# Computing $k$-Link Visibility Polygons in Environments with a Reflective Edge

Salma S. Mahdavi * Ali Mohades * Bahram Kouhestani *

## Abstract

In this paper we consider the $k$-link visibility polygon of an object inside a polygonal environment with a reflective edge called a *mirror*. The $k$-link visibility polygon of an object inside a polygon $P$ is the set of all points in $P$, which are visible to some points of that object with at most $k-1$ intermediate points, under the property that consecutive intermediate points are mutually visible. We propose an optimal linear time algorithm for computing the $k$-link visibility polygon of an object inside a polygon $P$ with a reflective edge. The object can be a point, a segment or a simple polygon. We observed that in computing $k$-link visibility polygons the mirror can affect in only two levels. We explain how to handle these levels efficiently to achieve an optimal algorithm.

## 1 Introduction

The visibility problem is a fundamental topic in computational geometry and different versions of it, such as art gallery problems, have been studied. The visibility polygon of an object is defined as the set of all points visible to some points of that object. Several linear time algorithms have been proposed to compute the visibility polygon of a point [8], a segment [4] or a polygon [7]. The minimum link path between two points of a polygon $P$ is a path inside $P$ that connects these points and has the minimum number of straight edges, called *links*. The link distance between two points is the number of links in their minimum link path. Suri [10] gave an $O(n)$ time algorithm for computing the minimum link path between two points in a simple polygon. The $k$-link visibility polygon of a point $q$ can be defined by using the minimum link distance concept; it is the set of all points having the link distance of at most $k$ from $q$. Using the window partitioning the $k$-link visibility polygon of a point can be computed in linear time [9].
In the visibility literature, reflective surfaces were first mentioned by Klee [5]. He asked if every polygon whose all edges are reflective can be illuminated from any interior point. Tokarsky [11] answered no to this question by constructing a polygon for which there exists a dark point by putting a light source at a particular position. In this polygon the dark point is collinear with both the light source and an edge of the polygon.
When all the edges of the polygon are reflective but each light beam is allowed to reflect once, Aronov et al. [2] showed that the resulting visibility polygon of a source light may not be simple. They present an $O(n^2 \log^2 n)$ algorithm for computing visibility polygons in such environments. Later they allowed at most $r$ reflections for each light beam and presented an $O(n^{2r} \log n)$ time and $O(n^{2r})$ space algorithm to compute visibility polygons of a source light [1].
Recently Kouhestani et al. [6] showed how to compute visibility polygons in environments with a single reflective edge in an optimal $O(n)$ time. In this paper, we extend this study and achieve a linear time algorithm for computing $k$-link visibility polygons in such environments. If the polygon is entirely contained within one of the 2 half-spaces determined by the line on which the mirror lies, the $k$-link visibility polygon of an arbitrary point can be easily computed by gluing $P$ and the reflection of $P$ in the mirror (called $P'$) together along the reflective edge. Known algorithms for computing the $k$-link visibility polygon [9] can be slightly modified to operate in such a polygon. To return the visibility polygon, the union of two visibility polygons is computed in linear time using the algorithm of Kouhestani et al. [6]. In the case that the polygon intersects both these half-spaces and vertices of the mirror are reflex, the resulting polygon from gluing $P$ and $P'$ is not simple anymore. Apart from the difficulties to adopt known algorithms to operate in this polygon, this method needs improvements to run in an optimal time. Consider the smallest value of $k$ for which the $k$-link visibility polygon enters $P'$. In the computation of $(k+1)$-link visibility polygon, points of $P$ which are visible from points of $k$-link visibility polygon located in $P$ must be added to those points of $P$ visible from the part of $k$-link visibility polygon located in $P$. We have a similar situation for the points of $(k+1)$-link visibility polygon located in $P'$. Therefore, in order to return the $(k+1)$-link visibility polygon the union of four polygons must be computed. It is not clear how to accomplish this task in linear time to obtain an optimal algorithm.
Suppose the $k$-link visibility polygon is constructed incrementally and for example in $i^{th}$ level the $i$-link visibility polygon is computed from the resulting polygon of the previous level. We observed that the reflective edge affects only two levels, therefore handling these levels ef-

*Laboratory of Algorithm and Computational Geometry, Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran, ss.mahdavi@aut.ac.ir, mohades@aut.ac.ir, b_kouhestani@alum.sharif.edu.

ficiently can produce a linear time algorithm. We clarify this observation in the first lemma of section 4.2.

