Extending Visibility Polygons by Mirrors to Cover Specific Targets

Arash Vaezi* Mohammad Ghodsi†

Abstract

The visibility polygon VP of a point q ($VP(q)$) inside a simple polygon P with n vertices, can be computed in linear time. We propose a linear time algorithm to extend VP, by converting some edges of P to mirrors, so that a given segment d can also be seen from the viewer. In linear time our algorithm finds every edge such that, when converted to a mirror, makes d visible to our viewer.

1 Introduction

Many variations of the visibility polygon have been studied so far. In general, we have a simple polygon P with n vertices, and a viewer point q inside P. The goal of the visibility problem is to find the maximal sub-polygon of P ($VP(q)$) visible to the viewer. There are linear time algorithms to compute $VP(q)$ ([6], [3]).

It was shown in 2010 that VP of a given point or segment can be computed in presence of a mirror in $O(n)$ [5]. Also, it was shown in the same paper that the union of two visibility polygons can be computed in $O(n)$.

We consider the problem of finding all edges any of which when converted to a mirror (and thus called *mirror-edge*) can make a specific segment visible (also called *mirror-visible*) to a given point. We propose a linear time algorithm for this problem.

This paper is organized as follows: In Section 2, notations are described. Next in Section 3, we present a linear time algorithm to solve the above problem. Section 4 contains some discussions and future works.

2 Notations

Suppose P is a simple polygon and $int(P)$ denote its interior. Two points x and y are visible to each other, if and only if the open line segment $\overline{xy}$ lies completely in $int(P)$. The visibility polygon of a point q in P, denoted as $VP(q)$, consists of all points of P visible to q. Edges of $VP(q)$ that are not edges of P are called *windows*. Weak visibility polygon of a segment d, denoted as $WVP(d)$, is the maximal sub-polygon of P visible to at least one point (not the endpoints) of d. The visibility of an edge v_iv_{i+1} of a simple polygon P can be viewed in different ways [1]: P is said to be *completely visible* from v_iv_{i+1} if every point $z \in P$ and for any point $w \in v_iv_{i+1}$, w and z are visible. All these different visibilities can be computed in linear time (see [4] for the weakly visibility polygon and [1] for the strongly).

Suppose an edge e of P is a mirror. Two points x and y are e-mirror-visible, if and only if they are directly visible with one specular reflection. $VP(q)$ with a mirror-edge e, is the maximal sub-polygon of P visible to q either directly or via e.

Two points or segments are *mirror-visible* if and only if they cannot see each other directly, but can with an edge middling as a mirror. We consider the whole edge as a mirror, thus two points can be mirror-visible by just a part of an edge. Also, if a point can see a part of a segment through a mirror, we call them mirror-visible.

*Department of Computer Engineering, Sharif University , Technology, avaezi@ce.sharif.ir

†Department of Computer Engineering, Sharif University of Technology, and Institute for Research in Fundamental Sciences (IPM), Tehran, Iran. ghodsi@sharif.edu. This author's research was partially supported by the IPM under grant No: CS1389-2-01

3 Expanding a point visibility polygon

3.1 Recognizing all mirror-edges

We intend to find all edges of P, any of which when converted to a mirror causes a given point q see a segment d.

Theorem 1 *Suppose P is a simple polygon with the complexity of n, q is a given point inside P, and d is a given segment which is not directly visible by q. All edges any of which can makes d mirror-visible to q can be found in $O(n)$ time.*

Remark. We will prove this theorem using a given diagonal of P, instead of the given segment. We will use the two endpoints of the diagonal. Since the assertion that the segment is actually a diagonal in not used in the proof, the stated proof holds for any segment inside P. Instead of the endpoints of the diagonal, we can use one endpoint of the closet edge of P to the given segment. Let at least one endpoint of this edge be upon the given segment inside the polygon.

