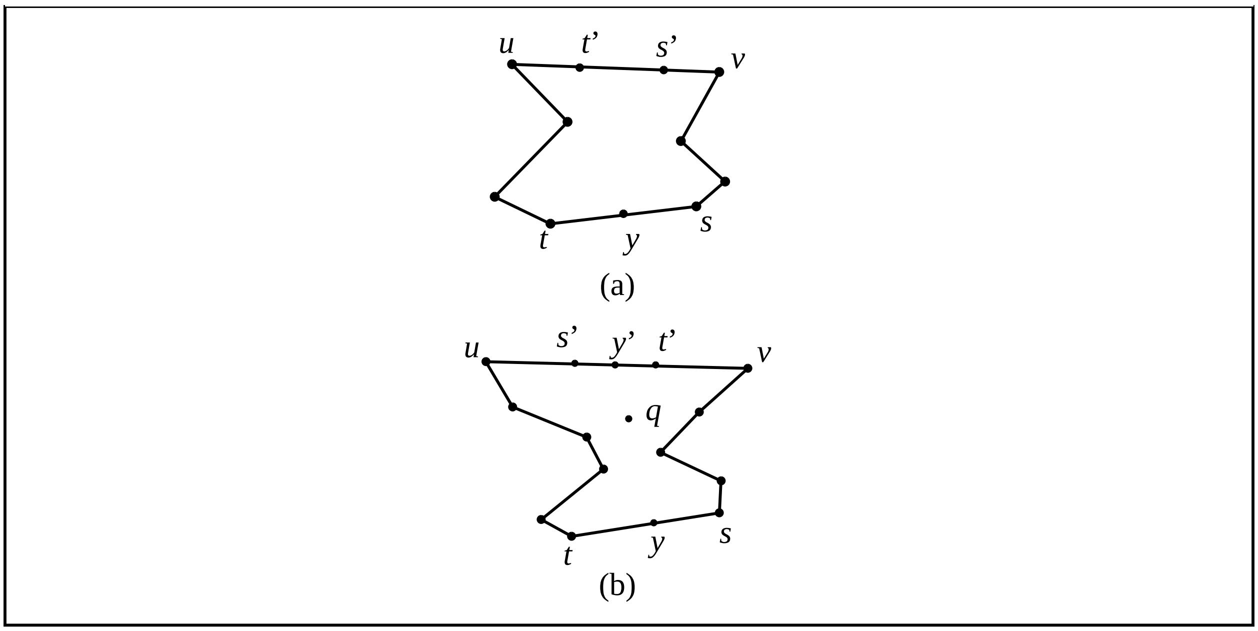

Fig. 4.

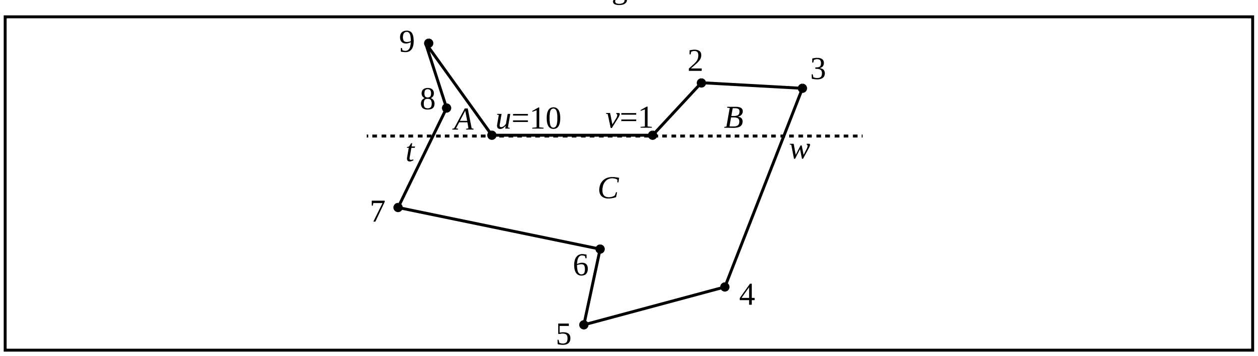

Fig. 5.