This paper is organized as follows: Section 2 describes notation and preliminaries, section 3 shows how to compute the 2-visibility polygon of an object inside a polygon with a reflective edge, from which in section 4 an algorithm to compute $k$-link visibility polygons in such an environment is proposed. Section 5 contains conclusions and discussions.

## 2 Notation and Preliminaries

Let $P$ be a simple polygon with $n$ vertices. $int(P)$ and $bd(P)$ denote the interior and boundary of $P$, respectively. Two points $x, y \in P$ are mutually visible (or can see each other directly), if the open line segment, $\overline{xy}$, lies completely in $int(P)$. An alternative definition used in many visibility papers, allows the line segment to touch the $bd(P)$. Throughout this paper we use the former definition which is sometimes called the *clear visibility*. Two points $x$ and $y$ inside $P$ are $k$-link visible (or for simplicity $k$-visible), if they can reach each other using $k-1$ intermediate points $a_1, ..., a_{k-1}$, under the property that $a_i$ and $a_{i+1}$ are mutually visible for $1 \leq i \leq (k-2)$ and $x$ sees $a_1$ and $y$ sees $a_{k-1}$. The visibility polygon of a point $q$ in $P$, denoted by $V_P(q)$ is the set of all points in $P$ visible to $q$. An edge of $V_P(q)$ that is not a part of an edge of $P$ is called a *window* of $V_P(q)$. Suppose $w_i$ is a window of $V_P(q)$. $w_i$ partitions $P$ into two subpolygons. The subpolygon which does not contain $V_P(q)$ is called the *pocket* of $w_i$ ($pocket(w_i)$).

The $k$-link visibility polygon of a point $q$ in $P$, denoted by $V_P^k(q)$ is the set of all points in $P$ which are $k$-visible to q. The weak visibility polygon of a segment $s$ denoted by $WV_P(s)$ is the set of all points of $P$ visible to some points on $s$, except the endpoints. In the same manner, the $k$-(weak)visibility polygon of a segment $s$, $V_P^k(s)$ can be defined. Let $q$ be a point inside $P$. $P$ can be partitioned into regions such that all points in the same region have the same link distance from $q$. This partitioning is called the *window partitioning* with respect to the point $q$ and can be done in $O(n)$ time [9]. Two regions are neighbors if they share a common window. The dual graph of the window partitioning is achieved by considering a node for each region and connecting each two nodes if their corresponding regions are neighbors.

Suppose one of the edges of $P$ is reflective, this edge is called a *mirror*. Two points $x$ and $y$ can see each other through the mirror $e$ (or indirectly), if and only if there exist a point $r$ lying on $e$, visible to both $x$ and $y$ such that $\overline{xr}$ and $\overline{yr}$ lie on the opposite sides of the inward normal at $r$ and make the same angle with it. $r$ is not considered as an intermediate point and hence $x$ and $y$ are 1-link visible. Figure 1 illustrates two points $y$ and $z$ which are 2-link visible to the point $x$ with the intermediate points $i_1$ and $i_2$, respectively.

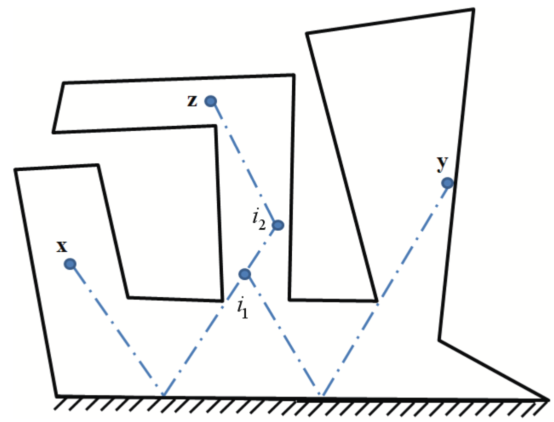


Figure 1: An illustration of 2-link visibility when one of the edges of the polygon is reflective.

Let $V_{P,e}(q)$, (resp. $V_{P,e}^k(q)$) denote the visibility polygon (resp. the $k$-link visibility polygon) of a point $q$ inside $P$ with the reflective edge $e$. Note that in $V_{P,e}^k(q)$, consecutive intermediate points can see each other directly or by using the mirror $e$.