Proof. First we prove that with an $O(n)$ time of preprocessing, we can answer any query of whether a particular edge of P can make d mirror-visible to q in $O(1)$ time. The processing time is for computing

$VP(q)$ and $WVP(d)$, and finding some reflex vertices which may block the mirror-visibility area.

Obviously, any mirror-edge that makes d visible to q should lie in the intersection of $VP(q)$ and $WVP(d)$ which can be computed in linear time. If goal is the visibility of the whole segment, we should compute the complete visibility polygon of the given segment instead of the weakly visibility polygon of which.

Suppose that e is intersected by both visibility polygon from $v_1(e)$ to $v_2(e)$ in the order of P's vertices. We assume that this part of e is mirror. We will find out whether any part of d is e-mirror-visible. Let L_1 and L_2 be two half-lines from the ray-reflection of q at $v_1(e)$ and $v_2(e)$ respectively (see Figure 1).

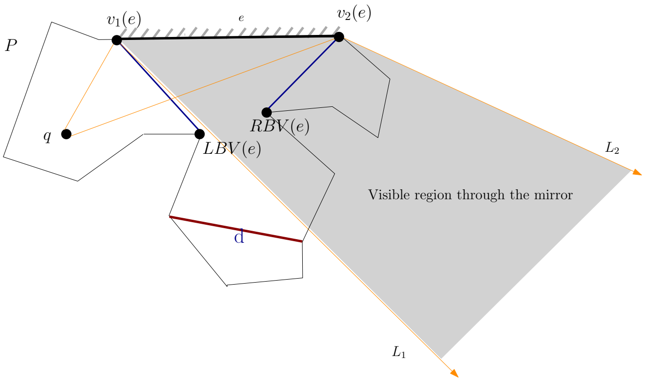

Figure 1: The region between L_1 and L_2 is the visible area by q through the mirror e.

If d is in the region between L_1 and L_2 and no part of P obstructs d, then d is e-mirror-visible (see Figure 1). Since P is simple, any obstruction has to contain a reflex vertex. Considering that P's vertices are ordered in clockwise direction, we define $LBV(e)$ (for Left Blocking Vertex of e) to be the reflex vertex before $v_1(e)$ that can obstruct most the e-mirror-visibility of d, and similarly $RBV(e)$ (for Right-Blocking Vertex of e) to be the reflex vertex after $v_2(e)$. Different mirror-edges may have the same LBVs or RBVs, but LBV and RBV for any edge e are unique and may be $v_1(e)$ or $v_2(e)$. It is sufficient to check only the corresponding LBV and RBV vertices not to block the mirror visibility area. We will show that we can find all LBVs and RBVs for all mirror-edges in linear time.

Following cases should be considered:

1. If L_1 and L_2 both lie on one side of d, d is not in the mirror-visible area. q cannot see d through this mirror-edge.

2. If L_1 and L_2 do not lie on one side of d and if d is in the middle of the mirror-visible area, q can see d through the mirror-edge. Because e is visible to d, and the visibility area from L_1 to L_2 is a continuous region.

3. Otherwise L_1 or L_2 crosses d, and we should check whether any part of P, obstructs *the whole* visible area through the mirror-edge (In the case of the completely visibility polygon of d, it is sufficient to check L_1 and L_2 not to cross d, except in its endpoints.

   Now, we should check the polygon not to block the rays from the right or form the left side of the mirror-edge. Consider the two segments $s_1 = \overline{LBV v_1(e)}$ also $s_2 = \overline{RBV v_2(e)}$. If L_2 crosses s_1, or if L_1 crosses s_2, consequently q and d are not e-mirror-visible.

