α-Visibility*

Mohammad Ghodsi[1,3,**], Anil Maheshwari[2], Mostafa Nouri[1], Jörg-Rüdiger Sack[2], and Hamid Zarrabi-Zadeh[1]

[1] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
{ghodsi,zarrabi}@sharif.edu, nourybay@ce.sharif.edu

[2] School of Computer Science, Carleton University, Ottawa, ON K1S 5B6, Canada
{anil,sack}@scs.carleton.ca

[3] Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

Abstract. We study a new class of visibility problems based on the notion of *α-visibility*. Given an angle α and a collection of line segments $\mathcal{S}$ in the plane, a segment t is said to be α-visible from a point p, if there exists an empty triangle with one vertex at p and the side opposite to p on t such that the angle at p is α. In this model of visibility, we study the classical variants of point visibility, weak and complete segment visibility, and the construction of the visibility graph. We also investigate the natural query versions of these problems, when α is either fixed or specified at query time.

1 Introduction

The study of visibility is at least 99 years old, when in 1913 Brunn [5] proved a theorem about the kernel of a set. By now, visibility has become one of the most studied notions in computational geometry. The reasons are two-fold: 1) such problems arise naturally in areas where computational geometry tools and algorithms find applications. 2) their solutions are required, or serve as building blocks in the development of solutions to other problems, such as motion planning problems. Many natural problem instances arise and have been extensively studied in two and higher dimensions. The reader is referred to [3,8].

Previous Work: Given a polygonal scene $\mathcal{S}$, the *visibility polygon* of a point p, denoted by $\mathrm{VP}(p)$, is the set of all points inside the scene that are visible from p. When the scene is a simple polygon or a polygonal domain, several algorithms exist to compute the visibility polygon of a point with/without preprocessing. Previous results for point-visibility inside a scene are summarized in Table 1.

Given a segment s, the *weak visibility polygon* $\mathrm{VP}(s)$ of s is the set of points in the scene that are visible from at least one point on s. Guibas *et al.* [11] showed how to compute the weak visibility polygon of a segment inside a simple polygon in $O(n)$ time. Suri and O'Rourke [22] established that the weak visibility polygon of a segment inside a polygon with holes has size $\Theta(n^4)$, but can be represented

* Research supported by NSERC, HPCVL and SUN Microsystems.
** This author's research was partially supported by IPM under grant No: CS1389-2-01.

F.V. Fomin and P. Kaski (Eds.): SWAT 2012, LNCS 7357, pp. 1–12, 2012.

Table 1. Summary of the previous work on point-visibility. Here, n is the total complexity of the scene, h is the number of holes, m_p is the complexity of VP(p) for a query point p, and $|E|$ is the number of edges in the visibility graph of the scene.

Scene	Prep. Time	Space	Query Time	Ref.
polygon	—	$O(n)$	$O(n)$	[3],[7],[15]
polygon	$O(n^3 \log n)$	$O(n^3)$	$O(\log n + m_p)$	[4]
polygon	$O(n^3)$	$O(n^3)$	$O(\log n + m_p)$	[12]
polygon	$O(n^2 \log n)$	$O(n^2)$	$O(\log^2 n + m_p)$	[1]
polygonal domain	—	$O(n)$	$O(n \log n)$	[22]
polygonal domain	—	$O(n)$	$O(n + h \log h)$	[13]
polygonal domain	$O(n^2)$	$O(n^2)$	$O(n)$	[2]
polygonal domain	$O(n^2 \log n)$	$O(n^2)$	$O(m_p \log(n/m_p))$	[23]
polygonal domain	$O(n^3 \log n)$	$O(n^3)$	$O(\min\{h, m_p\} \log n + m_p)$	[24]
convex polygons	$O(n \log n + \|E\|)$	$O(\|E\|)$	$O(m_p \log n)$	[20]

by a set of $O(n^2)$ triangles. They also gave an algorithm for computing the weak visibility polygon of a segment inside a polygon with holes in $O(n^4)$ time.

The *visibility graph* of a polygon is the undirected graph of the visibility relation on the vertices of the polygon. Optimal algorithms for computing visibility graphs exist. Ghosh and Mount [9] established its construction in $O(n \log n + |E|)$ time for a polygon with holes. Here, $|E|$ is the number of edges in the resulting visibility graph.

The *weak visibility graph* of a set of segments is defined as the graph, with a node for each segment and an edge between any pair of weak visible segments, that have at least two mutual visible points. Ghosh and Mount [9] and Keil *et al.* [14] computed the weak visibility graph in $O(n \log n + |E|)$ time. Nouri *et al.* [19] demonstrated how to detect the visibility relation between two query segments in $O(n^{1+\varepsilon})$ time, using $O(n^2)$ space and $O(n^{2+\varepsilon})$ preprocessing time, for any fixed ε. Gudmundsson and Morin [10] obtained similar results for testing weak visibility relation between a query point and a segment.

