How to Extend Visibility Polygons by Mirrors to Cover Invisible Segments

Arash Vaezi⋆ and Mohammad Ghodsi⋆⋆

Sharif University of Technology

Abstract. Given a simple polygon $\mathcal{P}$ with n vertices, the visibility polygon (VP) of a point q ($VP(q)$), or a segment $\overline{pq}$ ($VP(\overline{pq})$) inside $\mathcal{P}$ can be computed in linear time. We propose a linear time algorithm to extend VP of a viewer (point or segment), by converting some edges of $\mathcal{P}$ into mirrors, such that a given non-visible segment $\overline{uw}$ can also be seen from the viewer. Various definitions for the visibility of a segment, such as weak, strong, or complete visibility are considered. Our algorithm finds every edge such that, when converted to a mirror, makes $\overline{uw}$ visible to our viewer. We find out exactly which interval of $\overline{uw}$ becomes visible, by every edge middling as mirror, all in linear time.

1 Introduction

Many variations of visibility polygons have been studied so far. In general, we have a simple polygon $\mathcal{P}$ with n vertices, and a viewer which is a point (q), or a segment ($\overline{pq}$) inside $\mathcal{P}$. The goal is to find the maximal sub-polygon of $\mathcal{P}$ visible to the viewer ($VP(q)$ or $VP(\overline{pq})$). There are linear time algorithms to compute $VP(q)$ ([7]) or when the viewer is a segment [5].

It was shown in 2010 that VP of a given point or segment can be computed in presence of *one* mirror-edge in $O(n)$ [6]. Also, it was shown in the same paper that the union of two visibility polygons can be computed in $O(n)$.

We consider different problems of finding every edge e such that when converted to a mirror (and thus called *mirror-edge*) can make at least a part of a specific invisible segment visible (also called *e-mirror-visible*) to a given point or segment. We propose linear time algorithms for these problems. Considering a segment as a viewer, we deal with all different definitions of visibility, namely, weak, complete and strong visibility, which was introduced by [3]. Also, we can easily find mirror-visibile intervals of the invisible segment ($\overline{uw}$) considering all edges as mirrors in linear time corresponding to the complexity of $\mathcal{P}$.

This paper is organized as follows: In Section 2, notations are described. Next in Section 3, we present a linear time algorithm to recognize every mirror-edge

⋆ Department of Computer Engineering, Sharif University of Technology, avaezi@ce.sharif.edu

⋆⋆ Department of Computer Engineering, Sharif University of Technology, and Institute for Research in Fundamental Sciences (IPM), Tehran, Iran. This author's research was partially supported by the IPM under grant No: CS1392-2-01, ghodsi@sharif.edu

e of $\mathcal{P}$ that makes a given segment $\overline{uw}$ e-mirror-visible to q. In Section 4 we will show that e-mirror-visible interval of $\overline{uw}$ to q can be computed in constant time. In Section 5, we deal with a given segment instead of a point. And finally, Section 6 contains some discussions and future works.

2 Notations and assumptions

Suppose $\mathcal{P}$ is a simple polygon and $int(\mathcal{P})$ denotes 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 $int(\mathcal{P})$. The visibility polygon of a point q in $\mathcal{P}$, denoted as $VP(q)$, consists of all points of $\mathcal{P}$ visible to q. Edges of $VP(q)$ that are not edges of $\mathcal{P}$ are called *windows*. The weak visibility polygon of a segment $\overline{pq}$, denoted as $WVP(\overline{pq})$, is the maximal sub-polygon of $\mathcal{P}$ visible to at least one point (not the endpoints) of $\overline{pq}$. The visibility of an edge $e = (v_i, v_{i+1})$ of $\mathcal{P}$ can be viewed in different ways [3]: $\mathcal{P}$ is said to be *completely visible* from e if for every point $z \in \mathcal{P}$ and for any point $w \in e$, w and z are visible (denoted as CVP short from completely visible polygon). Also, $\mathcal{P}$ is said to be *strongly visible* from e if there exists a point $w \in e$ such that for every point $z \in \mathcal{P}$, w and z are visible (SVP). These different visibilities can be computed in linear time (see [5] for WVP and [3] for CVP and SVP).

Suppose an edge e of $\mathcal{P}$ is a mirror. Two points x and y are *e-mirror-visible*, if and only if they are directly visible with one specular reflection through a mirror-edge e. Specular reflection is the mirror-like reflection of light from a surface, in which light from a single incoming direction is reflected into a single outgoing direction. The direction in which light is reflected is defined by the law-of-reflection, which states that the incident, surface-normal and reflected directions are coplanar.

Since only an interval of a mirror-edge is useful, we can consider the whole edge as a mirror, and there is no need to split an edge.

We assume that n vertices of $\mathcal{P}$ are ordered in clockwise order (CWO).

3 Expanding point visibility polygon

We intend to find every mirror-edge e of $\mathcal{P}$ that causes a given point q see any interval of a given segment $\overline{uw}$ inside $\mathcal{P}$. We will find the exact interval of $\overline{uw}$ which is e-mirror-visible to q for every mirror-edge e of $\mathcal{P}$ in the next section.

3.1 Overview of the algorithm

Obviously, any potential mirror-edge e that makes $\overline{uw}$ visible to q should lie on $VP(q) \cap WVP(\overline{uw})$ which can be computed in linear time. If the goal is to check e-mirror-visibility of the whole $\overline{uw}$, we should instead compute the complete visibility polygon of $\overline{uw}$ (i.e. $VP(q) \cap CVP(\overline{uw})$).

