# Computing the visibility area between two simple polygons in linear time

Elmar Langetepe* Rainer Penninger* Jan Tulke†

**Abstract**

We consider a visibility problem for two non-intersecting open or closed simple polygonal chains $P$ and $Q$ in the plane. The visibility area between $P$ and $Q$ is the union of all line segments $pq$ where $p$ lies on the boundary of $P$ and $q$ on the boundary of $Q$ and $pq$ does neither intersect with $P$ nor with $Q$. We present an optimal linear time algorithm for computing this area. The given work generalizes known visibility results and has an application in computer aided construction management.

**Keywords:** Visibility in the plane, polygonal chains, optimal algorithm

## 1 Introduction

Computing the visibility among geometric objects is one of the most natural subjects in the field of computational geometry. Furthermore, efficient solutions of visibility problems have application in many fields of computer science such as robotics [8], computer graphics [3] and computer vision [4]. For an overview of efficient solutions for visibility problems in the plane one can consider the survey of Asano et al.[1] or the textbook of Gosh[5].

The given result has two main benefits. On one hand, we generalize a known visibility result inside simple polygons. It was already shown how to compute the inner visibility region of an edge $e$ of a simple polygon $P$ efficiently. Instead of $e$ and $P$ we allow more general polygonal objects $P$ and $Q$ while maintaining optimal linear time. The visibility area between $P$ and $Q$ is the union of all line segments $pq$ where $p$ lies on the boundary of $P$ and $q$ on the boundary of $Q$ and $pq$ does neither intersect with $P$ nor with $Q$, see Figure 1 for an example. Furthermore, with the same technique we can compute the visibility region of a polygon $P$ inside a polygon $Q$ in time proportional to $|P| + |Q|$ which is a natural extension of visibility of a single point or egde. On the other hand, the given problem arises in the context of a digital three dimensional building construction model. An efficient computation of the visibility area between two wall axes $A$ and $B$ in the digital model of a building is an important question for computer aided construction management.

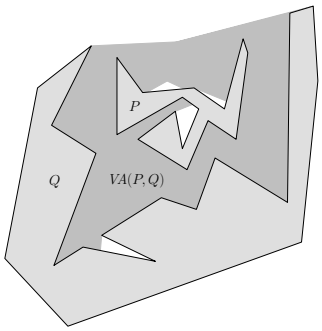


Figure 1: The visibility area between two non-intersecting polygonal chains $P$ and $Q$.

In Section 2 we extend the algorithm of Guibas et al. [6] and compute the visibility region of a polygonal subchain $C_P$ along the boundary of a polygon $P$ in linear time. Afterwards, we make use of this result and compute the visibility area between two non-intersecting polygonal objects $P$ and $Q$ in time $O(|P| + |Q|)$. Note that due to lack of space some proofs are omitted.

## 2 Visibility of a boundary subchain

We would like to compute the inner visibility region of a connected subchain $C_P = (p_1, \ldots, p_k)$ along the boundary $\partial P$ of a simple polygon $P = (p_1, \ldots, p_n, p_1)$, see Figure 2. A point $q \in P$ is *visible* from $p \in C_P$ if $pq \in P$ holds.

**Definition 1 (Visibility polygon)** *Given a polyon $P$, and a connected point set $X \in P$, the visibility polygon $Vis_P(X)$ is the set of points inside $P$ that are visible from any point $x \in X$.*

In order to compute $Vis_P(C_P)$ we first determine the shortest path $\pi_{p_1,p_k} = (r_1, \ldots, r_l)$ in $P$ from $p_1$ to $p_k$ which can be done in linear time ([6], [2]). Replacing $(p_1, \ldots, p_k)$ by $\pi_{p_1,p_k}$, we obtain a new polygon $R = (r_1, \ldots, r_l, p_{k+1}, \ldots p_n, r_1)$. Since the number of vertices of $\pi_{p_1,p_k}$ is at most $n$, the number of vertices of $R$ is at most $2n \in O(n)$. For short, we define $C_r = (r_1, \ldots, r_l)$ and $C_l = (r_l, p_{k+1}, \ldots, p_n, r_1)$.