New Model: Let $\mathcal{S}$ be a set of n line segments in the plane, which are non-intersecting except possibly at their end-points. Since each polygonal scene is composed of a set of segments, $\mathcal{S}$ can model polygonal scenes as well. Let α be a positive real number. In this paper, we study a new class of visibility problems based on the notion of *α-visibility* as follows:

Point-visibility: A segment $t \in \mathcal{S}$ is said to be α-visible from a point p, if p can see t with an angle at least α; that is, if there exists an empty triangle with one vertex at p and side opposite to p on t such that the angle at p equals α (Fig. 1.a).
Segment-visibility: A segment t is said to be *weakly α-visible* from a segment s, if there is a point on s from which t is α-visible (Fig. 1.b). A segment t is said to be *completely α-visible* from s, if for all points on s, t is α-visible (Fig. 1.c).
Visibility Graph: We define the *weak* (respectively *complete*) *α-visibility graph* of $\mathcal{S}$ as a directed graph G_α whose vertices are the segments of $\mathcal{S}$, and for any two

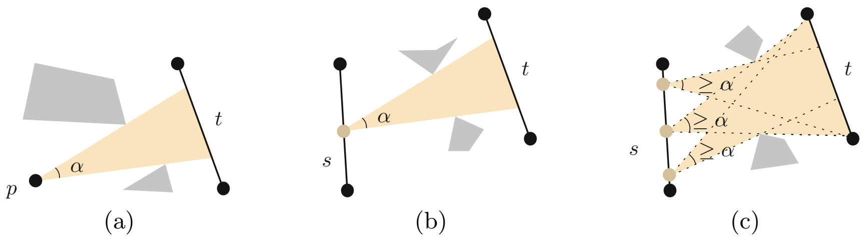


Fig. 1. (a) Segment t is α-visible from p. (b) Segment t is weakly α-visible from segment s. (c) Segment t is completely α-visible from segment s.

segments $s, t \in \mathcal{S}$, there is a directed edge from s to t if t is weakly (respectively, completely) α-visible from s.

The notion of *α-visibility* appears to be natural. Typically, all optical/digital imaging devices have limitations, quantified by their resolutions. Our α-visibility model is capable of capturing this limitation, and provides a more realistic alternative to the classical visibility models studied in the literature. The value α could be also employed to approximate the inaccuracy of a device used to provide visibility-related measurements. E.g., laser rangefinders do not return any data when they are too far off in angle from the surface normal. In general, there is a wealth of literature on approximation algorithms for geometric shortest path problems, and there is a close connection between visibility and shortest path problems, but still there are essentially no results on the notion of approximate visibility. This paper lays down a foundation in that respect, and will likely inspire further study of the notion of approximate visibility.

Our Results: In this paper, we present the first results for several variants of the point/segment visibility problems in the α-visibility model. The main idea that we use is to group the set of all possible visibility directions into $O(1/\alpha)$ directions, provide insights into α-visibility, and then employ a combination of geometric tools, such as trapezoid diagrams, shortest path maps, ray shooting, range searching, etc., to solve the visibility problem in each group of directions. In the following, $\mathcal{S}$ denotes the scene, i.e., the set of input segments in the plane.

In Section 3, we present efficient data structures that enable answering queries of the form "Is segment $t \in \mathcal{S}$ α-visible from a query point p in the plane?". When α is fixed, we preprocess $\mathcal{S}$ in $O(n \log n)$ time into a data structure of size $O(n)$ that answers aforementioned point-visibility queries in $O(\log n)$ time[1]. We also provide data structures for answering point-visibility queries when α is specified at the query time.

In Section 4, we show that the (weak and complete) α-visibility graph of $\mathcal{S}$ has linear size (in n), and that it can be computed in $O(n \log n)$ time. Once the graph has been computed in $O(n \log n)$ time, queries of the form "Is $t \in \mathcal{S}$ weakly/completely α-visible from $s \in \mathcal{S}$?" can be answered in $O(1)$ time. Then,

[1] The running times and space bounds of the data structures presented in this paper involve a factor $1/\alpha$, which is omitted when α is assumed to be a fixed constant.

we show how to preprocess $\mathcal{S}$ in $O(n \log n)$ time into a data structure of size $O(n)$ such that queries of the form "Is segment $t \in \mathcal{S}$ weakly α-visible from a query segment s in the plane?" can be answered in $O(\log n)$ time.

Note that one of the key differences between standard visibility (i.e., when $\alpha = 0$) and α-visibility lies in the size of the weak/complete visibility graph of the line segments in the plane. While the former has quadratic size, the latter is linear in size, which makes it appealing both theoretically and from an applied perspective especially when dealing with large data sets.