Suppose that e is intersected by $VP(q) \cap WVP(\overline{uw})$ from $v_1(e)$ to $v_2(e)$ in CWO. We use this part of e as mirror. We will find out whether any part of $\overline{uw}$

is e-mirror-visible. Let $L_1(e)$ and $L_2(e)$ be two half-lines from the ray-reflection of q at $v_1(e)$ and $v_2(e)$ respectively. Also let $q'(e_i)$ be the image of q considering e_i from $v_1(e_i)$ to $v_2(e_i)$ (e_i is the ith potential mirror-edge in CWO).

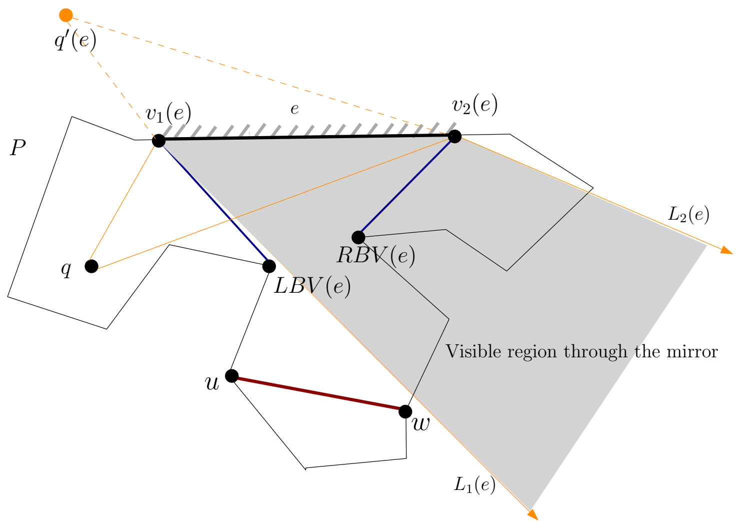

Fig. 1. The region between $L_1(e)$ and $L_2(e)$ is the visible area by q through e being a mirror from $v_1(e)$ to $v_2(e)$.

If $\overline{uw}$ intersects the region between $L_1(e)$ and $L_2(e)$ and no part of $\mathcal{P}$ obstructs $\overline{uw}$, then $\overline{uw}$ is e-mirror-visible (see Figure 1). Since $\mathcal{P}$ is simple, e-mirror-visibility can only be obstructed by reflex vertices.

For each mirror-edge e, we define $LBV(e)$ (for Left Blocking Vertex of e) and $RBV(e)$ (for Right Blocking Vertex of e) as below. In Subsection 3.2.2, we will prove that no other reflex vertex can block e-mirror-visibility area except for these two reflex vertices.

3.2 LBVs and RBVs

Definition 1. *Assume that $p_1, p_2, \ldots, p_k$ are the reflex vertices we meet when tracing $WVP(\overline{uw})$ starting from u in CWO before we reach a mirror-edge e. We define $LBV(e)$ to be that vertex p_j such that if $\overline{p_j q'(e)}$ (i.e. from p_j to $q'(e)$) holds all other p_i ($i \neq j$ $1 \leq i \leq k$) reflex vertices on its left side. In another word, if we move from p_j to $q'(e)$ all other p_i reflex vertices are on our left side (see Figure 6 in Appendix). If more than one vertex has this property, we choose the one with the lowest index. If no such vertex exits, we set $v_1(e)$ as $LBV(e)$. $RBV(e)$ is defined similarly when we trace $WVP(\overline{uw})$ from w in counter-clockwise order (CCWO).*

Different mirror-edges may have the same LBVs or RBVs. And, obviously through Definition 1, for each mirror-edge e, $LBV(e)$ and $RBV(e)$ is unique.

Algorithm 1 (to check whether q can see any interval of $\overline{uw}$ through mirror-edge e).
Assuming that e, from $v_1(e)$ to $v_2(e)$ is in $VP(q) \cap WVP(\overline{uw})$ and $L_1(e)$ and $L_2(e)$ are as defined above, the following cases are considered:

1. If $L_1(e)$ and $L_2(e)$ both lie in one side of $\overline{uw}$, then $\overline{uw}$ is not in the e-mirror-visible area. That is, q cannot see $\overline{uw}$ through e.
2. Otherwise, if $\overline{uw}$ is between $L_1(e)$ and $L_2(e)$. I.e., it is in the middle of the mirror-visible area, q can see $\overline{uw}$ through the mirror-edge e. Because e is visible to $\overline{uw}$, and the visibility area from $L_1(e)$ to $L_2(e)$ is a continuous region.
3. Otherwise, $L_1(e)$ or $L_2(e)$ crosses $\overline{uw}$. In this case, we check whether any part of $\mathcal{P}$, obstructs *the whole* visible area through e (In case of $CVP(\overline{uw})$, it is sufficient to check $L_1(e)$ and $L_2(e)$ not to cross $\overline{uw}$, except in its endpoints.) For this, it is checked whether $\mathcal{P}$ blocks the rays from the right or left side of e. If $LBV(e)$ lies on the left side of $L_2(e)$, and $RBV(e)$ lies on the right side of $L_1(e)$, then q can see $\overline{uw}$ through e.
   Otherwise, q and $\overline{uw}$ are not e-mirror-visible.

