# Visibility Maps of Segments and Triangles in 3D

*Esther Moet*

*Christian Knauer*

*Marc van Kreveld*

institute of information and computing sciences, utrecht university

technical report UU-CS-2005-049

www.cs.uu.nl

# Visibility Maps of Segments and Triangles in 3D*

Esther Moet[†], Christian Knauer[‡] and Marc van Kreveld[†]

December 2005

**Abstract**

Let $T$ be a set of $n$ triangles in three-dimensional space, let $s$ be a line segment, and let $t$ be a triangle, both disjoint from $T$. We consider the visibility map of $s$ with respect to $T$, i.e., the portions of $T$ that are visible from $s$. The visibility map of $t$ is defined analogously. We look at two different notions of visibility: strong (complete) visibility, and weak (partial) visibility. The trivial $\Omega(n^2)$ lower bound for the combinatorial complexity of the strong visibility map of both $s$ and $t$ is almost tight: we prove an $O(n^2 \log n)$ upper bound for both structures. Furthermore, we prove that the weak visibility map of $s$ has; complexity $\Theta(n^5)$, and the weak visibility map of $t$ has complexity $\Theta(n^7)$. If $T$ is a polyhedral terrain, the complexity of the weak visibility map is $\Omega(n^4)$ and $O(n^5)$, both for a segment and a triangle. We also present efficient algorithms to compute all discussed structures, i.e., in most cases computation is only an $O(\log n)$ factor worse than the worst-case complexity.

## 1 Introduction

Computations concerning visibility and the generation of shadows are important to obtain realistic images in computer graphics. Unfortunately, these computations can be very time-consuming. The *visibility map* of a point $p$ with respect to a set $T$ of $n$ disjoint triangles in 3D, which we call a *3D scene*, is the set of maximally connected components on $T$ that are visible from $p$. When projected onto a viewplane near $p$, the visibility map is a planar graph with $\Theta(n^2)$ vertices, edges, and faces [15, 17]. This graph is also referred to as the visibility map.

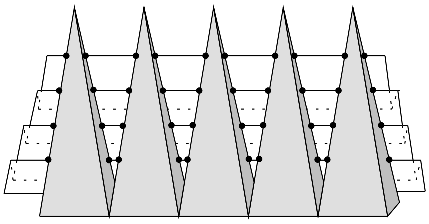

Figure 1: A visibility map of a point with $\Omega(n^2)$ vertices.

In this paper, we study the worst-case combinatorial complexity of the visibility map for a set $S$ that consists of more than a single point; in particular, we consider the cases where $S$ is a line segment or a triangle. The *strong visibility map* of $S$ identifies the maximally connected components on $T$ in which the points see *every* point of $S$, whereas the other points of $T$ do not

*This research was initiated during a visit of the first author to Freie Universität Berlin, which was supported by a Marie-Curie scholarship.

[†]Department of Information and Computing Sciences, Universiteit Utrecht, P.O.Box 80.089, 3508 TB Utrecht, the Netherlands. {marc,esther}@cs.uu.nl

[‡]Institute of Computer Science,Freie Universität Berlin, Takustraße 9, D-14195 Berlin, Germany. christian.knauer@inf.fu-berlin.de

see all of $S$, i.e., they see $S$ partly or not at all. The *weak visibility map* separates the points that see at least one point in $S$, and the points that do not see any point in $S$. An alternative way of looking at the weak visibility map is to consider $S$ to be a light source. Then the weak visibility map of $S$ is equivalent to the set of maximally connected components on $T$ that are illuminated; we will sometimes call the weak visibility map of $S$ the *shadow map* of $S$.

With the results of Wang and Zhu [24], the weak visibility map of a triangle in a general 3D scene can be computed in $O(n^9)$ time and $O(n^7)$ space; to compute the same structure in a polyhedral terrain, the algorithm of [24] can be used to achieve a running time of $O(n^7)$ time and $O(n^5)$ storage. In this paper, we improve significantly on these previous results and show that our bounds are almost tight by giving lower bound constructions. For a line (segment), a collection of general results can be found in [3]. Both of the above papers use the detection of transparent and opaque topology changes to compute the parts of a 3D scene that are weakly visible from a given view element; these topology changes are generally known as EEE events [12, 18].

Recently, an $O(n^2 \log n)$ time algorithm was presented to compute the weak visibility map of an edge in a terrain on another edge [6]. One of the main topics related to our study is the aspect graph [12, 14, 18]. This graph corresponds directly to the more frequently used *viewpoint space partition* of $\mathbb{R}^3$ into three-dimensional cells such that the visibility map with respect to the scene is combinatorially the same throughout each cell. The *critical points* where the view of the scene changes combinatorially lie exclusively on *critical surfaces* called reguli; see [12] and Sec. 2. The above mentioned spatial subdivision is induced by a set of $O(n^3)$ of these critical surfaces, and therefore has complexity $O(n^9)$. However, it has been shown that its complexity is almost a linear factor less for polyhedral terrains [1, 8].