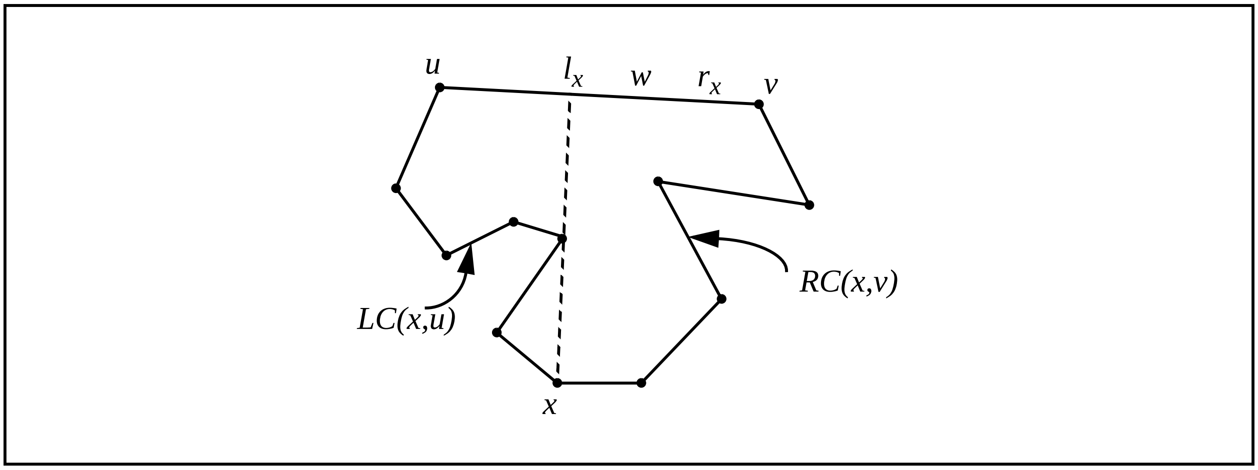

Fig. 3.

Lemma 2.3: Suppose all vertices of *P* are visible from *uv*. *P* is strongly visible from *uv* if and only if

$$l_{p1}r_{p1} \cap l_{p2}r_{p2} \cap \ldots \cap l_{pn}r_{pn} \cap uv \neq \dot{\emptyset}$$

Proof: If *P* is strongly visible from *uv*, then there exists a $w \in uv$ such that every vertex of *P* is visible from *w*. Thus, $w \in l_{pi}r_{pi}$ for $i = 1, 2, ..., n$ and the intersection (1) is non-empty.

On the other hand, suppose (1) is true and let *w* be some point in the intersection. By the remarks preceding Proposition 1, we need only show that every boundary point of *P* is visible from *w*. Let *st* be any edge of *P*. Since both *s* and *t* are visible from *w*, Lemma 2.1 implies that the entire edge *st* is visible from *w*. Thus, the entire boundary of *P* is visible from *w*, proving the “if” part of the lemma.

The following lemma completes our characterization of edge visibility.

Lemma 2.4: *P* is weakly visible from *uv* if and only if every vertex of *P* is weakly visible from *uv*.

Proof: The necessity of the condition is implied by the definitions. For the sufficiency, by Proposition 1 we need only consider a point *y* on the boundary of *P*. We will show that *y* is visible from some point on *uv*. Suppose *y* lies on the edge *st*. Then *s* is visible from some point $s' \in uv$ and *t* is visible from some point $t' \in uv$. There are two cases depending on whether or not *ss*’ and *tt*’ intersect inside *P*. These cases are illustrated in Fig. 4(a) and (b).