Obviously, collision checking of a constant number of points and lines can be done in $O(1)$ for any mirror-edge.

Computing LBV and RBV vertices

Algorithm 2. First consider the computation of LBV vertices. We already know that the potential mirror-edges lie in $VP(q) \cap WVP(\overline{uw})$. To make an easy understanding, these edges are numbered in CWO as $e_1, e_2, \ldots$.

Considering $WVP(\overline{uw})$ we construct a new polygon by adding $\overline{q'(e_i)v_1(e_i)}$ and $\overline{q'(e_i)v_2(e_i)}$ to each mirror-edge e_i, and eliminating $\overline{v_1(e_i)v_2(e_i)}$ interval from e_i.

We call this polygon $TP(\overline{uw})$ (Tracing Polygon). Obviously, $TP(\overline{uw})$ may not be a simple polygon, and has $O(n)$ vertices corresponding to the complexity of $\mathcal{P}$ (see Figure 7 in Appendix).

Starting from u (the left endpoint of $\overline{uw}$) we trace $TP(\overline{uw})$ in clockwise order. While doing so, we construct a convex shape on the reflex vertices of $\mathcal{P}$ we visit, using an algorithm similar to Graham's scan [4] in $\mathcal{P}$'s order of vertices. We consider u as one reflex vertex.

As we meet a new reflex vertex, we push the line containing the new constructed edge of the convex shape into a stack named S and update the stack as move forward. When $q'(e_i)$ of a mirror-edge e_i is reached in our trace, $q'(e_i)$ is compared with the line on the top of the stack called ℓ. If $q'(e_i)$ lies on the right side of ℓ, ℓ is popped from S. Otherwise, if $q'(e_i)$ lies on, or on the left side of ℓ, then we assign ℓ as the *chosen line* for e_i, denoted as $cl(i)$. ℓ (Top(S)) is then check with $q'(e_{i+1})$, $q'(e_{i+2})$, $\ldots$, to become their possible

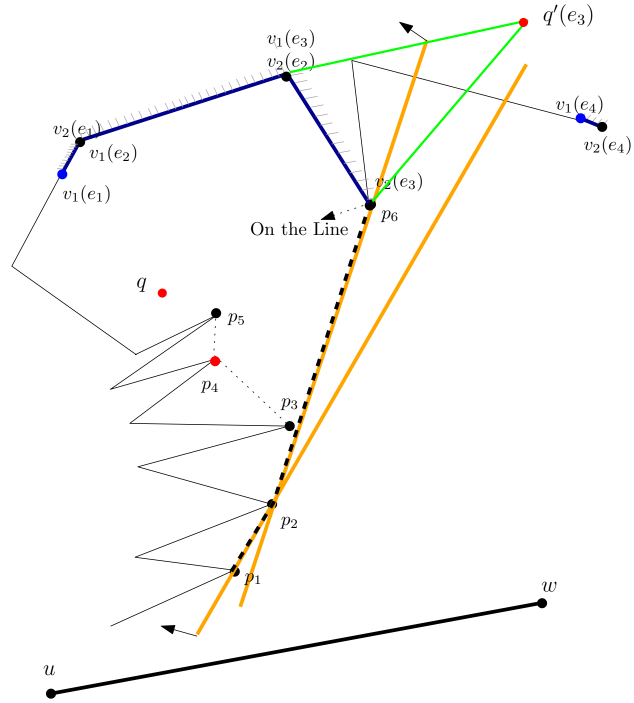

Fig. 2. Updating the convex shape while tracing $TP(\overline{uw})$ and facing with new reflex vertices. p_5 is chosen as $LBV(e_1)$, p_3 and p_2 as $LBV(e_2)$ and $LBV(e_3)$, respectively. If we consider u for the fourth mirror-edge, first we select p_1. But, later we should change $LBV(e_4)$ to be $v_1(e_4)$, because p_1 cannot block the e_4-mirror-visibility.

chosen lines, or popped up. If the stack is empty when we visit $q'(e_i)$, we assign $LBV(e_i) = v_1(e_i)$.

Obviously, when a new reflex vertex is met, the convex shape and the stack is updated accordingly (see Figure 2) and the algorithm continues.

See Figure 3, for an example. Here, the stack contains 3 lines (ℓ_1, ℓ_2, ℓ_3, the last on top) when we reach $v_1(e_1)$. We check $q'(e_1)$ with ℓ_3, which is on Top(S), to see if it has $q'(e_1)$ on its left. If $q'(e_1)$ lies on the right, then $q'(e_1)$ is checked with ℓ_2. Here, $cl(1) = \ell_3$.

At the end, for each mirror-edge e_i, we consider the two reflex vertices of $cl(i)$, say re_1 and re_2 (in $CCWO$). If $q'(e_i)$ lies on the left of $cl(i)$, then $LBV(e_i) = re_2$. Otherwise, it lies on $cl(i)$, then $LBV(e_i) = re_1$. If there are more than two reflex vertices consider the last on $cl(i)$ (re_1).

The RBV vertices are computed similarly by tracing $TP(\overline{uw})$ in counter-clockwise direction starting from w.