Shadows are of major interest in computer graphics. The strong and weak visibility polyhedron of a view element are closely related to the *antiumbra* and *antipenumbra* of a light source, respectively. Many papers consider computing the shadow from a given point light source, either in image space, see [25], or in object space [5, 11]. The exact shadow of a segment or area light source is considered too complex to compute exactly. This may be the reason that all studies until now have been incomplete; some address a particular type of 3D scene, e.g., a sequence of convex areal holes [23], others give algorithms that yield an approximation of the shadow [5, 6]. The main contribution of our paper is that we show exactly how hard computing shadows in general 3D scenes can be.

Table 1 gives a summary of our results. All displayed results are worst-case combinatorial complexities. We also present algorithms to compute the discussed structures within almost matching time bounds, i.e., in most cases computation is only an $O(\log n)$ factor worse than the worst-case complexity.

| | strong | weak (terrain) | weak (general) | |
|---|---|---|---|---|
| point | $\Theta(n^2)$ | | | [15] |
| segment | $\Omega(n^2)$ | $\Omega(n^4)$ | $\Theta(n^5)$ | Section 3 & 4 |
| triangle | $O(n^2 \log n)$ | $O(n^5)$ | $\Theta(n^7)$ | Section 5 |

Table 1: Combinatorial complexities of the visibility map in a 3D scene for various types of view elements under different models of visibility.

This paper is organized as follows. In Section 2, we give formal definitions concerning the problem at hand. We address the strong and the weak visibility map of an edge in Section 3 and 4, respectively. In Section 5, study the complexity of the two types of visibility maps when the view element is a triangle, and we round up in Section 6 with the conclusions and a discussion of the obtained results.

## 2 Preliminaries

In this paper, we consider sets of triangles in $\mathbb{R}^3$ with pairwise disjoint interiors. We call such a set $T$ a *3D scene*, or simply *scene*. Without loss of generality, we assume that the vertical direction is the $z$-direction. A special case of a scene is a *polyhedral terrain*, or simply *terrain*, in which $T$ is a piecewise-linear continuous function defined over the triangles of a triangulation in the $xy$-plane. Given a scene $T$ and two points $p, q \in \mathbb{R}^3$, we say that $p$ *sees* $q$ if the open line segment $pq$ does not intersect the interior of any triangle in $T$, i.e., grazing contact of the line-of-sight with the edges of the triangles of $T$ is permitted. In the case of terrains, two points can only be mutually visible if they both lie on or above the terrain. All our results, both the bounds on the worst-case complexities and the algorithms, hold when $T$ contains degeneracies. However, to simplify argumentation, we only consider general position configurations in our proofs. This is no restriction; the worst-case complexities can be achieved by a 3D scene in general position, and in the presented algorithms, we can apply standard symbolic perturbation techniques [9].

To study visibility in $\mathbb{R}^3$, it is important to consider collections of lines that intersect three given lines. Such collections are called *reguli*. More formally, the regulus $\mathcal{R}(l_1, l_2, l_3)$ for three lines $l_1$, $l_2$, and $l_3$, is the set of all lines that intersect $l_1$, $l_2$, and $l_3$, in this (or the inverse) order. A regulus has the following properties [19, §3]:

1. it is one ruling of either a hyperbolic paraboloid, or a hyperboloid of one sheet;
2. it is an algebraic surface of constant degree;
3. the projection of its boundary onto the $xy$-plane is a set of algebraic curves of constant degree;
4. it can be partitioned into a constant number of $xy$-monotone parts.

Let $\ell(s)$ denote the supporting line of a line segment $s$. For three edges $e$, $f$, and $g$ of $T$, all lines that intersect $e$, $f$, and $g$, in this (or the inverse) order, lie in a subset of $\mathcal{R}(\ell(e), \ell(f), \ell(g))$. This set consists of at most three connected components [4], and is denoted by $\mathcal{R}(e, f, g)$. The surface $\mathcal{R}(e, f, g)$ can be computed in $O(1)$ time [4]. The critical points where the view of the scene changes combinatorially lie exclusively on these reguli [12].

Throughout this paper, we adopt an appropriate model of computation, i.e., we assume that computing various elementary operations on algebraic curves and surfaces of constant degree can be performed in constant time; for a discussion of computational real algebraic geometry, see for instance [13].