If *ss*’ does not intersect *tt*’, then by the argument used above, the quadrilateral *t*’*s*’*st* must lie inside *P*, and hence *y* is visible from *uv*. In case (b), suppose *ss*’ intersects *tt*’ at *q* inside *P*. It follows that edges *tq*, *qs*, *st*, and *s*’*q*, *qt*’, *t*’*s*’ all lie in *P*. Therefore, the triangles *s*’*qt*’ and *sqt* lie inside *P*. Now extend *y* through *q* to a point *y*’ on *uv*. It follows that *yy*’ lies inside *P*, hence *y* is visible from *uv*.

u' is visible from some point *u*" on *uv*. There are two cases depending on whether or not *u*'*u*'' intersects *v*'*v*'', and these are illustrated in Fig. 2(a) and (b). In case (a), consider the simple polygon *T*{*v*'', *u*'', *u*', *y*, *v*'}. We may assume that *y* does not lie on either the line segment *u*'*u*'' or the line segment *v*'*v*'', for otherwise the proposition is immediate. It is clear that the boundary of *P* cannot intersect the visibility lines *v*'*v*'' and *u*'*u*''. Similarly, by construction the boundary of *P* cannot intersect *yv*' or *yu*'. Thus, *T* lies inside *P*. Since *T* is a pentagon with only one reflex vertex, namely, *y*, it follows that *y* is visible from any boundary point of *T*, and hence from *v*''*u*''. In case (b), suppose *u*'*u*'' intersects *v*'*v*'' at *q* inside *P*. It follows that the possibly degenerate triangle *T* = {*u*'', *q*, *v*''} lies inside *P*. If $y \in T$ then we are done. Otherwise, by construction *Q* = {*v*', *y*, *u*', *q*} is a quadrilateral that also lies inside *P*. Extend *y* through *q* to a point *y*'' on edge *uv*. Then *yy*'' lies inside *P* and *y* is visible from *uv*. Since *y* was any point in *P*, the "if" part of the proposition follows. The "only if" part follows trivially from the fact that the boundary of *P* is contained in *P*.

We will assume for convenience that the origin of our coordinate system is at *u* and that the edge *uv* lies along the positive *x*-axis. Following Shamos [4], we denote by *V*(*P*, *x*) the visibility polygon of *x*, which is the set of all points in *P* visible from *x*. Between any two vertices *x* and *y* of *P* there exists two chains of vertices: the *left chain LC*(*x*, *y*) and the *right chain RC*(*x*, *y*). In *LC*(*x*, *y*) the interior of *P* lies to the right as the vertices are traversed from *x* to *y*, whereas in *RC*(*x*, *y*) the interior of the polygon lies to the left.

Let *x* be any vertex of *P* that is visible from some point, say *w*, on the segment *uv*. We define the *right intercept* r_x as that point on *uv* farthest to the right of *w* that is visible from *x*, or equivalently, from which *x* is visible. We define the *left intercept* l_x as that point on *uv* farthest to the left of *w* that is visible from *x*. These definitions are illustrated in Fig. 3. Note that the possibly degenerate triangle xr_xl_x lies inside *P*. It is possible that $r_x = v$ and $l_x = u$. In fact, this condition is satisfied for all vertices *x* if and only if *P* is completely visible from *uv*, as we now demonstrate. To avoid boundary conditions, we define $r_u = r_v = v$ and $l_u = l_v = u$.

Lemma 2.1: Let *st* be any edge of *P*, and let *x* be any point in *P*. Then if both *s* and *t* are visible from *x*, so is the entire edge *st*.

Proof: Consider the triangle *T* = {*x*, *s*, *t*}. By the hypothesis of the lemma, the boundary of *P* does not intersect the open segments *xs* and *xt*. Since *st* is an edge of *P*, it follows that *T* lies inside *P*. Hence, the lemma follows.

Lemma 2.2: *P* is completely visible from *uv*, if and only if, for all vertices *x* of *P*, $r_x = v$ and $l_x = u$.

Proof: If *P* is completely visible from *uv*, then for every vertex *x* of *P*, *x* must be visible from *u* and *v*, so $r_x = v$ and $l_x = u$, thus proving the "only if" part of the lemma.