At the end, since there may be some false vertices chosen as LBV or RBV vertices, we will trace $WVP(\overline{uw})$ in both directions to correct these cases. First

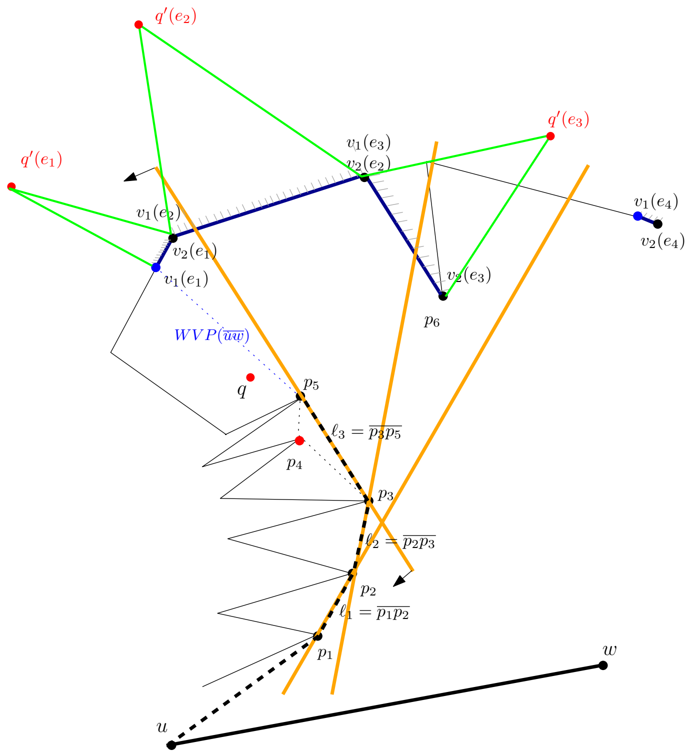

Fig. 3. Constructing the convex hull to distinguish LBV vertices for all mirror-edges. p_1, p_2, p_3 and p_5 are the reflex vertices that are used in the convex hull construction. Four mirror-edges e_1 to e_4 are shown. In this figure, p_5 is $LBV(e_1)$.

each $LBV(e_i)$ chosen by previous algorithm is compared with the segment $d = \overline{v_2(e_i)u}$. If $LBV(e_i)$ lies on the left side of d, or if $LBV(e_i) = u$, then $LBV(e_i)$ was falsely chosen since it is not obstructing the mirror-visibility area. The correction is made in this case by setting $LBV(e_i) = v_1(e_i)$. We proceed similarly for RBV vertices in the other direction.

Obviously, all these operations can be performed in $O(n)$ time. For more justification, do not consider the stack and see what happens to the lines (see [1] for more details).

Proof of correctness and analysis of the algorithm In this Subsection we present the proof and the analysis of the algorithm.

Theorem 1. *Suppose $\mathcal{P}$ is a simple polygon with n vertices, q is a given point inside $\mathcal{P}$, and $\overline{uw}$ is a given segment which is not directly visible by q. Every edge e that makes $\overline{uw}$ e-mirror-visible to q can be found in $O(n)$ time.*

Remark1. We will prove this theorem assuming that $\overline{uw}$ is a diagonal of $\mathcal{P}$. Since the assertion that $\overline{uw}$ is actually a diagonal is not used in the proof, the stated proof holds for any segment inside $\mathcal{P}$. To start tracing $TP(\overline{uw})$, instead of the endpoints of the diagonal, we can use one endpoint of the closest edge of $\mathcal{P}$ to the given segment. Let at least one endpoint of this edge be upon the given segment inside the polygon.

Remark2. Note that the algorithm covers some situations where $\overline{uw}$ does not have their endpoints on the boundary of $\mathcal{P}$. In these cases there might be some mirror-edge e which can see $\overline{uw}$ from its behind. In another word, e may see a part of the invisible target segment from w to u, and w is on the left side of the e-mirror-visible interval when we are standing on $\overline{uw}$ and facing to e (see Figure 9 in Appendix). And, we need to swap the position of u and w and run the above-mentioned algorithms one more time to see if these kind of mirror-edges exist that may make an interval of the target mirror-visible to q. So, we need to use $\overline{wu}$ instead of $\overline{uw}$. And, we need to run all above-mentioned algorithms one more time using $\overline{wu}$, which takes an additional $O(n)$ time complexity. Note that these two runs do not have any conflict with each other, and they find absolutely independent mirror-edges. This is because, a mirror-edge e which sees $\overline{uw}$ from behind will be eliminated in the first run. And this is because, in the first run, using $\overline{uw}$, w is placed on the left side of $L_1(e)$, and e will be eliminated through case 1 of Algorithm 1. Without lost of generality, for simplicity we assume that no mirror-edge can see $\overline{uw}$ from behind.

Proof.

1. The algorithm correctly computes all LBV's and RBV's in $O(n)$. This is clear from Definition 1 and Algorithm 2. This algorithm constructs two convex hulls.
2. Algorithm 1 correctly checks whether each mirror-edge e can make at least a part of the given segment $\overline{uw}$ e-mirror-visible to q. For this, we only need to prove that the algorithm is correct if case 3 occurs. Other cases are obvious. That is, if $L_1(e)$ or $L_2(e)$ or both cross $\overline{uw}$, and if $LBV(e) = p_j$ does not cross $L_2(e)$ where we decide that $\overline{uw}$ is e-mirror-visible from q, then no other reflex vertices can completely obstruct the e-mirror-visible area. Suppose on the contrary, that another vertex p_l completely obstructs the visible area while p_j does not. In this case, $\overline{q'(e)p_l}$ is on the right side of $L_2(e)$ and thus is on the right side of $\overline{q'(e)p_j}$ which contradicts p_j being $LBV(e)$. Similar arguments hold for RBV. We can also prove that no other reflex vertices (other than the left and right chains that appear when we trace the $WVP(\overline{uw})$) can obstruct the visibility.