**Definition 1** *Let $T$ be a 3D scene, and let $p$ be a point in $\mathbb{R}^3$. We denote by $\mathrm{VP}(p, T)$ the visibility polyhedron of $p$ with respect to $T$, i.e., the connected subset of $\mathbb{R}^3$ that contains all points that see $p$. The subdivision of the triangles of $T$ induced by $\mathrm{VP}(p, T)$ is the visibility map of $p$ and is denoted by $\mathrm{VM}(p, T)$.*

**Definition 2** *Let $T$ be a 3D scene, and let $s$ be a low-dimensional simplex, for example, an edge or triangle of $T$. The* strong visibility polyhedron *of $s$, denoted by $\mathrm{SVP}(s, T)$, is the set of points in $\mathbb{R}^3$ that see $s$ completely. The subdivision of the triangles of $T$ induced by $\mathrm{SVP}(e, T)$ is the* strong visibility map *of $s$; we denote this structure by $\mathrm{SVM}(s, T)$.*

**Definition 3** *Let $T$ be a 3D scene, and let $s$ be a low-dimensional simplex, for example, an edge or triangle of $T$. The* weak visibility polyhedron *of $s$, denoted by $\mathrm{WVP}(s, T)$, is the set of points in $\mathbb{R}^3$ that see at least one point of $s$. The subdivision of the triangles of $T$ induced by $\mathrm{WVP}(e, T)$ is the* weak visibility map *of $s$; we denote this structure by $\mathrm{WVM}(s, T)$.*

In general, visibility maps consist of multiple connected components, whereas visibility polyhedra are connected sets.

Let $e$ be an edge of $T$, and let $t$ be a triangle of $T$. In this paper, we study the worst-case combinatorial complexities of the following four structures: $\mathrm{SVM}(e, T)$, $\mathrm{WVM}(e, T)$, $\mathrm{SVM}(t, T)$, and $\mathrm{WVM}(t, T)$. We assume that $e$ or $t$ is a feature of $T$, but this is no restriction; if an arbitrary segment or triangle (1- or 2-simplex) $s \subset \mathbb{R}^3$ does not intersect any triangle in $T$, and lies above $T$ in case $T$ is a polyhedral terrain, then we can transform $s$ into a feature $s'$ of a 3D scene $T'$ in $O(n \log n)$ time in such a way that the visibility map of $s$ in $T$ is essentially the same as the visibility map of $s'$ in $T'$.

As mentioned in the Introduction, the strong and weak visibility polyhedron are closely related to the *antiumbra* and *antipenumbra* of a light source, respectively, which are well-known structures in computer graphics. More precisely, the antiumbra is the volume from which all points on the light source can be seen, and the antipenumbra is the volume from which some, but not all, of the light source can be seen [23]. In this setting, the strong visibility polyhedron is equal to the antiumbra, and the weak visibility polyhedron is equal to the closure of the union of the antiumbra and antipenumbra.

A trivial lower bound for the complexity of all four types of visibility maps is $\Omega(n^2)$, which is a lower bound for the complexity of the visibility map of a single point [15].

**Definition 4** *[14] Let $T$ be a 3D scene. The* aspect graph *induced by, and restricted to, $T$ is the subdivision of the triangles of $\mathbb{R}^3$ into cells in which the perspective projection of $T$ is combinatorially the same.*

The points in one cell of the aspect graph all see exactly the same features of $T$; in particular, throughout each cell the simplex $s$ is either visible or invisible. Thus, the complexity of the aspect graph restricted to $T$ is an upper bound on both $\mathrm{SVM}(s, T)$ and $\mathrm{WVM}(s, T)$. On each of the $n$ triangles of $T$, the aspect graph is defined by the arrangement of $O(n^3)$ critical surfaces, which has complexity $O(n^6)$, and thus the overall complexity is $O(n^7)$. Hence, the complexity of each of the above defined visibility maps is $\Omega(n^2)$ and $O(n^7)$; these bounds also hold for visibility maps that we do not discuss here, e.g., the shadow map of a polyhedral light source.

## 3 The Strong Visibility Map of an Edge

The 2D counterpart of the strong visibility polyhedron of an edge in a 3D scene is the strong (or alternatively, complete) segment visibility polygon $\mathrm{SVP}(e, P)$ for an edge $e$ in a simple polygon $P$, where $P$ may have holes. This strucure has complexity $\Theta(n)$ [20, p837]. In fact, the strong visibility polygon of a segment $pq$ within $P$ is equal to the visibility polygon of the point $p$ *within* the visibility polygon of the point $q$ in $P$, i.e., $\mathrm{SVP}(pq, P) = \mathrm{VP}(p, \mathrm{VP}(q, P))$.

Before we prove a lower and upper bound on the complexity of $\mathrm{SVM}(e)$, we prove that the analog of the above holds in three dimensions as well.

**Lemma 5** *Let $T$ be a 3D scene $T$, and let $e = vw$ be an edge of $T$. The strong visibility polyhedron $\mathrm{SVP}(e, T)$ is equal to $\mathrm{VP}(v, \mathrm{VP}(w, T))$.*

Proof. First, we show that if $p \in \mathrm{SVP}(e, T)$ then $p \in \mathrm{VP}(v, \mathrm{VP}(w, T))$. The triangle $\Delta_{ep}$ with apex $p$ and base $e$ does not intersect any triangle of $T$. In particular, the line segment $vp$ does not intersect $T$, so $p \in \mathrm{VP}(v, T)$. Because the whole of $\Delta_{ep}$ is free from intersections with $T$, $v$ sees all points on the line segment $wp$ as well. This means that the line segment $wp \subset \mathrm{VP}(v, T)$ and thus that $w$ sees $p$ within $\mathrm{VP}(v, T)$.