It can be shown that for computing $Vis_P(C_P)$ it suffices to compute $Vis_R(C_r)$.

*Department of Computer Science I, RFW University Bonn, Institut für Informatik, Abt. I, {elmar.langetepe,penninge}@cs.uni-bonn.de

†HOCHTIEF ViCon GmbH, Alfredstrasse 236, D-45133 Essen

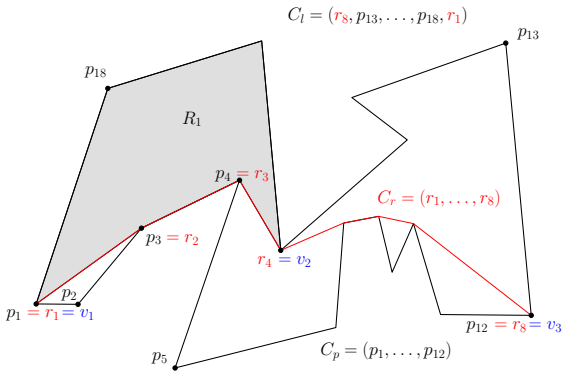


Figure 2: For computing the visibility polygon of a subchain $C_P = (p_1, \ldots, p_k)$ of $P$ it suffices to compute visibility polygons along the shortest path $\pi_{p_1,p_k}$.

**Lemma 1** *With the notation from above we have:*

$$Vis_P(C_P) = Vis_R(C_r) \cup (P \setminus R).$$

The remaining task is how to compute the visibility polygon $Vis_R(C_r)$ of the shorthest path $C_r$. Note that $R$ need not be a simple polygon because the shortest path $C_r$ may touch $C_l$, compare $r_4$ in Figure 2. We will make use of such *tangent points* and subdivide the polygon $R$ into several simple polygons $R_i$. The aim is to show that the union of the visibility polygons of the corresponding parts of $C_r$ is equal to $Vis_R(C_r)$.

Formally, the division into subpolygons is defined as follows: Because, originally, $C_r$ is a shortest path between two vertices of $P$, its set of vertices $V[C_r]$ also consists of the vertices of $P$. Let $(v_1, \ldots, v_t) = (V[C_l] \cap C_r)$ denote the vertices of $C_l$ that lie on $C_r$. The vertices $v_i$ lie consecutively on $C_r$. We have, by definition, $v_1 = r_1$ and $v_t = r_l$. We denote with $C_r(i) := (v_i, \ldots, v_{i+1})$ the polygonal chain on $C_r$ between $v_i$ and $v_{i+1}$. Since $C_r$ is a shortest path we conclude that $C_r(i)$ has to be an *outward bent chain*, i.e., while walking from $v_i$ to $v_{i+1}$ on $C_r(i)$ we always turn to the right at the vertices in between. Now let $C_l(i)$ denote the polygonal chain between $v_i$ and $v_{i+1}$ on $C_l$. We conclude that $R_i := C_r(i) \cup C_l(i)$ is a simple polygon (unless $V[R_i]$ consists of only the two vertices $v_i, v_{i+1}$), see Figure 2. The following Lemma states that we can obtain $Vis_R(C_r)$ by computing $Vis_{R_i}(C_r(i))$ for all $R_i$.