4 Specifying the visible part of $\overline{uw}$

In this section we present an algorithm to determine the visible interval of the given segment ($\overline{uw}$) which is e-mirror-visible by middling of a given mirror-edge (e).

Lemma 1. *We have a simple polygon $\mathcal{P}$, a point q as a viewer, and a segment $\overline{uw}$, inside $\mathcal{P}$. In linear time corresponding to the complexity of $\mathcal{P}$, for every mirror-edge e, we can compute the exact interval of $\overline{uw}$ that is e-mirror-visible.*

Proof. We will show for a specified mirror-edge e, while we have $LBV(e)$, we can find e-mirror-visible part of $\overline{uw}$ in constant time. Therefore, it takes $O(n)$ time to distinguish the visible intervals of $\overline{uw}$, for every mirror-edge.

Consider a mirror-edge e, without loss of generality suppose we know $\overline{uw}$ is e-mirror-visible. We can find the visible part of $\overline{uw}$ using the following algorithm:

Algorithm 3 (to find the visible part of $\overline{uw}$ through mirror-edge e).
Let $u'(e)$ and $w'(e)$ corresponding to u and w, be the endpoints of the visible interval of $\overline{uw}$, respectively.

Note that Algorithm 2 provides all LBV and RBV vertices of all mirror-edges.

1. If $LBV(e) = v_1(e)$: Then the intersection of $L_1(e)$ and $\overline{uw}$ determines $u'(e)$. Clearly, if $L_1(e)$ places in the left side of $\overline{uw}$ then u itself is $u'(e)$.
2. If $LBV(e) \neq v_1(e)$: If $LBV(e)$ does not lie on the right side of $L_1(e)$, then again the intersection of $L_1(e)$ and $\overline{uw}$ determines $u'(e)$. Otherwise, we compute the intersection of the protraction of $\overline{q'(e)LBV(e)}$ and $\overline{uw}$. The intersection point is $u'(e)$.

Acting the same way we can find w'.

Correctness and analysis of Algorithm 3

First step is obvious because there is nothing to obstruct the mirror-visibility area, and it takes constant time. About the second step, if $LBV(e)$ lies–on or– on the left side of $L_1(e)$, the intersection point of $L_1(e)$ and $\overline{uw}$ is $u'(e)$. Note that we know $L_1(e)$ is not in the right side of w because we knew $\overline{uw}$ is e-mirror-visible to q. If $LBV(e)$ lies on the right side of L_1, then from Definition 1 we know $LBV(e)$ is e-mirror-visible. We only need to prove that the protraction of $\overline{q'(e)LBV(e)}$ determines $u'(e)$. There may be several reflex vertices on the right side of $L_1(e)$. Suppose on the contrary, $u''(e)$, the intersection of $\overline{uw}$ and $\overline{q'p_j}$ ($p_j \neq LBV(e)$ is a reflex vertex on the right side of L_1), is closer to u. Then, the line $\overline{q'p_j u''(e)}$ must be on the right side of $LBV(e)$, which contradicts Definition 1 (see Figure 8 in Appendix).

Since no direction for $L_1(e)$, or property of q being in the left side of e was used, the same proof holds for $RBV(e)$.

5 Extending a segment visibility polygon

In this section, we deal with different cases of the problem of making two invisible segments mirror-visible to each other.

Lemma 2. *We are given a simple polygon $\mathcal{P}$ and two segments, say $\overline{xy}$ and $\overline{uw}$, inside $\mathcal{P}$. Assume that $\overline{uw}$ is not visible to $\overline{xy}$. For every mirror-edge e, we can find out if $\overline{uw}$ is weakly, completely, or strongly mirror-visible to $\overline{xy}$, in linear time corresponding to the complexity of $\mathcal{P}$.*

Proof. To prove Lemma 4 we simply use Algorithm 1 in Section 3. Here, as we deal with a segment as a viewer, we encounter more difficulties than the previous sections. For instance, we need to consider different vertices in place of $v_1(e)$, or $v_2(e)$ in Algorithm 1. And, to find these verices the intersection of different visibility polygons maybe required. Also, different half-lines may be as replacement for $L_1(e)$ and $L_2(e)$.

We have the following cases:

1. The whole $\overline{xy}$ can see the whole $\overline{uw}$.
2. The whole $\overline{xy}$ can see at least one point of $\overline{uw}$.
3. $\overline{xy}$ can see the whole $\overline{uw}$ in a weak visible way.
4. At least one point of $\overline{xy}$ can see at least one point of $\overline{uw}$.

We deal with these cases in the following subsections. Without loss of generality, consider a mirror-edge e on $\mathcal{P}$. In each subsection, we find appropriate substitutes for $v_1(e)$, $v_2(e)$, $L_1(e)$, and $L_2(e)$.

5.1 The whole $\overline{xy}$ can see the whole $\overline{uw}$

First, we compute the intersection visibility polygon of the endpoints of $\overline{xy}$ (x and y). Then, while tracing the completely visibility polygon of $\overline{uw}$ ($CVP(\overline{uw})$), we select the common part of each edge with the intersection visibility polygon of the endpoints. As a result, we have $v_1(e)$ and $v_2(e)$ for every mirror-edge e. Obviously, this step only takes $O(n)$ time complexity.