Second, we show that if $p \in \mathrm{VP}(v, \mathrm{VP}(w, T))$ then $p \in \mathrm{SVP}(e, T)$. We have that $p$ sees both vertices $v$ and $w$, which means the line segments $vp$ and $wp$ are not intersected by $T$. Furthermore, both these line segments, and the edge $e$, lie in $\mathrm{VP}(v, T)$, which is a star-shaped polyhedron. This immediately implies that the triangle $\Delta_{ep}$ completely lies in $\mathrm{VP}(v, T)$. This means that $\Delta_{ep}$ is free from intersections with $T$, and thus $p \in \mathrm{SVP}(e, T)$. ⊔⊓

This analysis does not provide us with a good upper bound on the complexity of $\mathrm{SVP}(e, T)$, since the visibility polyhedron of a point in a 3D scene already has complexity $\Theta(n^2)$ in the worst case, and the visibility polyhedron of a point in a scene with $O(n^2)$ triangles has complexity $O(n^4)$. Therefore, we take a different approach.

In the remainder of this section, we are given a 3D scene $T$ with $n$ triangles and an edge $e$ of $T$, and we aim to bound the complexity of the strong visibility map $\mathrm{SVM}(e,T)$ of $e$ on $T$. A trivial lower bound for $\mathrm{SVM}(e,T)$ is $\Omega(n^2)$; see Section 2 and Figure 2.

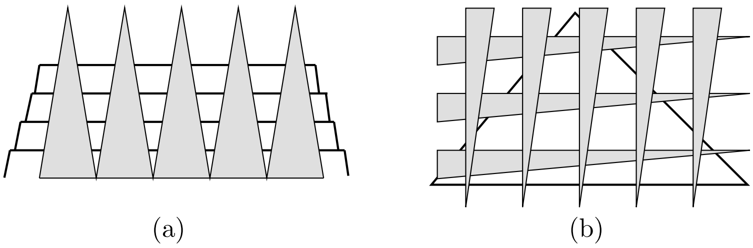

Figure 2: (a) A terrain that induces a strong edge visibility map of total complexity $\Omega(n^2)$. (b) A single triangle in a 3D scene (no polyhedral terrain) with a strong edge visibility map of $\Omega(n^2)$ on it. In both figures, the shaded triangles in the front induce edges and vertices on the white triangle(s) in the background.

**Lemma 6** *The strong visibility polyhedron* $\mathrm{SVP}(e,T)$ *of an edge $e$ in a 3D scene $T$ is connected.*

Proof. Let $p$ and $q$ be two points in $\mathbb{R}^3$ that strongly see $e$; this implies that the interiors of the triangles $\Delta_{ep}$ and $\Delta_{eq}$ are free from intersections with the interiors of the triangles in $T$. Let $x$ be any point on $e$. The two line segments $px$ and $qx$ are free from intersections with $T$ as well. Furthermore, for every point $y$ on either $px$ or $qx$, the triangle $\Delta_{ey}$ does not intersect $T$, and thus $y$ is in $\mathrm{SVP}(e,T)$. Therefore, $\mathrm{SVP}(e,T)$ is connected. ◻

Since the triangles of $T$ already induce $\mathrm{SVP}(e,T)$, the complexity of $\mathrm{SVM}(e,T)$ is at most the complexity of $\mathrm{SVP}(e,T)$. We analyze the complexity of $\mathrm{SVP}(e,T)$ to obtain an almost tight $O(n^2 \log n)$ upper bound on the complexity of $\mathrm{SVM}(e,T)$.

**Definition 7** *Let $p$ be a point, and let $s$ be a line segment, both in $\mathbb{R}^3$. We define $\mathcal{P}(p,s)$ to be the set of points $q$ for which there exists a ray starting at $p$ that first passes through $s$ and then through $q$. Similarly, $\mathcal{P}(s,p)$ is the set of points $q$ for which there exists a ray starting at a point on $s$ that first passes through $p$ and then through $q$.*

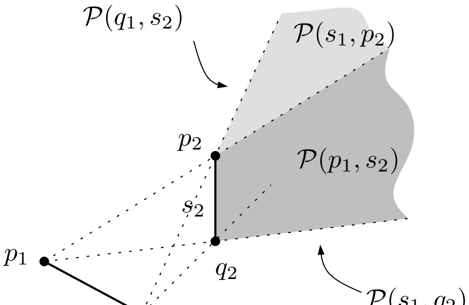

Figure 3: Illustration to Definition 7.