**Lemma 2** *With the notation from above we have:*

$$Vis_R(C_r) = \bigcup_{i=1}^{t-1} Vis_{R_i}(C_r(i)).$$

It remains to show how to compute the visibility polygon of an outward bent chain $C_r$ of the boundary of a simple polygon $R_i$. Let $C_l$ denote the remaining boundary of $R_i$, see Figure 3. The key idea of the algorithm of Guibas et al. [6] is the following: Roughly speaking, a point $p \in R_i$ is visible from an edge $e = (a, b)$ iff the two shortest paths $\pi_{a,p}$ and $\pi_{b,p}$ are outward convex[1]. The algorithm traverses two shortest path trees rooted at $a$ and $b$, and cuts off those points $p$ of which the two paths $\pi_{b,p}$ and $\pi_{a,p}$ are not outward convex. In our case the situation is slightly different. The visibility condition is no longer correct if we replace $e$ by a polygonal chain $C_r$ that is bent outwards. It is possible that the shortest paths begin with a part of $C_r$, and that the resulting paths are not outward convex, although $p$ is visible from $C_r$. This is fixed by ignoring the first vertices on the shortest paths that belong to $C_r$, thus adapting the test for outward convexity to the new situation. So the new algorithm does the same as before but ignores the first edges between vertices of $C_r$ on the shortest path trees when testing for outward convexity of the shortest paths. Figure 3 illustrates these observations. Point $z$ is not visible from $C_r$ because the paths $\pi_{r_2,z}$ and $\pi_{r_4,z}$ are not outward convex. Point $x$ is visible from $C_r$.

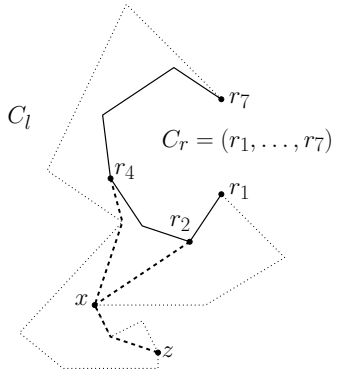


Figure 3: Shortest paths from $r_2$ and $r_4$ to $x$ and $z$.

Altogether we have:

**Lemma 3** *The visibility polygon of an outward bent subchain $C_r$ of the boundary of a polygon $R_i$ can be computed in linear time.*

Finally, using the previous three lemmata we are able to compute the inner visibility polygon of a boundary subchain $C_P$ of a simple polygon $P$ in linear time.

**Theorem 4** *Given a polygon $P$ and a subchain $C_P$ of $\partial P$, we can compute $Vis_P(C_P)$ in time $O(|P|)$.*

## 3 The visibility area of two boundary chains

The main tool for computing the visibility area between two polygons will be the computation of the

[1] A pair of paths $\pi_{a,p}$ and $\pi_{b,p}$ is outward convex iff the convex hull of each path lies outside the open region bounded by $\pi_{b,p} \cup \pi_{a,p} \cup |ab|$, where $|ab|$ denotes the line segment connecting $a$ and $b$.

visibility area between two chains of the boundary of a polygon.

**Definition 2 (Segment type)** *A segment $pq$ between two mutually visible points $p$ and $q$ is (of type) $PQ$, iff $p \in P$ and $q \in Q$, where $P$ and $Q$ are sets of points.*

**Definition 3 (Visibility area)** *Given two polygonal chains $P$ and $Q$, then the visibility area $VA(P,Q)$ consists of all segments of type $PQ$. If $P$ and $Q$ are subchains of the same polygon, we restrict $VA(P,Q)$ to segments inside the polygon.*

In a first step we would like to compute the visibility area between two subchains $A$ and $C$ on the boundary of a polygon $P$. Here, the computation of the visibility area between $A$ and $C$ can be reduced in a natural way to the computation of constrained visibility chains.