On the other hand, suppose for all vertices *x* of *P*, $r_x = v$ and $l_x = u$. Let *w* be any point of *uv*, and let *st* be any edge of *P*. Since both *s* and *t* are visible from *w*, Lemma 2.1 implies that the entire edge *st* is visible from *w*. Thus, the boundary of *P*, and hence *P* itself, is completely visible from *uv*, proving the "if" part if the lemma.

The left and right intercepts are also of use in characterizing strong visibility.

2. Definitions and Preliminary Results

Let *P* denote a simple planar polygon which is represented by a set of *n* points $p_1, p_2, ..., p_n$ in the Euclidean plane. We assume that the points are given in clockwise order, so that the interior of the polygon lies to the right as the boundary of the polygon is traversed. We say that a line segment lies *inside* *P* if the interior of the line segment lies in the interior of *P*. Similarly, a simple polygon *Q* lies inside *P* if the interior of *Q* lies in the interior of *P*.

Two points are said to be *visible* if the line segment joining them lies inside *P*. In this paper we discuss visibility of *P* from some fixed edge *uv* of *P*. We begin by giving three natural definitions of visibility from an edge.

1) *P* is said to be *completely visible* from an edge *uv* if for every $z \in P$ and every $w \in uv$, *w* and *z* are visible.

2) *P* is said to be *strongly visible* from an edge *uv* if there exists a $w \in uv$ such that for every $z \in P$, *z* and *w* are visible.

3) *P* is said to be *weakly visible* from an edge *uv* if for each $z \in P$, there exists a $w \in uv$ (depending on *z*) such that *z* and *w* are visible. This latter definition has appeared previously in mathematics literature [6]. In Valentine's terminology the edge *uv* is a "set of visibility" of *P*. In [6] Valentine characterizes minimal sets of visibility. For additional types of external visibility of sets in two and higher dimensions, see Buchman and Valentine [7].

These definitions are illustrated in Fig.1. As motivation for the definition, consider the placement of a guard on edge *uv*, whose job is to observe the entire polygon *P*. If *P* is completely visible from *uv*, the guard can be positioned at any location on *uv*. If *P* is strongly visible from *uv*, then there always exists at least one fixed location *w* on *uv* from which the guard can observe *P*. Finally, with only weak visibility, it is necessary for the guard to patrol along some section of *uv* in order to observe the entire polygon.

Lee and Preparata [5] have found a linear algorithm for determining the kernel of a polygon. Their algorithm can also be used for testing both strong and complete visibility. First find the kernel and then determine its intersection with the given edge *uv*. The algorithm given in their paper does not appear to be useful in determining weak visibility.

We will begin by making a simplification which is intuitively satisfying. We will show that a polygon *P* is visible in any of the three senses given above if and only if the boundary of *P* is visible in the corresponding sense. This fact follows easily from the definition in the cases of complete and strong visibility.

Proposition 1: *P* is weakly visible from *uv* if and only if the boundary of *P* is weakly visible from *uv*.

Proof: Suppose that the boundary of *P* is weakly visible from *uv*. Let *y* be any point in the interior of *P*. We will show that *y* is visible from some point on *uv*. (1)

First, extend *uy* to the nearest point *u*' on the boundary of *P*. Similarly, extend *vy* to the nearest point *v*' on the boundary of *P*. By assumption, *v*' is visible from some point $v'' \in uv$ and