## 3 The 2-visibility polygon of objects in a polygon with a single reflective edge

We first concentrate on computing the 2-visibility polygon of an object when an edge of the polygon is reflective. Let $P$ be a simple polygon with $n$ vertices and $e$ be the reflective edge of $P$. Let $O$ be an object inside $P$. The general idea of computing $V_{P,e}^2(O)$ is to compute $V_{P,e}(O)$ and identify parts of $P$ that are 1-visible to $V_{P,e}(O)$. It is easy to see that any point of these parts is visible to a window of $V_{P,e}(O)$. If $O$ is a point or a segment $V_{P,e}(O)$ is computed in $O(n)$ time [6]. When $O$ is a simple polygon with $m$ vertices, we use the following process to compute $V_{P,e}(O)$:

First we compute $V_P(O)$ in $O(n+m)$ time by using the algorithm of Langetepe et al. [7], and then determine the parts of $P$ that $O$ sees through $e$. This can be done with a slight modification to the algorithm of Kouhestani et al. [6]. The union of these parts is $V_{P,e}(O)$ which can be computed in $O(n+m)$ time [6].

**Lemma 1** *Let $Q$ be a simple polygon with $m$ vertices inside the simple polygon $P$. $V_{P,e}(Q)$ can be computed in $O(n+m)$ time.*

$V_{P,e}^2(O)$ is the set of all points in $P$ which see some points of $O$ using at most one intermediate point. The points of $V_{P,e}^2(O)$ can be categorized into four groups:

($a$) The points which see the intermediate point directly for which the intermediate point sees the object directly.

($b$) The points which see the intermediate point using the mirror, but for which the intermediate point sees the object directly.
($c$) The points which see the intermediate point directly, but for which the intermediate point sees the object using the mirror.
($d$) The points which see the intermediate point using the mirror for which the intermediate point sees the object using the mirror.
We denote these groups by $\{-e, -e\}$, $\{+e, -e\}$, $\{-e, +e\}$, $\{+e, +e\}$, in which $-e$ means seeing directly and $+e$ means seeing through the mirror $e$.

**Lemma 2** *Let $P$ be a simple polygon with a reflective edge $e$, and $O$ be an object completely inside $P$. At most two windows of $V_{P,e}(O)$ can intersect $e$.*

**Proof.** Any line segment in $P$, can intersect with at most two windows of a visibility polygon [3]. Therefore, the edge $e$ intersects at most two windows of $V_{P,e}(O)$. $\square$

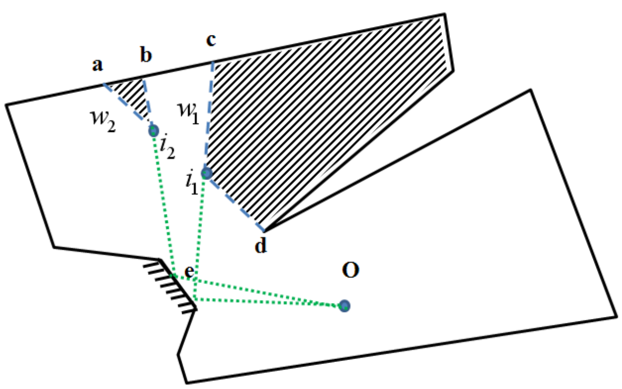


Figure 2: The windows $w_1 = (c, i_1, d)$ and $w_2 = (a, i_2, b)$ with interior points $i_1, i_2$. The shaded regions show the pocket of $w_1$ and $w_2$.

Note that some windows of $V_{P,e}(O)$ may have an interior point (see Figure 2).

**Lemma 3** *Suppose $w_i$ is a window of $V_{P,e}(O)$ and its pocket dose not intersect the mirror edge $e$. Then, all points of $V^2_{P,e}(O)$ which are located in $pocket(w_i)$, are directly visible from $w_i$. Therefore, other windows of $V_{P,e}(O)$ can not see additional points in $pocket(w_i)$ and $w_i$ can not see new points in $pocket(w_i)$ through the mirror.*

**Proof.** Let $w_j$ be a window of $V_{P,e}(O)$ and $w_j \neq w_i$. If a point $x$ in $pocket(w_i)$ is visible to $w_j$ (either directly or by using the mirror), the line segment (or the path) $\overline{w_j x}$ will intersect $w_i$ in a point $y$. Therefore, $x$ and $y$ are visible and $x$ can see $w_i$ directly (see Figure 3). With a similar argument $w_i$ can not see new points in its pocket through the mirror. $\square$