**Definition 4 (Visibility cut)** *A visibility cut is a boundary edge of a visibility polygon $Vis_P(X)$, that does not belong to $\partial P$.*

**Definition 5 (Constrained visibility chain)** *Given two subchains $A, C$ of a closed polygonal chain $P$, and the visibility polygon $Vis_P(A)$ of $A$, we define a polygonal chain $Vis_P(A)_{|C}$ as the union of all points $c \in C$ visible from $A$, and all visibility cuts connecting points of $C$ visible from $A$.*

First, we compute the visibility polygon $P_A = Vis_P(A)$. The boundary of $P_A$ contains the constrained visibility chain $C' = Vis_P(A)_{|C}$ of $C$ w.r.t. visibility from $A$. Then computing the visibility polygon $P_{AC}$ of $C'$ in $P_A$ we obtain all points of $P$ that are visible from both $A$ and $C$. Note that $P_{AC}$ contains $A' = Vis_P(C)_{|A}$ on its boundary, as well as $C'$. There are two chains $B, D$ connecting $A', C'$ on the boundary of $P_{AC}$. It can be shown that by replacing $B$ and $D$ with the shortest paths between their endpoints, we subtract all points from $P_{AC}$ that do not lie on a segment of type $AC$. Thus we have finally obtained the visibility area between $A$ and $C$.

**Lemma 5** *The visibility area between two subchains $A, C$ on the boundary of a polygon $P$ can be computed in time $O(|P|)$.*

**Proof.** The two visibility polygons $P_A$ and $P_{AC}$ can be computed in linear time by Theorem 4. After triangulating the resulting polygon $P_{AC}$, the two shortest paths can be computed in linear time, also. □

## 4 Two arbitrary closed polygonal chains $P$ and $Q$.

We now describe the most complex case where $P$ lies outside of $Q$, but lies completely inside the convex hull $ch(Q)$ of $Q$. The other cases can be done analogously.

Since $P$ is contained in $ch(Q)$, but is not contained in $Q$, $P$ is contained in a pocket of polygon $Q$. It suffices to compute the visibility area between $P$ and this pocket. The pocket can be identified in linear time by counting the intersections with a ray starting from a vertex of $P$ [7]. Let $pocket(Q)$ denote the part of $Q$'s boundary contributing to the pocket, and $e = (a, b)$ be the edge of $ch(Q)$ defining the pocket. From now on we denote with $\overline{Q}$ the closed polygonal chain $pocket(Q) \cup e$, so $P$ lies inside the bounded region defined by $\overline{Q}$. In order to compute $VA(P,Q)$ we do the following:

1. Compute $VA(P, \overline{Q})$.
2. Remove those points from $VA(P, \overline{Q})$ that only lie on segments of type $Pe$ (but not on a segment of type $PQ$).

### 4.1 Compute $VA(P, \overline{Q})$

We consider the polygon $R = \overline{Q} \setminus Int(P)$. If $R$ were a simple polygon we knew, by Lemma 5, how to compute $VA(P, \overline{Q})$, which is contained in $R$. But $P$ defines a hole in $R$, so $R$ is not a simple polygon. But there exists a pair of edges $e_l, e_r$ of type $PQ$ on the convex hull of $Q$ relative[2] to $P$, which can be identified in linear time and have a nice property: No segment of type $P\overline{Q}$ intersects *both* $e_l$ and $e_r$.

Inserting two copies of $e_x$ into $R$, where $x = l$ or $x = r$, we obtain a simple polygon $R_x$, of which $P$ and $\overline{Q}$ are boundary chains, connected by the two copies of $e_x$. We compute $VA_x = VA(P, \overline{Q})$ in $R_x$. Now by the property of the edges $e_l$ and $e_r$ we have $VA(P, \overline{Q}) = VA_l \cup VA_r$. We unite $VA_l$ and $VA_r$ as follows: The boundary of $VA_x$ consists of parts of the boundary of $P$, of $\overline{Q}$, and of visibility cuts of type $PP$ or $\overline{QQ}$. We compute the set of cuts for both $VA_x$ and cut off the parts of $R$ that lie behind a visibilty cut from $VA_l$ and from $VA_r$. Traversing the boundary of $R$ once, this can be done in linear time.