For two line segments $s_1 = p_1q_1$ and $s_2 = p_2q_2$, the (unbounded) region enclosed by $\mathcal{P}(s_1,p_2)$, $\mathcal{P}(p_1,s_2)$, $\mathcal{P}(q_1,s_2)$, and $\mathcal{P}(s_1,q_2)$ contains exactly the points $p$ for which the triangle with base $s_1$ and apex $p$ intersects the segment $s_2$; see Figure 3. For our purposes, if $v$ is a vertex and $e$

an edge of $T$, where $v$ and $e$ are not incident to each other, we represent the unbounded object $\mathcal{P}(e,v)$ by a large triangle, by clipping it with a vertical plane that lies beyond all features of $T$. Also, we clip and split $\mathcal{P}(v,e)$, and represent it by two triangles.

**Definition 8** *Given a 3D scene $T$ and an edge $e = vw$ of $T$, we define $S(e,T)$ to be the following set of triangles:*

$$\begin{aligned} S(e,T) \quad = \quad & \{\mathcal{P}(v,f) \mid f \textit{ edge of } T\} \ \cup \ \{\mathcal{P}(w,f) \mid f \textit{ edge of } T\} \ \cup \\ & \{\mathcal{P}(e,u) \mid u \textit{ vertex of } T\} \ \cup \ \{t \mid t \textit{ triangle of } T\} \end{aligned}$$

*$\mathcal{A}(S(e,T))$ is the arrangement induced by the triangles in $S(e,T)$.*

**Lemma 9** *Let $T$ be a 3D scene, and let $e$ be an edge of $T$.* SVP$(e,T)$ *is the union of a subset of cells of $\mathcal{A}(S(e,T))$.*

Proof. We prove the lemma by proving that every point of the boundary of SVP$(e,T)$ lies on one of the triangles of $S(e,T)$. Let $p$ be a point on the boundary of SVP$(e,T)$. Then, for some $\epsilon > 0$, there exist two points $p_1$ and $p_2$ in an $\epsilon$-neighborhood of $p$, such that $p_1 \in$ SVP$(e,T)$ and $p_2 \notin$ SVP$(e,T)$. Now imagine a point $p'$, at first located at the visible point $p_1$, which moves over $p$ to the invisible point $p_2$. When $p'$ coincides with the critical point $p$, $e$'s visibility from $p'$ changes from strongly visible to partially, or even completely, invisible. There are two ways in which this can happen: (i) $p'$ stops seeing one of the vertices of $e$, let $q$ be this vertex, or (ii) $p'$ stops seeing a point $q$ in the interior of $e$. In case (i) the segment $qp'$ intersects an edge of $T$, and in case (ii) $qp'$ intersects a vertex of $T$. And thus, by Definition 7, $p' = p$ lies on one of the triangles in $S(e,T)$. □

**Lemma 10** *Let $T$ be a 3D scene, and let $e$ be an edge of $T$.* SVP$(e,T)$ *is exactly one cell of $\mathcal{A}(S(e,T))$.*

Proof. By Lemma 9, SVP$(e,T)$ is the union of a subset of cells of $\mathcal{A}(S(e,T))$. Now, for the sake of contradiction, suppose there exists a point $p$ that lies in the interior of SVP$(e,T)$, i.e., it sees every point of $e$, and that simultaneously lies on a triangle in $S(e,T)$. The point $p$ lies (i) in $\mathcal{P}(v,f)$ where $v$ is one of the vertices of $e$, and $f$ is an edge of $T$, (ii) in $\mathcal{P}(e,w)$, where $w$ is a vertex of $T$, or (iii) on a triangle $t$ of $T$; see Figure 4.

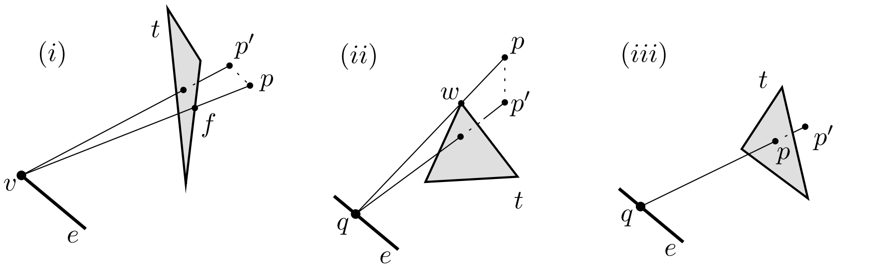

Figure 4: Illustration of the proof of Lemma 10.

For the first two cases, let $t$ be a triangle of $T$ that is incident to $f$ (i) or to $w$ (ii). Because the line-of-sight of $p$ to any point on $e$ is not allowed to graze the interior of $t$, there exists a point $p'$ infinitesimally close to $p$, such that the interior of the triangle with base $e$ and apex $p'$ intersects the interior of $t$. Therefore, $p'$ does not strongly see $e$. But then $p$ must lie on the boundary of SVP$(e,T)$, which is a contradiction. □

The arrangement $\mathcal{A}(S(e,T))$ is induced by the $O(n)$ triangles of the set $S(e,T)$, and thus has complexity $O(n^3)$ overall, but a single cell of this arrangement has complexity at most $O(n^2 \log n)$ [2, 22]. In conclusion, we have the following theorem.