2 Preliminaries

Ray Shooting: A typical range shooting problem in the plane has the following form: given a set of n segments in the plane, build a data structure that, for any query ray r, report the first segment intersected by r quickly.

Theorem 1 (Chan [6]). *Given a set of n line segments in the plane, there is a data structure requiring $O(n \log^3 n)$ preprocessing time and $O(n \log^2 n)$ space, such that we can find the first point of intersection between a query ray and the set in $O(\sqrt{n} \log^2 n)$ expected time.*

Splinegons (or informally curved polygons) are defined as generalizations of polygons [21]. A splinegon S is formed from a polygon P by replacing one or more edges of P with curved edges such that the region bounded by each curved edge and the segment joining its end-points is convex.

Theorem 2 (Melissaratos and Souvaine [17]). *Given a simple splinegon S with n edges, there is a data structure requiring $O(n)$ preprocessing time and $O(n)$ space, such that, for any query point p and a ray $\mathbf{r}$ emanating from p, the first intersection of $\mathbf{r}$ with the splinegon can be reported in $O(\log n)$ time.*

Simplified Trapezoidal Diagram: Given a set $\mathcal{S}$ of segments in the plane and a direction d, we define a subdivision of the plane such that, each region in the subdivision is the maximal region with the property that all points in that region see the same segment in direction d. We can construct this subdivision by drawing a line in the reverse direction of d, from each end-point of all the segments of $\mathcal{S}$, until it meets another segment. We call this subdivision, the *simplified trapezoidal diagram* of $\mathcal{S}$ in direction d, and denote it by $\mathcal{T}_d(\mathcal{S})$.

Theorem 3. *Given a set $\mathcal{S}$ of n segments in the plane and a direction d, we can construct the simplified trapezoidal diagram $\mathcal{T}_d(\mathcal{S})$ by a plane-sweep in the direction perpendicular to d in $O(n \log n)$ time and $O(n)$ space.*

Shortest Path Maps: For a point s in a simple polygon P, the *shortest path map*, $SPM(s)$, is a partition of P into cells such that for all points t in a cell, the sequence of vertices of P along the shortest path from s to t is fixed. It is well-known that the complexity of $SPM(s)$ is $O(n)$ and it can be built in $O(n)$ time, where n is the number of vertices in P. If we preprocess $SPM(s)$ for point location, for a query point p we can find in $O(\log n)$ time, the last vertex of P in the shortest path from s to p, which can be thought of as the root associated to the cell containing p.

3 Point Visibility

In this section, we show how to build data structures that efficiently determine, for a query point p in the plane and a query segment $t \in \mathcal{S}$, whether t is α-visible from p. First, let $\alpha > 0$ be a fixed constant.

Visibility testing for fixed α:

Theorem 4. *We can preprocess $\mathcal{S}$ into a data structure of size $O(n)$ in $O(n \log n)$ time, such that α-visibility testing can be carried out in $O(\log n)$ time.*

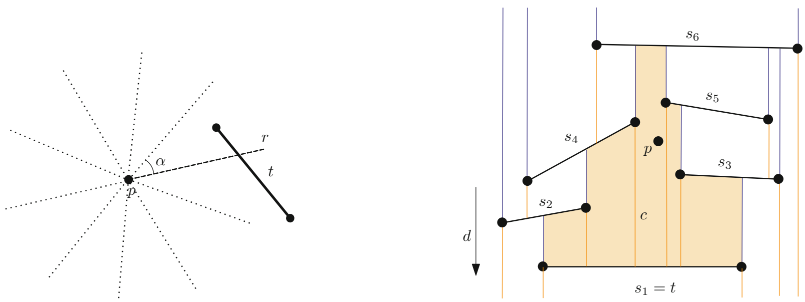


Fig. 2. (a) Segment t is α-visible from p, and r intersects t. (b) A trapezoidation of a set of segments in direction d.

Proof. Assume that we have a set of $\lceil \frac{2\pi}{\alpha} \rceil$ rays emanating from p, as in Fig. 2a, so that the angle between any two consecutive rays is α (except possibly between a pair, where it is $\leqslant \alpha$). Let D denote the set of directions of these rays. If t is visible from p with an angle at least α, t must intersect one of the rays drawn from p as in Fig. 2a. Let r be a ray that intersects t and let d be the direction of r. Consider the trapezoidal diagram of $\mathcal{S}$ in direction d, as in Fig. 2b. Observe that p lies inside the trapezoid that sees t in direction d. Therefore, if t is α-visible from p, p must be inside a trapezoid that sees t, in the trapezoidal diagram drawn for $\mathcal{S}$, based on directions in D.