Consider x as a viewer, let the reflective ray from $v_1(e)$ be $L_{1,x(e)}$, and the reflective ray from $v_2(e)$ be $L_{2,x(e)}$. Similarly, we define $L_{1,y(e)}$ and $L_{2,y(e)}$.

We should use $L_{1,x(e)}$ as $L_1(e)$, and $L_{2,y(e)}$ as $L_2(e)$ in Algorithm 1. Since we know any potential mirror-edge from $v_1(e)$ to $v_2(e)$ is completely visible for $\overline{xy}$, it is sufficient to check $L_{1,x(e)}$ to lie in the left side of u, and $L_{2,y(e)}$ to lie in the right side of w.

5.2 The whole $\overline{xy}$ can see at least one point of $\overline{uw}$

In this subsection, we want to find out if there is any point on $\overline{uw}$ which is e-mirror-visible to the whole $\overline{xy}$.

We can use a method similar to the previous subsection, only now the strongly visibility polygon of $\overline{uw}$ ($SVP(\overline{uw})$) is required. We use $L_{1,x(e)}$ as $L_1(e)$, and $L_{2,y(e)}$ as $L_2(e)$.

Considering $SVP(\overline{uw})$, there is an interval or at least a point on $\overline{uw}$ which holds the property of being strongly visible.

For the last step, we need to find out if this point or segment has intersection with the interval from $u'(e)$ to $w'(e)$.

5.3 $\overline{xy}$ can see the whole $\overline{uw}$ in a weak visible way

There may be no point on $\overline{xy}$ to see the whole $\overline{uw}$ by itself. Here, we want to find out if $\overline{uw}$ is completely e-mirror-visible considering all the points on $\overline{xy}$.

We use the intersection of $WVP(\overline{xy})$ and $CVP(\overline{uw})$, to find all the potential mirror-edges (v_1 and v_2 vertices).

Since we deal with the weak visibility polygon, we may face some mirror-edges which are visible to none of the endpoints of $\overline{xy}$, but to an interval of $\overline{xy}$ in the middle. We need to find this interval for each mirror-edge. In fact different mirror-edges may have different points on $\overline{xy}$, to make their L_1 and L_2 half-lines. It is sufficient to check these half-lines with the endpoints of $\overline{uw}$ to make sure that the mirror-visibility region covers $\overline{uw}$ completely.

For a specific mirror-edge e_i, let $x(e_i)$ and $y(e_i)$ be the points on $\overline{xy}$ corresponding to x and y respectively. We can use the ray reflection of $x(e_i)$ on e_i as $L_1(e_i)$, and the ray reflection of $y(e_i)$ as $L_2(e_i)$ in Algorithm 1. In $O(n)$ time we can find these points on $\overline{xy}$ for all mirror-edges through the following way:

Definition 2. *Consider a potential mirror-edge e (from $v_1(e)$ to $v_2(e)$) such that there are two reflex vertices that block the visibility of a portion of $\overline{xy}$ before $v_1(e)$ and after $v_2(e)$ in $\mathcal{P}$'vertex order. Define $r_1(e)$ and $r_2(e)$ to be these reflex vertices, respectively.*

Obviously, if there is no $r_1(e)$ or $r_2(e)$ then there is no obstruction, and we can use corresponding $v_1(e)$ and $v_2(e)$, to find $L_1(e)$ and $L_2(e)$.

See Figure 4, in this figure we have $r_1(e)$ and $r_2(e)$ vertices. The blue sub-segment of $\overline{xy}$ can see e completely, but all the points –from $x(e)$ to the blue sub-segment, and from the blue sub-segment to $y(e)$– cannot see at least some part of e. For the points on the other side of these yellow points, e is not visible. The reflected rays from e ,which is between the green half-lines, is the area which segment $\overline{xy}$ can see, in a weak visible way, through e. We call these half-lines $L_{1,y(e)}$ and $L_{2,x(e)}$.

In order to find $x(e)$ and $y(e)$, we only need $r_1(e)$ and $r_2(e)$, because we can protract $\overline{v_2(e)r_1(e)}$ and $\overline{v_1(e)r_2(e)}$ to find their intersection with $\overline{xy}$. The intersection points are $x(e)$ and $y(e)$.

Suppose there are m potential mirror-edges, we should find $r_1(e_j)$ and $r_2(e_j)$ $1 \leq j \leq m$. The idea is similar to Algorithm 2.

Computing $r_1(e)$ and $r_2(e)$ reflex vertices for all mirror-edges:

To compute these reflex vertices we use two convex shapes over the reflex vertices in two directions. For a particular mirror-edge e, $\overline{r_1(e)v_2}$ should hold all left-side reflex vertices on its left, and of course $\overline{r_2(e)v_1}$ should hold all the right-side reflex vertices on its right. Note that it is not important if there were more than one reflex vertex on either $\overline{r_1(e)v_2}$ or $\overline{r_2(e)v_1}$ (see Figure 5).

In this subsection, we use $L_{1,y(e)}$ and $L_{2,x(e)}$ instead of $L_1(e)$ and $L_2(e)$ respectively. Also, while using Algorithm 2, we need $CVP(\overline{uw})$ in place of $WVP(\overline{uw})$ to construct $TP(\overline{uw})$.