**Theorem 11** *Let $T$ be a 3D scene with $n$ triangles, and let $e$ be an edge of $T$. The strong visibility map* $\mathrm{SVM}(e,T)$ *has complexity* $\Omega(n^2)$ *and* $O(n^2 \log n)$.

Lemma 10 directly gives a way to compute the strong visibility map.

**Algorithm** Compute_Strong_VM :
input: A set $T$ of $n$ triangles in $\mathbb{R}^3$ and an edge $e$ of $T$.
output: $\mathrm{SVM}(e,T)$

1. Generate the set $S(e,T)$ as defined in Definition 8.
2. Compute the cell $c$ of $\mathcal{A}(S(e,T))$ that contains $e$.
3. Compute the connected components of $c$ on each triangle of $T$ and output this set, which is $\mathrm{SVM}(e,T)$.

The first step trivially takes $O(n)$ time, the second step can be performed in randomized expected time $O(n^2 \log^3 n)$ by combining results of de Berg et al. [7] and Tagansky [22], and the third step can be carried out in $O(n^2 \log n)$ time in a fairly straightforward way. Thus, we have the following theorem.

**Theorem 12** *Let $T$ be a 3D scene with $n$ triangles, and let $e$ be an edge of $T$. The strong visibility map of $e$ on $T$ can be computed in* $O(n^2 \log^3 n)$ *randomized expected time.*

# 4 The Weak Visibility Map of an Edge

## 4.1 Lower Bounds

The weak visibility region of an edge in a polygon with holes can have complexity $\Omega(n^4)$, as was shown by Suri and O'Rourke [21]. In their construction, an edge light source and two rows of segments with suitably placed gaps $A$ and $B$ produce a quadratic number of narrow light cones that all mutually intersect. These intersections of cones result in $\Omega(n^4)$ illuminated vertices of the visibility region; see Figure 5 for a schematic illustration of this construction.

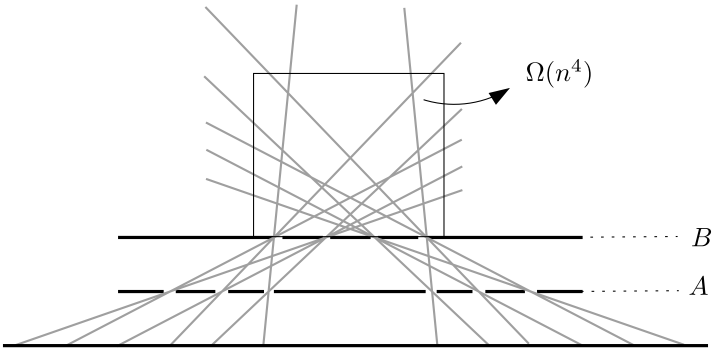

Figure 5: A schematic illustration of the lower bound construction for the weak visibility region of an edge in a polygon with holes [21].

We can create this same construction with a polyhedral terrain, where the 2D situation described above is replicated on the $xy$-plane, such that a single triangle contains all the $\Omega(n^4)$ intersections of the light cones. Thus, we have the following lemma.

**Lemma 13** *The weak visibility map of an edge in a polyhedral terrain with $n$ triangles can have* $\Omega(n^4)$ *vertices.*

For general scenes, we can improve this lower bound. We construct a scene such that every one of the $\Omega(n^2)$ light cones is unbounded in the $z$-direction, i.e., we have $\Omega(n^2)$ illuminated vertical planes, and each of the $\Omega(n^4)$ intersections of two cones is an illuminated vertical line. By placing a linear number of triangles vertically above each other, we can replicate the triangle that contains $\Omega(n^4)$ vertices of the visibility map $\Omega(n)$ times. Note that the triangles in this stack should be spaced in such a way that the light can extend through to the back of the stack. A schematic display of the lower bound construction for general 3D scenes is shown in Figure 6. Due to the vertically stacked triangles in $C$, this construction cannot be built as a polyhedral terrain.

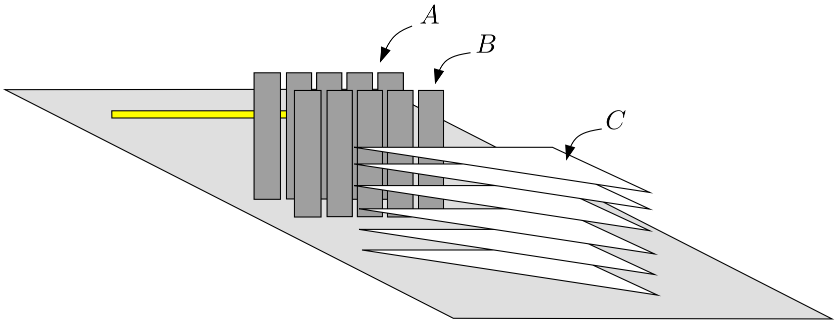

Figure 6: A schematic display of a 3D scene in which the weak visibility map of an edge has complexity $\Omega(n^5)$.