It remains only to be checked whether p sees t with an angle of at least α. Consider the simplified trapezoidal diagram $\mathcal{T}_d(\mathcal{S})$ in direction d. Note that p is inside a region c, whose visible segment in direction d is t (see Fig. 2b)[2]. For a point p in c, the shortest paths from p to the end-points of t, inside c, consist of two convex chains. The maximum visible part of t from p, including the intersection point of r with t, is determined by extending the first edge of each of the shortest paths. Therefore, to compute the maximum visible part of t from p, it is sufficient to find the first turning points on the shortest paths from p to the end-points of t, in c.

[2] Note that the region c is essentially a simple polygon.

The complexity of the trapezoidal map in direction d is $O(n)$ and can be computed in $O(n \log n)$ time. We can use the trapezoidal map to locate the trapezoid containing p and find the visible segment in direction d. The shortest path map of the end-points of each segment t, in the corresponding cell c, has complexity proportional to the size of c. Since, the sum of the complexities of cells in the simplified trapezoidal map is $O(n)$, all the shortest path maps can be computed in $O(n \log n)$ time using $O(n)$ space. We repeat this construction for all directions of D. To answer a query, for each direction d, we locate the trapezoid in which p lies, in the trapezoidal map of $\mathcal{S}$. This can be done in $O(\log n)$ time. The trapezoid gives us a segment that is visible in direction d. If the segment is not t we proceed to the next direction. Otherwise, based on the shortest path map associated to the cell of the simplified trapezoidal map, we can find the maximum portion of t that is visible from p, and check if that portion forms an angle of at least α with t. If so, we report "yes", otherwise we check the next direction. The first turning point in the shortest path from p to the end-points of t can be located in $O(\log n)$ time, and since the number of directions is $O(1/\alpha)$ (a constant), the total query time is $O(\log n)$. □

Visibility Testing for Non-fixed, Constant α: Our objective here is to build a data structure, such that given a query point p, and a query segment $t \in \mathcal{S}$ and an angle $\alpha > 0$, we need to determine if t is α-visible from p.

Theorem 5. *We can preprocess $\mathcal{S}$, in $O(n \log^3 n)$ time, into a data structure of size $O(n \log^2 n)$, such that we are able to detect α-visibility of query segment $t \in \mathcal{S}$ from a query point p in $O(\sqrt{n} \log^3 n)$ expected time.*

Proof. Assume that we have a set of $\lceil \frac{2\pi}{\alpha} \rceil$ rays emanating from p, so that the angle between any two consecutive rays is at most α (Fig. 2a). If t is α-visible from p, then it is visible from p along at least one of these rays. Let r be such a ray. To check if the visible part around the intersection point of r with t constitutes an angle $\geqslant \alpha$, we need to find the maximum visible part of t from p around that intersection point. If the visible part forms an angle $\geqslant \alpha$, we answer to the visibility query affirmatively. Therefore, we need to solve the following two sub-problems: *i*) What is visible from p along r? *ii*) If we rotate r around p, when does the visibility from p along r change?

Problem (*i*) is ray shooting among segments, which has already been discussed in Theorem 1. We use range searching to solve problem (*ii*) as follows. Preprocess the end-points of segments in $\mathcal{S}$, for the two level half-plane range searching problem, and for each canonical subset of the result, we compute the convex hull of its points. Without loss of generality, assume that we want to find the first visibility change when we rotate r counterclockwise. The result has $O(\sqrt{n} \log^2 n)$ canonical sets, and for each canonical set we have pre-computed the convex hull of its points. Hence, for each canonical set, we can find the first point visited from that set, while we rotate r counterclockwise around p. The final result is the point that is visited first among all such points. Therefore, it can be found in $O(\sqrt{n} \log^3 n)$ expected time. The total complexity is now derived from the complexities of the two sub-problems. □

Remark: Using techniques introduced by [16], one can achieve a space/query-time tradeoff for the problem. Using $O(m)$ space, for any $n \log^2 n \leqslant m \leqslant n^2$, one obtains a query time of $O((n/\sqrt{m}) \operatorname{polylog} m)$.

4 Segment Visibility

The weak α-visibility graph, G_α, for $\mathcal{S}$ is defined as follows. Each segment of $\mathcal{S}$ is associated to a unique vertex in G_α. Furthermore, for any two segments $s, t \in \mathcal{S}$, if t is weakly α-visible from s, then there is a directed edge in G_α from the vertex corresponding to s to the vertex corresponding to t.