Obviously, collision checking of a constant number of points is done in $O(1)$ for any candidate edge in the intersection polygon of $VP(q)$ and $WVP(d)$, which in addition to the processing time leads to an $O(n)$ algorithm to find all feasible mirror-edges. □

3.1.1 Computing LBV and RBV vertices

First we consider the computation of the LBV vertices. Starting from d_1 (the left endpoint of the given diameter) and similar to Graham's algorithm [2] in finding the convex hull of points, we trace $WVP(d)$ using the reflex vertices of P in the P's order of vertices. We act as the following:

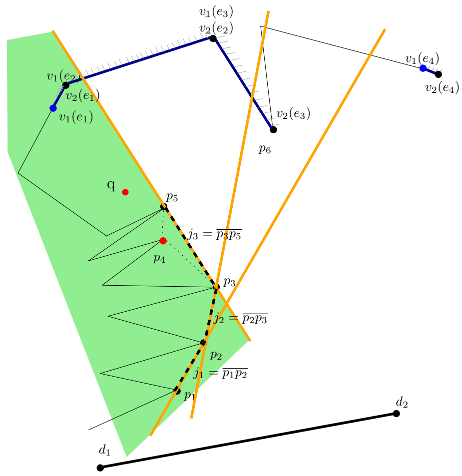

Figure 2: Constructing the convex hull for distinguishing LBV vertices for all the mirror-edges. p_1, p_2, p_3 and p_5 are the vertices of the convex hull. Four mirrors are shown, for example p_5 is $LBV(e_1)$.

Suppose the reflex vertices from d_1 to d_2, are $p_1, p_2, ..., p_k$, where $k \in O(n)$. We start making the convex hull of the reflex vertices, whose concave region is directed to the outside of the polygon (see Figure 2). When we reach a new mirror-edge, we use the lines containing the edges of the updated convex shape till that moment.

Assume $v_2(e_i)$ represents the second endpoint of the ith candidate mirror-edge. When we reach the ith mirror-edge we compare $v_2(e_i)$ with the *chosen line* for the $(i-1)th$ mirror-edge (the last visited mirror-edge). If $v_2(e_i)$ lies on–or on the left side of–that line, then the ith and $(i-1)$th mirror-candidates have the same reflex vertex as their LBV. Otherwise, we should compare $v_2(e_i)$ with the line which is not checked yet, and contains the most recent constructed segment of the convex shape.

For example, suppose the convex shape has 3 segments (j_1, j_2, j_3) before we reach the first mirror-edge (e_1) with $v_2(e_1)$. We should check $v_2(e_1)$ with the line, which contains the last constructed segment of the convex hull (j_3), to see if it has $v_2(e_1)$ on its left. If $v_2(e_1)$ lies on the right side of that line, we check $v_2(e_1)$ with j_2. We can continue this way, if there is no more line, there is no $LBV(e_1)$. Assume we select the line containing j_3 for e_1. We should check this line for e_2 too, because they may have the same LBV.

At the end, suppose for a mirror-edge e we chose line L, whose interior contains more than one vertex of the convex shape. If $v_2(e)$ is on the left side of L then we should choose the most recent joined vertex of the convex shape on L. But, if $v_2(e)$ itself lies on the L, then we choose the first reflex vertex which joined the convex shape on L. This reflex vertex is the *closet* one on L to d.

However, While we trace $WVP(d)$, facing with any new reflex vertex we should update the convex shape (see Figure 3).

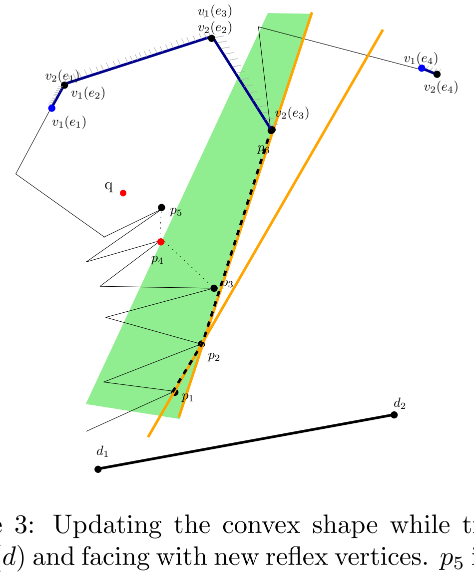

Figure 3: Updating the convex shape while tracing $WVP(d)$ and facing with new reflex vertices. p_5 is chosen as $LBV(e_1)$, p_3 and p_2 as $LBV(e_2)$ and $LBV(e_3)$, respectively. If we had a reflex vertex p_0 for the fourth mirror-edge, first we may have selected p_1. But later we should change it to $v_1(e_4)$, because p_1 cannot block the e_4-mirror-visibility.