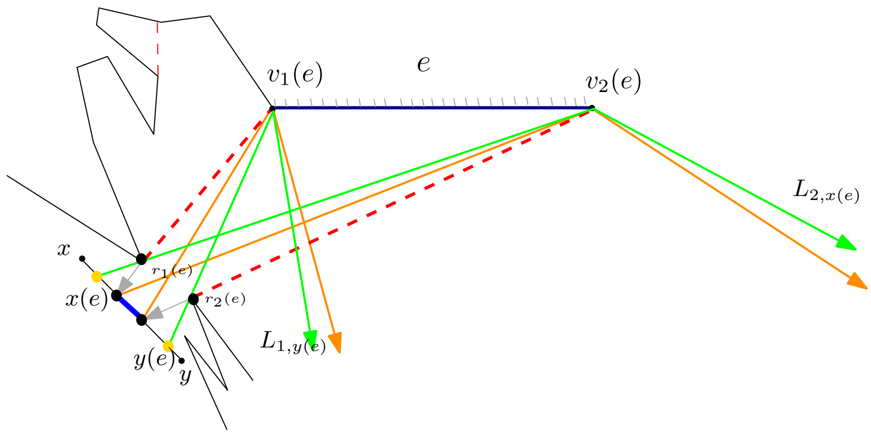

Fig. 4. $r_1(e)$, $r_2(e)$, $x(e)$ and $y(e)$ are shown for mirror-edge e.

5.4 At least one point of $\overline{xy}$ can see at least one point of $\overline{uw}$

Here we can behave similar to the previous subsection except that we need $WVP(\overline{xy}) \cap WVP(\overline{uw})$ to find potential mirror-edges. And, considering a mirror-edge e, we use $L_{1,x(e)}$ and $L_{2,y(e)}$ half-lines to be used in Algorithm 1.

Also, we need $WVP(\overline{uw})$ in the construction of $TP(\overline{uw})$ because it is sufficient to make e-mirror-visible any point on $\overline{xy}$ to any point on $\overline{uw}$.

6 Discussion

We dealt with the problem of extending the visibility polygon of a given point or a segment in a simple polygon, so that another segment becomes visible to the viewer.

We tried to achieve this purpose by converting some edges of the polygon to mirrors. The goal is to find all such kind of edges, and the mirror-visible part of the target segment by each of these edges individually. Using the algorithm we proposed, this can be done in linear time corresponding to the complexity of the simple polygon.

We covered all the possible types of visibility when we dealt with a given segment as a viewer, and we wanted to extend its visibility to see another given segment. We proved all the possible cases need just $O(n)$ time.

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 [7]. Also, having h mirrors, the number of vertices of the resulting visibility polygon, can be $O(n + h^2)$, and for h mirrors, each projection, and its relative visibility polygon can be computed in $O(n)$ time, which leads to overall time complexity of $O(hn)$.

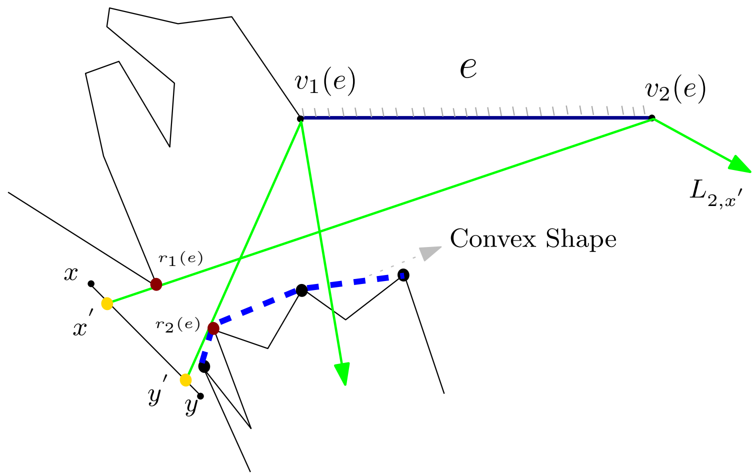

Fig. 5. Constructing convex shape similar to Algorithm 2.

The problem can be extended as; put mirrors inside the polygon, a point with a limited visibility area, find some edges which can give the point a specific vision or different visions and so on.

References

1. A. Vaezi, M. Ghodsi. Extending Visibility Polygons by Mirrors to Cover Specific Targets. *EuroCG2013*, 13–16, 2013.
2. B. Aronov, A. Davis, T. Day, S. P. Pal, D. Prasad. Visibility with one reflection. *Discrete & computational Geometry*, 19: 553–574, 1998.
3. 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.
4. 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.
5. 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.
6. 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.
7. D. T. Lee. Visibility of a simple polygon. *Computer Vision, Graphics, and Image Processing*, 22: 207–221, 1983.