Lemma 1. *The weak α-visibility graph G_α of $\mathcal{S}$ has linear size.*

Proof. Fix a set of $O(1/\alpha)$ directions, D, such that the angle between any two adjacent directions is at most α. Assume that $t \in \mathcal{S}$ is weakly α-visible from $s \in \mathcal{S}$. Then, there is a point $p \in s$ that sees t with an angle at least α. Observe that there exists a direction $d \in D$ that is inside the angle of view from p. Let r be the segment connecting p to t in direction d. Clearly, r does not intersect any other segment of $\mathcal{S}$. Now slide r in the direction perpendicular to d, while it still connects s to t, until r meets an end-point of a segment in $\mathcal{S}$. One of two cases may arise: (i) r reaches an end-point of s or t, or (ii) r reaches an end-point of a segment other than s or t. In the first case, an end-point of s sees t in direction d, or an end-point of t sees s in the direction opposite to d. In the second case, a segment exists in $\mathcal{S}$ for which one of its end points sees t in direction d. Hence, segment-to-segment weak α-visibility can be mapped to a *unique* point-to-segment α-visibility. The number of directions is constant, and the number of end-points of segments in $\mathcal{S}$ is $O(n)$. The total number of distinct segment pairs (s, t), such that t is weakly α-visible from s, is $O(n)$. □

Theorem 6. *$\mathcal{S}$ can be preprocessed into a data structure of size $O(n)$ in $O(n \log n)$ time, so that weak α-visibility testing for any two query segments $s, t \in \mathcal{S}$ can be carried out in $O(1)$ time.*

Proof. Firstly, we compute G_α. Recall the set-up in the proof of Lemma 1. Assume that $t \in \mathcal{S}$ is weakly α-visible from $s \in \mathcal{S}$ and $p \in s$ sees t with an angle of at least α. Let d be the direction in D, that is inside the angle of view of p. Assume that the segment r connects p to t in direction d. Now slide r without changing its direction until it meets an end-point of a segment. The region so swept is a strip w,in which every point on s, on one side of w, sees a point on t, on the opposite side of w, in direction d. Thus, w is a trapezoid in the trapezoidal map of $\mathcal{S}$, in direction d. We denote the nodes associated with s and t in the weak α-visibility graph by s^* and t^*, respectively. Therefore, the first condition that must be met for an edge from s^* to t^* is that s and t are two facing (i.e., opposite) edges of a trapezoid in one of the trapezoidal maps constructed for directions in D.

While this condition is necessary, it is not sufficient. To check whether t is actually weakly α-visible from s, we need to find a point on s that can see t with

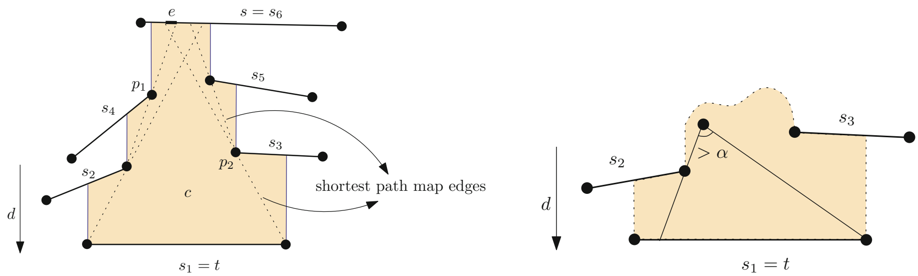


Fig. 3. (a) Segment s is partitioned into sub-segments. (b) The α-visibility region in direction d for $t \in \mathcal{S}$ is shaded.

an angle $\geqslant \alpha$. We do so by partitioning s into sub-segments, such that for each sub-segment, the shortest paths are combinatorially identical (i.e., the shortest paths to the end-points of t have the same set of turning points). This can be done by computing the shortest path map of the endpoints of t inside its adjacent region in the simplified trapezoidal map, as in the previous section, and finding the intersection points of s with the edges of the shortest path maps. Assume that s is partitioned into such subsegments. Let e be one of the subsegments (see Fig. 3a). We now want to determine the point on e which has the greatest view angle to t. Since all points on e have the same combinatorial shortest paths, the largest angle of view from each point on e towards t is determined by the two fixed points. These are the first turning points, say p_1 and p_2, in the shortest paths from any point in e to the end-points of t. The problem thus reduces to finding a point on e with the maximum view through p_1 and p_2. This point is on the intersection of the smallest circle through p_1 and p_2 that intersects e. Therefore, if the circle through p_1 and p_2 which is tangent to the supporting line of e, is incident on the segment e itself, then that supporting point on e will have the largest view. Otherwise, it will be one of the end-points of e. If the view angle is $\geqslant \alpha$, then add the edge from s^* to t^* in G_α. The above procedure is repeated for each sub-segment of s.