We trace the $WVP(d)$ in counter-clockwise direction starting from d_2 to find all RBV vertices similarly. At the end, since there may be some false chosen LBV or RBV vertices, we should trace $WVP(d)$ again in both directions. First we compare all LBV vertices of the mirror-edges with the corresponding segment $\overline{v_2(e_i), d_1}$. If it was in the left side of the segment, then the chosen vertex is not obstructing the mirror-visibility area, hence, we change the chosen LBV to $v_1(e_i)$ for the ith mirror-edge. We proceed similarly for RBV vertices in the other direction.

3.1.2 Proof of correctness

First, the following lemma:

Lemma 2 *For each mirror-edge e the reflex vertex as its LBV, is among the reflex vertices before e in the P's order of vertices (p_i $0 \leq i \leq k$). And it is the closet one to d (p_j) which the corresponding segment $\overline{p_j v_2(e)}$ holds all the other p_i ($i \neq j$ $0 \leq i \leq k$) reflex vertices on its left side (see Figure 4).*

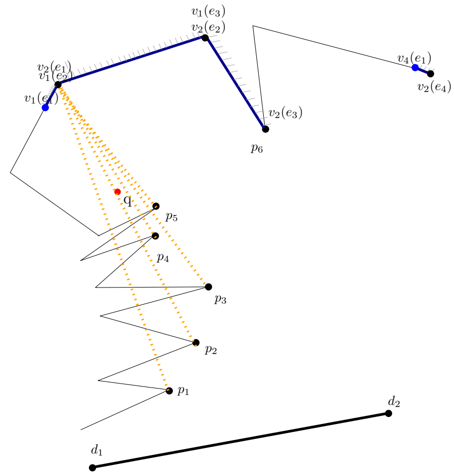

Figure 4: From Lemma 2 point p_5 must be the best choice for $LBV(e_1)$.

Proof. Suppose e is our mirror-edge and p_j is the chosen $LBV(e)$ using Lemma 2. We will show neither a farther reflex vertex nor a closer one to d is a better choice for being $LBV(e)$. Actually, we will show there are examples of violations for any other reflex vertices, either farther or closer to d.

Suppose there is a reflex vertex p_l, closer than p_j to d. Also assume L_2 crosses the polygon on the left of p_l but not on the left side of p_j. Therefore, p_l obstructs the e-mirror-visible area so that the viewer cannot see d through e. This leaves p_j, chosen from Lemma 2, not to be $LBV(e)$. We know that L_2 should place on the right side of $\overline{d_1 v_2(e)}$, because otherwise the whole mirror-visibility region is on the left side of d. But,

p_j is chosen by Lemma 2, hence, it should lie on the right side of $\overline{p_l v_2(e)}$ (see Figure 5). Thus, L_2 cannot cross the polygon on the left side of p_l but not on the left side of p_j. In the stated analysis if the reflex vertices such as p_j or p_l lie *on* L_2, they can block the mirror visibility area. Therefore, we treat them as if they were on the right side of L_2.