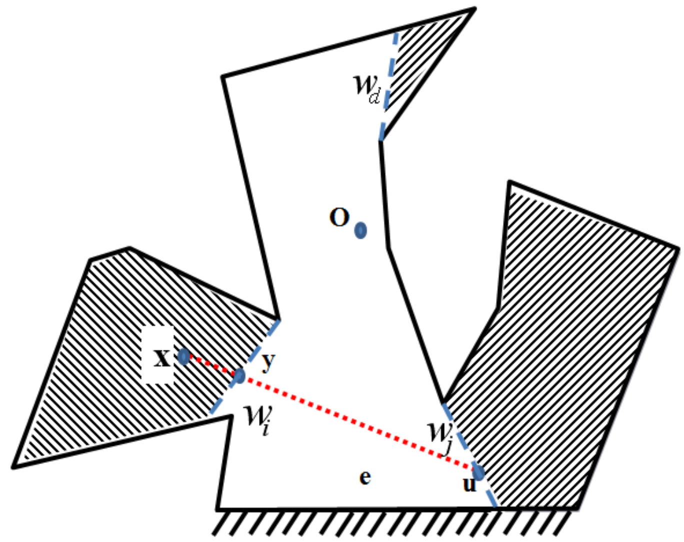


Figure 3: The illustration of Lemma 3.

**Lemma 4** *Let $w_i$ be a window of $V_{P,e}(O)$ whose pocket intersects the reflective edge $e$. The set of points of $V^2_{P,e}(O)$ located in $pocket(w_i)$ is equal to $V_{pocket(w_i),e}(w_i)$.*

**Proof.** The proof is similar to the proof of the previous lemma. $\square$

**Lemma 5** *Let $Q$ be a polygon constructed by adding all $V_{pocket(w_i),e_i}(w_i)$ to $V_{P,e}(O)$, where $w_i$ is a pocket of $V_{P,e}(O)$ and $e_i$ is the intersection of $e$ and $pocket(w_i)$. Then, $Q$ is $V^2_{P,e}(O)$.*

**Proof.** Windows of $V_{P,e}(O)$ are 1-visible to $O$, so newly added points are 2-visible to $O$. Therefore, $Q$ is a subset of $V^2_{P,e}(O)$. As mentioned before, points of $V^2_{P,e}(O)$ can be categorized into four groups. We show that a point of each group lies in $Q$:
Case 1: $\{-e, -e\}$; a point of type $\{-e\}$ is in $V_P(O)$, and $\{-e, -e\}$ is a point which is directly visible to $V_P(O)$, so it is visible to a window of $V_P(O)$ and is located in the pocket of this window. $V_P(O)$ is a subset of $V_{P,e}(O)$, therefore, the point is in $Q$.
Case 2: $\{+e, -e\}$; a point of type $\{+e\}$ is in $V_{P,e}(O)$, $\{+e, -e\}$ is a point which is directly visible to a window of $V_{P,e}(O)$, so it is located in the pocket of this window, therefore, it is in $Q$.
Case 3: $\{-e, +e\}$; a point of type $\{-e, +e\}$ is visible to $V_P(O)$ by using the mirror, $V_P(O)$ is a subset of $V_{P,e}(O)$, so this point is visible to a window of $V_{P,e}(O)$ by using the mirror, and therefore, it is in $Q$.
Case 4: $\{+e, +e\}$; $\{+e, +e\}$ is a point visible to a window of $V_{P,e}(O)$ by using the mirror, so it is in $Q$. $\square$

Now we can present an algorithm to compute the 2-visibility polygon of an object $O$ in linear time.

**Algorithm 3**

1. Compute $V_{P,e}(O)$.
2. Let $\{w_1, ..., w_d\}$ be the windows of $V_P(O)$ and $e_i$ be the intersection of $e$ and $pocket(w_i)$ for $i = 1, 2, ..d$.
3. Compute all $V_{pocket(w_i),e_i}(w_i)$ and add them to $V_{P,e}(O)$.
4. Return the resulting polygon.

**The time complexity of the algorithm:**
Step 1 is computed in $O(n + m)$ time due to Lemma1. Suppose $n_i$ is the number of vertices of $pocket(w_i)$ for $i = 1, 2, ..d$. $V_{pocket(w_i),e_i}(w_i)$ is computed in $O(n_i)$ time. Each two pockets of $V_{P,e}(O)$ can at most share one vertex, therefore, the sum of vertices of these pockets is $O(n)$ and step 3 is computed in linear time.
Therefore, we can conclude the following theorem.