Appendix

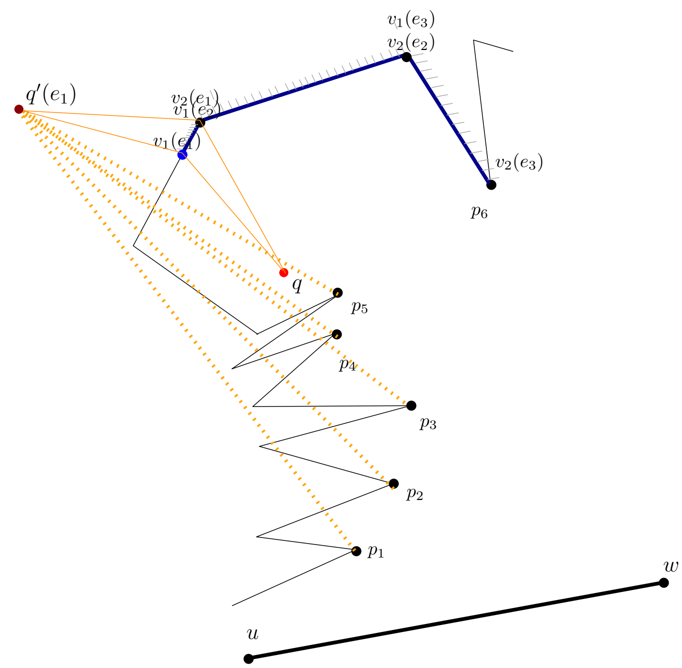

Fig. 6. From Definition 1, vertex p_5 is $LBV(e_1)$.

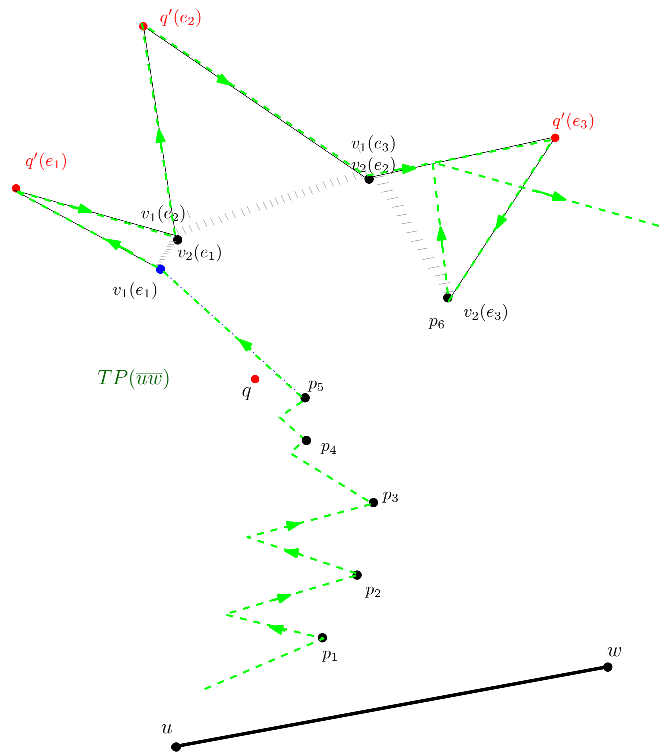

Fig. 7. Constructing $TP(\overline{uw})$, which is useful to distinguish LBV vertices for all mirror-edges. p_1, p_2, ... ,p_6 are the reflex vertices of $\mathcal{P}$.

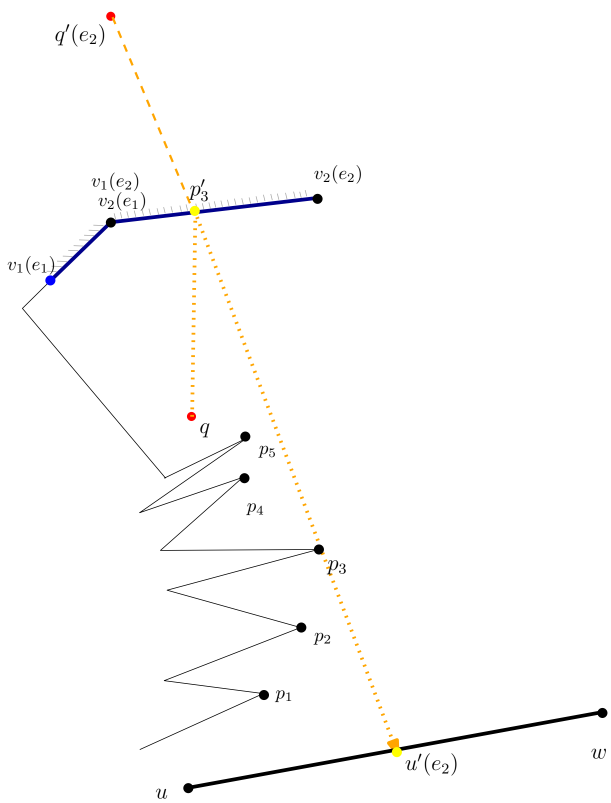

Fig. 8. $p_3 = LBV(e_2)$, and the intersection of the protraction of $\overline{q'(e_2)p_3}$ and $\overline{uw}$ is $u'(e_2)$.

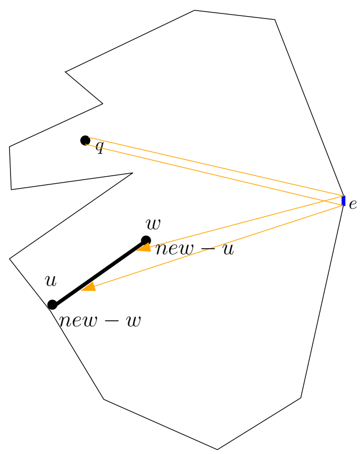

Fig. 9. Mirror-edge e sees $\overline{uw}$ from its behind. And, we need to replace w with u and run all algorithms one more time in order to find these kinds of mirror-edges.