**Lemma 14** *The weak visibility map of an edge in a 3D scene with $n$ triangles can have $\Omega(n^5)$ vertices.*

## 4.2 Upper Bounds

Throughout this section, we are given a 3D scene $T$ with $n$ triangles and a designated edge $e$ of $T$ that acts as a light source. We prove upper bounds on the worst-case complexity of the shadow that $e$ induces, together with the scene triangles, onto different features of $T$. In Section 4.2.1, we prove that the problem of deciding whether a single point in $\mathbb{R}^3$ is illuminated is $\Theta(n \log n)$, in Section 4.2.2 we prove a bound of $\Theta(n^2)$ for the complexity of the shadow on an edge of $T$, and in Section 4.2.3 we prove $\Theta(n^4)$ for the complexity of the shadow on a triangle of $T$. Note that this last result, together with Lemma 14, implies that the complexity of the visibility map of an edge in a general 3D scene has complexity $\Theta(n^5)$.

### 4.2.1 Is a Point Illuminated?

First, we look at the shadow that an edge light source induces on a single point $p$ on a 3D scene, which is equivalent to asking whether $p$ is illuminated or not.

**Lemma 15** *Let $p$ be a point $p$ and let $e$ be an edge, both belonging to one of the $n$ triangles of a 3D scene $T$. If $e$ acts as a light source, then we can compute in $O(n \log n)$ time whether $p$ is illuminated.*

Proof. Let $\Delta$ be the triangle that has $p$ as its apex and the edge $e$ as its base. We compute the set $S$ of line segments that are intersections of the $O(n)$ triangles of $T$ with $\Delta$. This can be done in $O(n)$ time. Then, we project $S$ onto $e$ with respect to $p$, and compute whether these projections, which are intervals on $e$, cover $e$ completely. We can do this by using an $O(n \log n)$ sweepline algorithm. It is easy to see that $p$ is not illuminated by $e$ if and only if $e$ is completely covered by the projections of $S$, which can be checked while we sweep the intervals. □

The above bound on deciding whether or not a point is illuminated is in fact tight. The $\Omega(n \log n)$ lower bound follows from an adaptation of the lower bound proof of Klee's Measure Problem: given a set of $n$ (possibly overlapping) intervals $[a_i, b_i]$, compute the measure of their union. This problem is known to have a lower bound of $\Omega(n \log n)$ [10], and the proof can easily be adjusted to provide an $\Omega(n \log n)$ lower bound for this illumination problem.

### 4.2.2 Complexity on an Edge

By the lower bound construction in [21], the shadow map restricted to an edge can have complexity $\Omega(n^2)$, since there can be a quadratic number of light cones that intersect the edge in distinct places. The aspect graph gives us a trivial upper bound on the shadow of $O(n^3)$, because that is the maximum number of different views of the light source $e$ seen from $f$. We show that the $\Omega(n^2)$ lower bound is tight.

**Lemma 16** *Let $T$ be a 3D scene with $n$ triangles, and let $e$ and $f$ be two edges of $T$. If $e$ acts as a light source, then the shadow that $T$ casts on $f$ has complexity $O(n^2)$.*

Proof. If no point on $f$ is illuminated, then the shadow map on $f$ has complexity $O(1)$ and the lemma is trivially true. Otherwise, let $p$ be an arbitrary illuminated point on $f$, and let $q$ be a point on $e$ that illuminates $p$; the open line segment $pq$ does not intersect $T$. Now, we start moving $p$ over $f$, in either of the two directions. We can keep moving $p$ either until we reach the end of $f$, see Figure 7(a), or until the line segment $pq$ intersects an edge $f'$ of $T$. By construction, the point $p'$ we determine in this way is illuminated, namely by $q$. Next, we move the point $p'$ over $f$ in the same direction as we moved $p$, while maintaining contact with $f'$. This corresponds to sweeping the plane defined by $p'$ and $f'$ over $e$. We keep moving until we reach the end of $f$, which is similar to the case in Figure 7(a), we reach the end of either $e$ or $f'$ (see Figure 7(b)), or touch a second edge $g$ of $T$ (see Figure 7(c)). In each case, we determine a point $p''$ on $f$ that is illuminated by a point $q''$ on $f$.

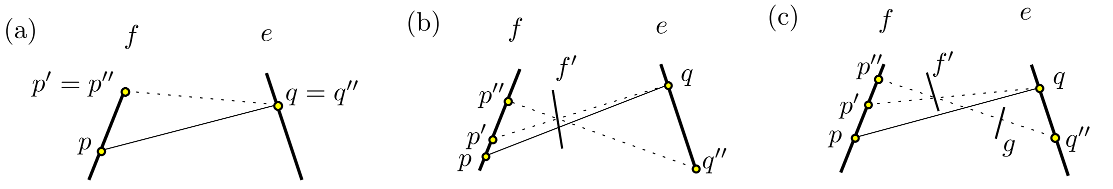

Figure 7: Three cases that arise in the proof of Lemma 16.