**Theorem 6** *The 2-visibility polygon $V^2_{P,e}(O)$ of an object $O$ inside a simple polygon $P$ with a reflective edge $e$, can be computed in $O(n + m)$ time, where $n$ is the number of vertices of $P$ and $m$ is the complexity of $O$.*

## 4 $k$-visibility polygons

### 4.1 The $k$-visibility polygon of an object

Let $O$ and $s = \overline{xy}$ be an object and a segment inside a simple polygon $P$ with $n$ vertices. The minimum link path between $O$ and $s$, is a path in $P$, connecting some points of $O$ to $s$, which has the minimum number of links. The link distance between $O$ and $s$, is the number of the links in their minimum link path. The set of all points with the link distance of 1 from $O$ is the visibility polygon of $O$. Let $w_i$ be the window of $V_P(O)$ which its pocket completely contains $s$. If there is no such a window then the link distance between $O$ and $s$ is 1. Otherwise, the link distance between $O$ and $s$ is 1 + the link distance between $s$ and $w_i$. The link distance between two segments can be computed in $O(n)$ time [10, 9]. Therefore, the link distance between an object and a segment can be computed in linear time. Note that the window partitioning is computed using the link distance concept [9], so the time complexity of the window partitioning of an object is linear.

**Lemma 7** *Let $P$ be a simple polygon with $n$ vertices and $O$ be an object inside $P$ with the complexity of $m$. $V^k_P(O)$ can be computed in $O(n + m)$ time.*

**Proof.** A point $x$ inside $P$ is in $V^k_P(O)$ if the link distance between $O$ and $x$ is less or equal to $k$. By using the window partitioning of $O$ all points with the link distance of at most $k$ from $O$ can be computed in $O(n+m)$ time. □

### 4.2 The $k$-visibility polygon of an object inside an environment with a single reflective edge

Let $e$ be a reflective edge in $P$. Two points in $P$ are $k$-visible if they can see each other using at most $k - 1$ intermediate points. Consecutive intermediate points see each other directly or by using $e$. $V^k_{P,e}(O)$ is the set of all points which are $k$-visible to some points of $O$. By using the computation of the 2-visibility polygon of $O$, we present an algorithm to compute $V^k_{P,e}(O)$.
For an example of a 3-link visibility polygon of a point in the presence of a mirror, see Figure 4. In this figure, windows of each level are shown with the same color.

**Lemma 8** *Let $P$ be a simple polygon with a reflective edge $e$ and $O$ be an object inside $P$. Let $m$ be the link distance between $O$ and $e$. Suppose $Q = V^{m+1}_{P,e}(O)$ is constructed. $V_{P,e}(Q)$ is equal to $V_P(Q)$.*

**Proof.** $O$ and $e$ are $m$-link visible, therefore, $e$ is completely inside $V^{m+1}_P(O)$ and no pockets of $V^{m+1}_P(O)$ intersect $e$. Let $pock_i$ be a pocket of $V^{m+1}_P(O)$. Similar to Lemma 3, all points of $V^{m+2}_P(O)$ in $pock_i$ are directly visible to the window of $pock_i$. Therefore, $V_{P,e}(Q)$ is equal to $V_P(Q)$. □

Now we present an algorithm to compute $V^k_{P,e}(O)$:

**Algorithm 4.2**

1. Compute the link distance between $O$ and $e$ and store it in $m$.
2. If $k < m$, then, compute $V^k_P(O)$ and return.
3. If $k \geq m$, then,
   (a) Compute $Q = V^{m-1}_P(O)$.
   (b) If $k = m$, compute $V_{P,e}(Q)$ and return.
   (c) Compute $R = V^2_{p,e}(Q)$
   (d) Let $\{w_1, w_2, ..., w_d\}$ be windows of $R$.
   (e) Compute $V^{k-m-1}_{pocket(w_i)}(w_i)$ for all $i = 1, 2, ..., d$ and add them to $R$
   (f) Return the resulting polygon

**Theorem 9** *Let $P$ be a simple polygon with $n$ vertices and $O$ be an object inside $P$, with the complexity of $m$. Let $e$ be a reflective edge of $P$. $V^k_{P,e}(O)$ can be computed in $O(n + m)$ time.*