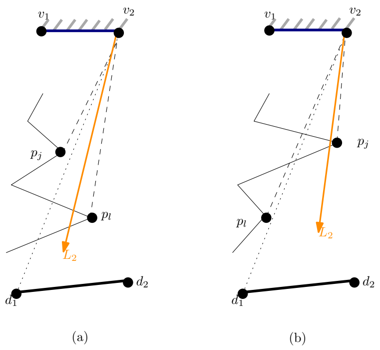

Figure 5: No lower reflex vertex can be a better choice than the one chosen by Lemma 2

Similarly, assume a reflex vertex p_h exists, which is farther than p_j to d. And L_2 crosses the segment $\overline{p_h v_2(e)}$ and not $\overline{p_j v_2(e)}$. From Lemma 2 p_h should be at the left side of $\overline{p_j v_2(e)}$, and we know that L_2 has intersection with d on the right side of $\overline{d_1 v_2(e)}$. Consequently, L_2 crosses d on the right side of p_j, and crosses $\overline{p_h v_2(e)}$ while it lies on–or on the left side of– p_j. Therefore, it should cross $\overline{p_j v_2(e)}$, and we are done. □

Thus, using Lemma 2, when there are $O(n)$ reflex vertices LBVs for all the mirror-edges can be computed within $O(n^2)$ time complexity.

Now consider two mirror-edges e_1 and e_2 and two reflex vertices p_i and $p_{(i-1)}$, which the segment $\overline{p_i v_2(e_1)}$ has all other reflex vertices on its left side for e_1. Also, the segment $\overline{p_{(i-1)} v_2(e_2)}$ acts the same for e_2. Hence, p_i and $p_{(i-1)}$ are $LBV(e_1)$ and $LBV(e_2)$, respectively. The segment $\overline{p_i p_{(i-1)}}$ is between those lines. Therefore, for all the mirror-edges (e) which have their v_2 endpoints on the left side of $\overline{p_i p_{(i-1)}}$ the segment $\overline{p_i v_2(e)}$ is covering all the other reflex vertices on its left. The segments on the right side of $\overline{p_i p_{(i-1)}}$ should be considered one by one. s$\overline{p_{(i-1)} v_2(e)}$ can act the same for those mirror-edges whose v_2 vertices lie on the right side of $\overline{p_i p_{(i-1)}}$.

The lines containing the $\overline{p_i p_{(i-1)}}$ segments $(1 \leq i \leq k)$ evidently satisfy the property of being convex hull, the direction of the concave region of which is to the outside of the polygon.

As we mentioned before, each $LBV(e)$ vertex should lie on the right side of $\overline{d_1 v_2(e)}$ of e, otherwise, we should exchange the chosen $LBV(e)$ with $v_1(e)$. Therefore, at the end we should check all of LBV vertices. The reason is, in these cases there is nothing to obstruct the visibility region from the left side of the mirror-edge. Clearly, all the stated analysis holds in computing RBV vertices, too.

4 Discussion

We dealt with the problem of extending the visibility polygon of a given point in a simple polygon, so that another segment becomes visible to it. For this purpose we convert some of the polygon edges to mirrors. The problem is to find all such kind of edges. Using the algorithm we proposed, it is possible in linear time corresponding the complexity of the simple polygon. We only discussed finding the edges to be mirrors, but it is shown that having two mirrors, the resulting visibility polygon, may not be a simple polygon [6]. The problem can be extended as; put mirrors inside the polygon, a point with a limited visibility area and so on.

References

[1] D. Avis, G. T. Toussaint. An optional algorithm for determining the visibility of a polygon from an edge. *IEEE Transactions on Computers*, C-30: 910-1014, 1981.

[2] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf Computational Geometry Algorithms and Applications. Springer, third edition *Department of Computer Science Utrecht University*, 13,14 2008.

[3] H. ElGindy, D. Avis. A Linear algorithm for computing the visibility polygon from a point. *Journal of Algorithms*, 2: 209-233, 1987.

[4] L. J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2: 209–233, 1987.

[5] B. Kouhestani, M. Asgaripour, S. S. Mahdavi, A. Nouri and A. Mohades. Visibility Polygons in the Presence of a Mirror Edge. *In Proc. 26th European Workshop on Computational Geometry*, 26: 209–212, 2010.

[6] D. T. Lee. Visibility of a simple polygon. *Computer Vision, Graphics, and Image Processing*, 22: 207–221, 1983.