The complexity of shortest path map is $O(n)$ and each edge in the map can intersect at most one segment from $\mathcal{S}$. Therefore, the total number of sub-segments is $O(n)$. For each sub-segment, the first turning points on the shortest paths to the end-points of a visible segment can be found in $O(\log n)$ time. Given the turning points, the point with the greatest angle of view can be determined in $O(1)$ time. Therefore, G_α can be computed in $O(n \log n)$ time. For each direction $d \in D$, let $G_d = (V, E_d)$ be the subgraph of G_α with only edges $(s^*, t^*) \in E$ such that s and t are two facing edges in $\mathcal{T}_d$. Obviously G_d is planar and can thus be stored using $O(n)$ space. Checking whether a pair of vertices is connected by an edge takes constant time [18]. In order to determine if t is weakly α-visible from s, we need to verify the existence of edge (s^*, t^*) in G_d for each $d \in D$. This requires $O(1/\alpha)$ time, which is a constant. $\square$

4.1 One Arbitrary Query Segment

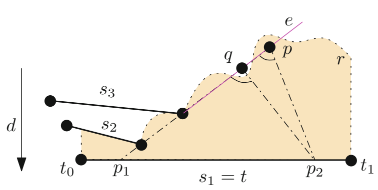
Next, we want to build a data structure, such that given two query segments $s \notin \mathcal{S}$ and $t \in \mathcal{S}$, we can determine if t is weakly α-visible from s. As part of the preprocessing, we compute, for each segment $t \in \mathcal{S}$, the set of all points in the plane from which t is α-visible. If t is weakly α-visible from s, then s must have a point in this set. First, we define the *α-visibility region in direction d for t*, as the region containing the points from which t is α-visible and d is inside the angle of view (see, Fig. 3b).

Lemma 2. *For any direction d and the segment t, the boundary of the α-visibility region, say r_t, in direction d for t consists of two monotone curves with respect to the direction perpendicular to d.*

Corollary 1. *The α-visibility region for any $t \in \mathcal{S}$ and for any direction d, does not contain a hole.*

Lemma 3. *The total complexity of the α-visibility regions, in direction d, for all the segments in $\mathcal{S}$, is linear.*

Proof. Let c denote the cell in $\mathcal{T}_d$ associated to t, and let r_t denote the α-visibility region in direction d for t. Obviously, $r_t \subseteq c$. Construct shortest path maps from the end-points of t inside c. Our first claim is that the boundary of r_t intersects each edge of the shortest path map at most once. By contradiction, as shown in Fig. 4a, assume that for an end-point of t, say t_0, there is an edge e in the shortest path map of t_0, such that the boundary of r_t intersects e at least twice. We can choose two points $p, q \in e$, such that $p \in r_t$ and $q \notin r_t$ and $\overrightarrow{pq}$ points towards t. Let p_1 and p_2 be the two extreme points on t visible from p through its angle of view. Let p_1 be closer to t_0 than p_2. Because p and q are on an edge of the shortest path map of t_0, both are visible from p_1. Moreover, p_1 is defined by the intersection of t and the supporting line of e. We know that $\angle p_1 p p_2 \geqslant \alpha$. Now observe that $\angle p_1 q p_2$ is greater than $\angle p_1 p p_2$ and is empty. Therefore, q can see t with angle $\geqslant \alpha$, and because $q \in c$, it is also in r_t, which contradicts the assumption that $q \notin r_t$. Therefore, the boundary of r_t intersects each edge of the shortest path map at most once.



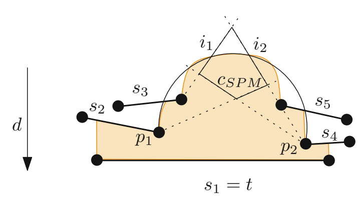


Fig. 4. (a) Points p and q are two points on e. p is in the α-visibility region in direction d for segment t while q is outside the region. (b) Points i_1 and i_2 are two consecutive points which are the intersections of the α-visibility region, in direction d for the segment t, with c_{SPM}.

Consider next the example shown in Fig. 4b. Let i_1 and i_2 be two consecutive intersection points of the boundary of r_t with the two shortest path maps of the end-points of t. Then, i_1 and i_2 are two points on the boundary of a cell, c_{SPM}, in the overlay of the two shortest path maps. Our second claim is that the boundary of r_t between i_1 and i_2 has complexity proportional to the size of c_{SPM}. This holds because there is a combinatorially unique shortest path in c_{SPM} towards the end-points of t. Therefore, for all points on the boundary of r_t, between i_1 and i_2, the largest view to t is determined by two fixed points, say p_1 and p_2. The set of all points which can see t through p_1 and p_2, with an angle $\geqslant \alpha$, are inside, or on the boundary of, the circle through p_1, p_2 having inscribed angle α lying on p_1p_2 clipped by segment p_1p_2. Hence, the boundary of r_t between i_1 and i_2 is determined by the intersection of that circle with c_{SPM}. This intersection has complexity at most $2 * |c_{SPM}|$ because any edge in c_{SPM} can be intersected by the circle at most twice. The complexity of r_t is equal to the number of intersections of its boundary with the shortest path maps and segments of $\mathcal{S}$. Thus, the size of r_t is $O(|c|)$. The total complexity of the cells in $\mathcal{T}_d$ is linear. Therefore, the total complexity of the α-visibility regions in direction d, for all the segments, is $O(n)$. □