**Proof.** The algorithm 4.2 computes $V^k_{P,e}(O)$. Lemma 8 ensures the correctness of the algorithm. We show that the time complexity of this algorithm is $O(n+m)$. Steps 3(a), 3(b), 3(c) and 3(d) are computed in $O(n+m)$ time due to Lemma 7, Lemma 1 and Theorem 6. Suppose the number of vertices of $pocket(w_i)$ is $n_i$, for all $i =$

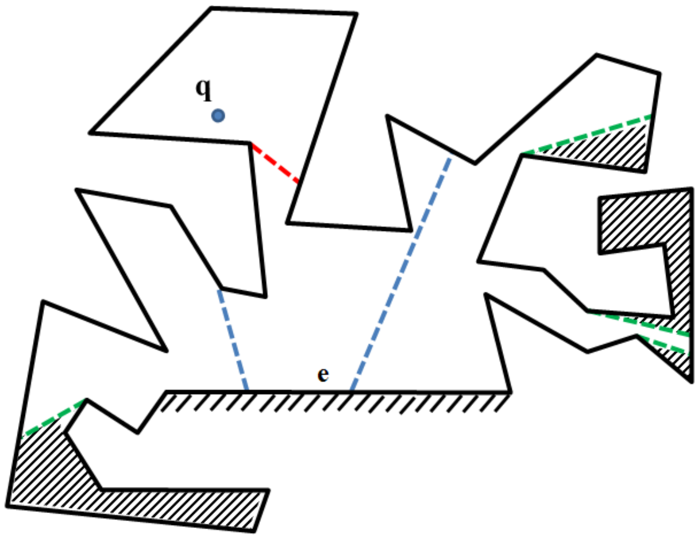


Figure 4: The 3-link visibility polygon of a point, when one of the edges of the polygon is reflective.

$1, 2, ..., d$. $V^{k-m-1}_{pocket(w_i)}(w_i)$ is computed in $O(n_i)$. The sum of all $n_i$ for $i = 1, 2, ..., d$ is less or equal to $n$. Therefore, step 3(e) runs in $O(n)$ time and the algorithm has the time complexity of $O(n + m)$. □

Note that the time complexity of computing $V^k_{P,e}(O)$ is independent to the value of $k$ and for any $k = 1, ..., n$, $V^k_{P,e}(O)$ is computed in $O(n + m)$ time.

## 5 Conclusion

In this paper we presented a linear time algorithm to compute the $k$-link-visibility polygon of an object inside a polygonal environment with a reflective edge. The object is considered to be a point, a segment or a simple polygon. If the polygon has two or more reflective edges, a light beam can be reflected repeatedly between these mirrors. It is not clear how to compute the visibility polygon of a point when there is no restriction on the number of the reflections. An interesting question will be how to compute the $k$-link visibility polygon of a point in a polygon with $m$ reflective edges when each light beam can reflect at most $t$ time. This question is the subject of our future study.

## Acknowledgments

We would like to thank Mansour Davoudi and Farnaz Sheikhi for their fruitful comments.

## References

[1] B. Aronov, A. Davis, T. K. Dey, S. P. Pal and D. C. Prasad. Visibility with multiple reflections . *Discrete and Computational Geometry*, 20: 61-78, 1998.

[2] B. Aronov, A. Davis, T. K. Dey, S. P. Pal and D. C. Prasad. Visibility with one reflection. *Discrete and Computational Geometry*, 19: 553-574, 1998.

[3] P. Bose, A. Lubiw, and J. Munro. Efficient visibility queries in simple polygons. *Computational Geometry: Theory and Applications*, 23: 313-335, 2002.

[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] V. Klee. Is every polygonal region illuminable from some point? *Computational Geometry: Amer.Math. Monthly*, 76: 180, 1969.

[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] E. Langetepe, R. Penninger and J. Tulke. Computing the visibility area between two simple polygons in linear time. *In Proc. 26th European Workshop on Computational Geometry*, 26: 237-240, 2010.

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

[9] A. Maheshwari, J. -R. Sack and H. N. Djidjev. Link Distance Problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, 519-558, 2000.

[10] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. *Computer Graphics Vision, and Image Processing*, 35: 99-110, 1986.

[11] G. T. Tokarsky. Polygonal rooms not illuminable from every point. *American Mathematical Monthly*, 102: 867-879, 1995.