### 4.2 Remove points from $VA(P, \overline{Q})$

Our goal is, after computing $VA(P, \overline{Q})$, to remove the points of $VA(P, \overline{Q})$ that do not belong to $VA(P,Q)$. Only some special points have to be removed: The points $p$ that see a point of $r \in Int(e)$ are candidates for subtraction. Suppose we look from $p$ to $r$. Then $p$ has to be subtracted iff turning clockwise until seeing $r$ again

1. we have only seen points of $P$ and of $e$, or

[2] See Toussaint [9] for details.

2. we have at some point seen a point of $Q$, then a point of $P$, before then again seeing a point of $Q$, *and* there is no segment of type $PQ$ containing $p$.

The points of type 1 are not seen from $Q$ – they lie in some sort of pocket of $P$. Those of type 2 are the points which lie close to edge $e$, and thus do not lie on a segment of type $PQ$, but on segments of type $Pe$. See Figure 4 for an example.

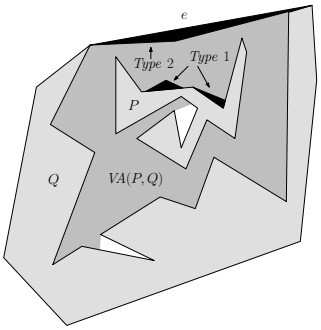


Figure 4: Points of type 1 and of type 2 that do not belong to $VA(P,Q)$, but to $VA(P,\overline{Q})$.

**Lemma 6** *The points of type 1 can be subtracted from $VA(P,\overline{Q})$ in linear time.*

**Proof.** Two subchains of the boundary of the visibility region of $e$ in $VA(P,\overline{Q})$ stem from the polygon $Q$. Computing the visibility polygons in $VA(P,\overline{Q})$ of those subchains we obtain the cuts defining the pockets containing the points of type 1, in linear time. As above, we only cut off points that lie behind cuts stemming from both subchains. □

**Lemma 7** *The points of type 2 can be subtracted from $VA(P,\overline{Q})$ in linear time.*

The above observations prove our main Theorem:

**Theorem 8** *The visibility area $VA(P,Q)$ between two polygons $P$ and $Q$ can be computed in time $O(|P|+|Q|)$.*

## 5 Concluding remarks

The algorithms for closed polygonal chains also apply to open polygonal chains.

Given a polygon $P$ and an obstacle polygon $Q$ we can compute $Vis(P)$ w.r.t. $Q$ – see Figure 5 – analogously to $VA(P,Q)$, with the following modifications: First, points not visible from $Q$ need not be subtracted from the polygon $R$ between $P$ and $Q$. Second, if $Vis(P)$ is unbounded, we need to replace parts of the boundary of $VA(P,Q)$ with adequate visibility cuts of infinite length.

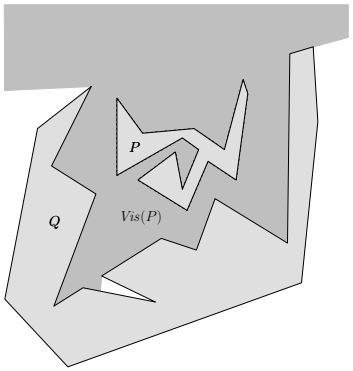


Figure 5: The visibility region of polygon $P$ with obstacle polygon $Q$.

## References

[1] T. Asano, S. K. Ghosh, and T. C. Shermer. Visibility in the plane. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 829–876. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[2] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.

[3] D. P. Dobkin and S. Teller. Computer graphics. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 42, pages 779–796. CRC Press LLC, Boca Raton, FL, 1997.

[4] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.

[5] S. K. Gosh. *Visisbility Algorithms in the plane*. Cambridge University Press, 2006.

[6] 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.

[7] R. Klein. *Algorithmische Geometrie*. Addison-Wesley, Bonn, 1997.

[8] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[9] G. Toussaint. On separating two simple polygons by a single translation. *Discrete Comput. Geom.*, 4:265–278, 1989.

[10] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE MELECON '83*, pages A10.02/1–4, 1983.