Lemma 4. *The α-visibility region in direction d, for all segments in $\mathcal{S}$, can be computed in $O(n \log n)$ time using $O(n)$ space.*

Proof. Let r_t denote the α-visibility region in direction d for t. We first compute $\mathcal{T}_d$ for $\mathcal{S}$. For each segment $t \in \mathcal{S}$, we construct the two shortest path maps of the end-points of t in the cell c associated to t in $\mathcal{T}_d$. Let t_0 and t_1 be the end-points of t; let $SPM(t_0)$ and $SPM(t_1)$ denote the shortest path maps of the end-points, respectively. By definition, the end-points of t are in r_t. We compute the boundary of r_t, by traversing the boundary of c from t_0 until we reach t_1. The events are the intersection points of r_t with $SPM(t_0)$ and $SPM(t_1)$ and the boundary of c. In the first part of r_t, the shortest paths to t_0 and t_1 are two direct segments. The set of points with this property that can see t with angle at least α lies inside or on the boundary of a circle through t_0 and t_1 and on one side of t. We need to compute the intersections of this circle with c, $SPM(t_0)$ and $SPM(t_1)$. The first intersection point with $SPM(t_0)$ and $SPM(t_1)$ affects one of the shortest paths. For this shortest path, the last turning point is changed. Therefore, the circle needs to be updated so that it passes through this point and the last turning point of the other shortest path. After this update, we continue with updated paths in a similar manner until we reach t_1.

$\mathcal{T}_d$ can be computed in $O(n \log n)$ time. $SPM(t_0)$ and $SPM(t_1)$ can be computed in $O(n)$ time. Computing the first intersection point of the current circle with $SPM(t_0)$ and $SPM(t_1)$ takes time proportional to the complexity of the current cells in $SPM(t_0)$ and $SPM(t_1)$. This yields a total time of $O(|c|)$ for all circles. Computing the intersection points of each circle with c can also be done in $O(|c|)$ total time as well. Therefore, the α-visibility regions in direction d for all segments can be computed in $O(n \log n)$ time using $O(n)$ space. □

Theorem 7. *We can preprocess $\mathcal{S}$ into a data structure of size $O(n)$ in $O(n \log n)$ time, such that given two query segments $s \notin S, t \in \mathcal{S}$, their weak α-visibility can be tested in $O(\log n)$ time.*

Proof. As before, we first fix a set D of $O(1/\alpha)$ directions with the property that the angle between any two adjacent directions is at most α. If t is weakly α-visible from s, then there is a direction d in D, and a point q on s from which t is α-visible, and q can see t in direction d inside its angle of view. It is easy to see that q is in the α-visibility region in direction d for t. So the problem reduces to checking the intersection of s with each of the α-visibility regions computed for t, with respect to all directions in D.

We compute the α-visibility regions in all directions $d \in D$ for all segments $t \in \mathcal{S}$ and preprocess each region for ray shooting queries. An α-visibility region is a bounded, hole-free region and its boundary consists of straight line segments and circular arcs. Therefore, it is a splinegon and we can use Theorem 2 for ray shooting queries. Given two query segments $s \notin \mathcal{S}$ and $t \in \mathcal{S}$, we first find r_t (the α-visibility region in direction d for t). We need to know if s has any point in r_t. Let s_0 and s_1 be the end-points of s. We first check whether $s_0 \in r_t$, but because the ray shooting algorithm [17] has point location as a basis, we can use it and determine if $s_0 \in r_t$. If this is the case, t is α-visible from s_0 and weakly α-visible from s. If $s_0 \notin r_t$, shoot a ray in r_t (the splinegon) to find the first intersection point of the ray originating from s_0 in the direction towards s_1. If the intersection point is on s itself, then s intersects r_t and therefore t is weakly α-visible from s, otherwise it is not.