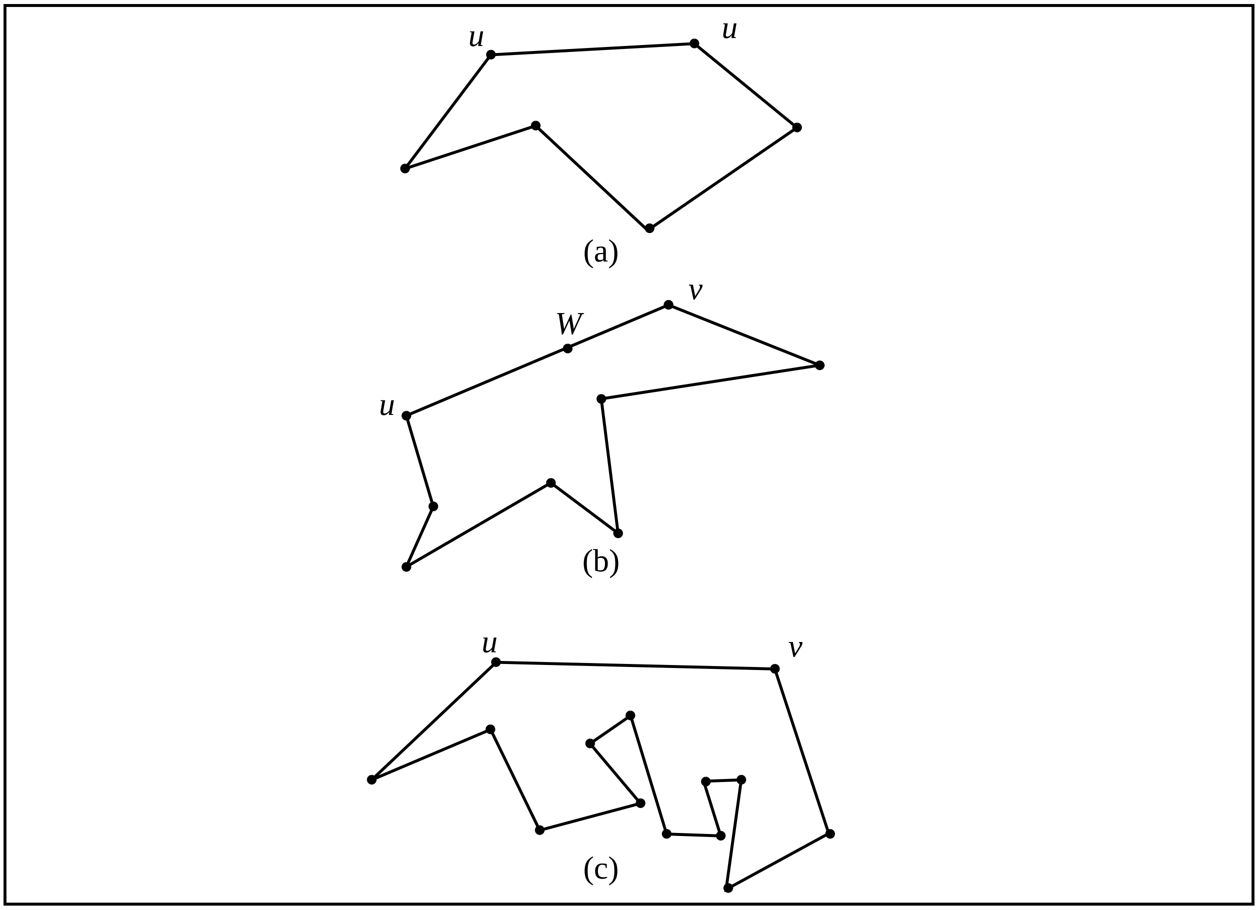

Fig. 1. (a) Complete visibility. (b) Strong visibility. (c) Weak visibility.