Computing the α-visibility regions for $\mathcal{S}$ takes $O(n \log n)$ time and $O(n)$ space. Preprocessing the α-visibility regions for ray shooting queries takes the same time and space. Point location and ray shooting queries can be performed in $O(\log n)$ time. This estalishes the bound claimed in the theorem. □

4.2 Complete α-Visibility

We want to build a data structure such that given two query segments $s, t \in \mathcal{S}$, we can determine if t is completely α-visible from s. A segment t is *completely α-visible from another segment s* if and only if t is α-visible from all points on s. Note that the complete α-visibility graph is a subgraph of the weak α-visibility graph, and hence its size is linear. We follow a scheme similar to that in Theorem 6. Here, we need to ensure that the collection of all the points on s that can see t, with respect to α-visibility, cover whole of s. For this, we employ shortest path maps of end-points of t. The details are similar to that in Theorem 6.

Theorem 8. *We can preprocess $\mathcal{S}$ into a data structure of size $O(n)$ in $O(n \log n)$ time, such that we can answer complete α-visibility testing queries in $O(1)$ time.*

References

1. Aronov, B., Guibas, L., Teichmann, M., Zhang, L.: Visibility queries and maintenance in simple polygons. Discrete Comput. Geom. 27(4), 461–483 (2002)
2. Asano, T., Asano, T., Guibas, L., Hershberger, J., Imai, H.: Visibility of disjoint polygons. Algorithmica 1(1), 49–63 (1986)

3. Asano, T., Ghosh, S., Shermer, T.: Visibility in Plane in the Handbook in Computational Geometry. Elsevier Science (1999)
4. Bose, P., Lubiw, A., Munro, J.: Efficient visibility queries in simple polygons. In: Proc. 4th Canadian Conf. Comput. Geom., pp. 23–28 (1992)
5. Brunn, H.: Über Kerneigebiete. Math. Ann. 73, 436–440 (1913)
6. Chan, T.M.: Optimal partition trees. In: Proceedings of the 2010 Annual Symposium on Computational Geometry, SoCG 2010, pp. 1–10. ACM, N.Y. (2010)
7. ElGindy, H.A., Avis, D.: A linear algorithm for computing the visibility polygon from a point. J. Algorithms 2(2), 186–197 (1981)
8. Ghosh, S.: Visibility Algorithms in the Plane. Cambridge University Press (2007)
9. Ghosh, S., Mount, D.: An output-sensitive algorithm for computing visibility. SIAM J. Comput. 20, 888–910 (1991)
10. Gudmundsson, J., Morin, P.: Planar visibility: testing and counting. In: Proceedings of the 2010 Annual Symposium on Computational Geometry. ACM (2010)
11. Guibas, L., Hershberger, J., Leven, D., Sharir, M., Tarjan, R.: Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. Algorithmica 2, 209–233 (1987)
12. Guibas, L., Motwani, R., Raghavan, P.: The robot localization problem. SIAM J. Comput. 26(4), 1120–1138 (1997)
13. Heffernan, P., Mitchell, J.: An optimal algorithm for computing visibility in the plane. SIAM J. Comput. 24(1), 184–201 (1995)
14. Keil, M., Mount, D., Wismath, S.: Visibility stabs and depth-first spiralling on line segments in output sensitive time. Int. J. Comput. Geometry Appl. 10(5), 535–552 (2000)
15. Lee, D.: Visibility of a simple polygon. Computer Vision, Graphics, and Image Processing 22(2), 207–221 (1983)
16. Matoušek, J.: Range searching with efficient hierarchical cuttings. Discrete & Computational Geometry 10, 157–182 (1993)
17. Melissaratos, E., Souvaine, D.: Shortest paths help solve geometric optimization problems in planar regions. SIAM Jl. Computing 21(4), 601–638 (1992)
18. Munro, J., Raman, V.: Succinct representation of balanced parentheses and static trees. SIAM J. Comput. 31, 762–776 (2002)
19. Nouri, M., Zarei, A., Ghodsi, M.: Weak Visibility of Two Objects in Planar Polygonal Scenes. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007, Part I. LNCS, vol. 4705, pp. 68–81. Springer, Heidelberg (2007)
20. Pocchiola, M., Vegter, G.: The visibility complex. Int. J. Comput. Geometry Appl. 6(3), 279–308 (1996)
21. Souvaine, D.: Computational geometry in a curved world (algorithm). PhD thesis. Princeton University, Princeton (1986), AAI8629439
22. Suri, S., O'Rourke, J.: Worst-case optimal algorithms for constructing visibility polygons with holes. In: Proceedings of the 1986 Annual Symposium on Computational Geometry, pp. 14–23 (1986)
23. Vegter, G.: The Visibility Diagram: A Data Structure for Visibility Problems and Motion Planning. In: Gilbert, J.R., Karlsson, R. (eds.) SWAT 1990. LNCS, vol. 447, pp. 97–110. Springer, Heidelberg (1990)
24. Zarei, A., Ghodsi, M.: Query point visibility computation in polygons with holes. Comput. Geom. Theory Appl. 39, 78–90 (2008)