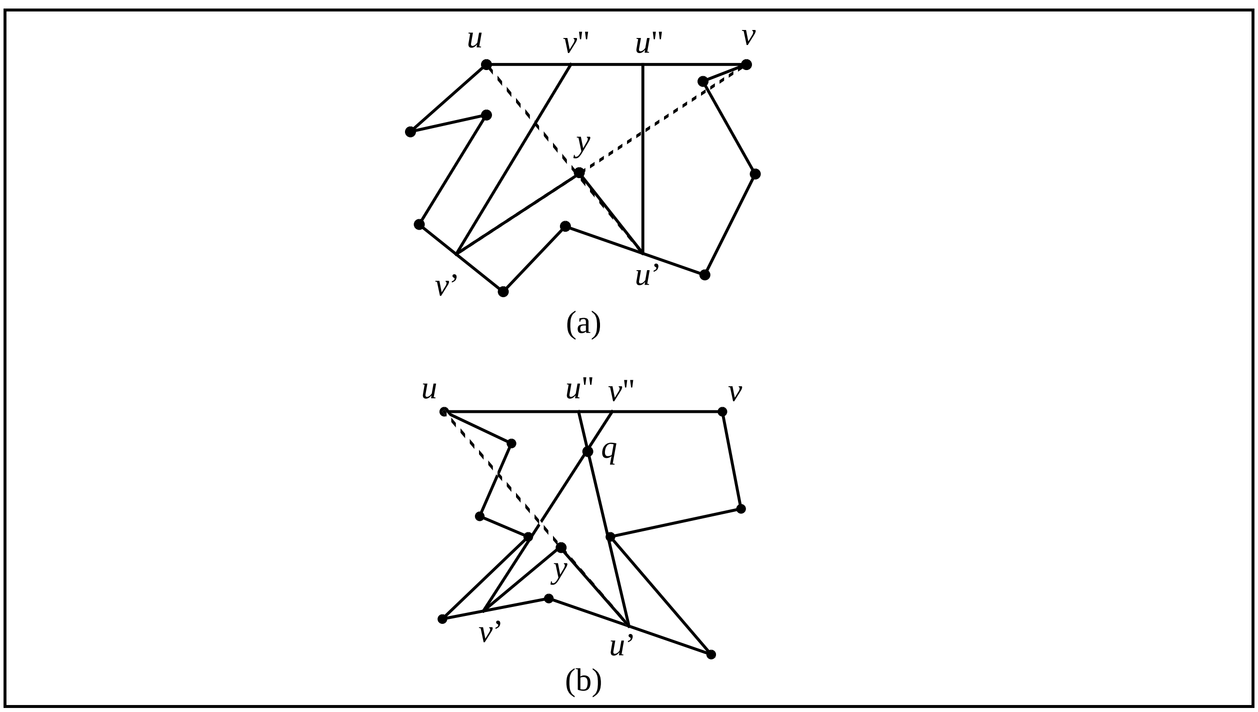

Fig. 2.

An Optimal Algorithm for Determining the Visibility of a Polygon from an Edge

David Avis
and
Godfried Toussaint,

ABSTRACT

In many computer applications areas such as graphics, automated cartography, image processing, and robotics the notion of visibility among objects modeled as polygons is a recurring theme. This paper is concerned with the visibility of a simple polygon from one of its edges. Three natural definitions of the visibility of a polygon from an edge are presented. The following computational problem is considered. Given an n-sided simple polygon, is the polygon visible from a specified edge? An O(n), and thus optimal, algorithm is exhibited for determining edge visibility under any of the three definitions. The paper closes with an interesting characterization of visibility and some open problems in this area.

Index Terms - Algorithms, computational complexity, computational geometry, computer graphics, hidden line problems, image processing, robotics, simple polygon, visibility.

1. Introduction

The notion of visibility in geometric objects is one that appears in many applications: the hidden line problem of graphics [1], in image processing [2], surveillance, and control of robots [3]. Several papers [2], [4], [5], [11], [12] have appeared concerning the problem of visibility in a polygonal region from a fixed point. In this paper we discussed what might be termed the “jail-house” problem, i.e., the problem of polygonal visibility from an edge. It is convenient to imagine a guard or robot patrolling a portion of the boundary of a polygonal region. It is natural to ask under what circumstances the entire region can be observed. In this paper we introduce three natural definitions of visibility from an edge of a polygon. Our main result is a linear algorithm for determining whether or not a given polygon is visible, under any of the definitions, from a given edge.

Manuscript received March 25, 1980; revised January 7, 1981. This work was supported by the N.S.E.R.C. under Grants A3013 and A9293.
The authors are with the School of Computer Science, McGill University, Montreal, P.Q., Canada.