Observe that $p''$ is (a) an endpoint of $f$, (b) an extremal point on $f$ of one of the connected components of the regulus $\mathcal{R}(e, f', f)$ for some edge $f'$ of $T$, or (c) a point of intersection of $f$ with the regulus $\mathcal{R}(e, f', g)$ or the regulus $\mathcal{R}(e, g, f')$, where $f'$ and $g$ are edges of $T$. Analogously, we can find a similar illuminated point if we start by moving $p$ in the opposite direction. In the remainder of the proof, we bound the number of such points on $f$.

For every pair of edges $(f', g)$ of $T$, we consider the regulus $\mathcal{R}(e, f', g)$. If this set exists, then it intersects the edge $f$ at most in a constant number of points (see Section 2). Consider the arrangement $\mathcal{A}(f)$ on $f$ that is induced by the intersections of the reguli $\mathcal{R}(e, f', g)$ for all ordered pairs $(f', g)$ of edges of $T$. Since there are $O(n^2)$ such reguli, the arrangement $\mathcal{A}(f)$ has complexity $O(n^2)$. By the discussion above, every interval $i \in \mathcal{A}(f)$ is completely illuminated or completely dark. Thus, $O(n^2)$ is an upper bound on the complexity of the shadow map of $e$ on $f$. $\square$

### 4.2.3 Complexity on a Triangle

By the lower bound construction in [21], the shadow on a triangle $t$ of a 3D scene can have complexity $\Omega(n^4)$. The aspect graph gives us a trivial upper bound on the shadow of $O(n^6)$. Below, we proof that the lower bound is tight.

**Lemma 17** *Let $T$ be a 3D scene with $n$ triangles, let $e$ be an edge of $T$, and let $t$ be a triangle of $T$. If $e$ acts as a light source, then the shadow that $T$ casts on $t$ has complexity $O(n^4)$.*

PROOF. If no point on $t$ is illuminated, then the shadow has complexity $O(1)$ and the lemma is trivially true.

Otherwise, let $p$ be an arbitrary illuminated point on $t$, and let $q$ be a point on $e$ that illuminates $p$; the open line segment $pq$ does not intersect $T$. Now we choose any arbitrary straight line segment $s$ through $p$ that divides $t$ in two. By the argumentation in the proof of Lemma 16, we can find an illuminated interval on $s$, not necessarily maximal, that contains $p$ and is bounded on both sides by either an endpoint of $s$, or by a point that is the intersection of $s$ with a line that goes through $e$ as well as two edges $f$ and $g$ of $T$. Because $s$ was chosen arbitrarily, we can find a two-dimensional illuminated region on $t$ that contains $p$ and is bounded by curves that are intersections of $T$ with one or more reguli $R(e, f_i, g_j)$, where $f_i$ and $g_j$ are edges of $T$.

Consider the arrangement $\mathcal{A}(t)$ on $t$ that is induced by the curves that are intersections of all possible $R(e, f, g)$ for $f$ and $g$, with $f \neq g$ edges of $T$. Because there are $O(\binom{n}{2})$ edge pairs $(f, g)$, there are $O(n^2)$ such curves, and this arrangement on $t$ has complexity $O(n^4)$. By the discussion above, every cell $c \in \mathcal{A}(t)$ that contains at least one illuminated point, must be completely illuminated. Every cell $c' \in \mathcal{A}(e)$ that contains at least one shadow point, by the same arguments, must lie completely in the dark. Thus, $O(n^4)$ is an upper bound on the complexity of the shadow on $t$. □

Putting Lemmas 13, 14 and 17 together yields the following theorem.

**Theorem 18** *Let $T$ be a 3D scene with $n$ triangles, and let $e$ be an edge $T$. Then* $\mathrm{WVM}(e, T)$ *has complexity $\Theta(n^5)$ in the worst case. If $T$ is a polyhedral terrain, then the complexity of* $\mathrm{WVM}(e, T)$ *is $\Omega(n^4)$ and $O(n^5)$.*

## 4.3 Algorithm

To compute the shadow of $T$ that is cast onto an edge $f$ by the illuminating edge $e$, we use a sweep algorithm. Our algorithm is very similar to the one described in [6]; however, we sweep the *illuminated* edge instead of the *illuminating* one. The reason for this adjustment is that, in this way, we immediately obtain the subdivision of $f$ into illuminated and dark points, i.e., the shadow map of $e$ on $f$.

First, we compute the set $R$ of all reguli $R(e, f', g)$ for all ordered edge pairs $(f', g)$ of $T$; there are $O(n^2)$ such reguli. Then we compute $\mathcal{A}(R, f)$, the arrangement on $f$ induced by the intersections of the reguli in $R$ with $f$. For one of the endpoints of $f$, say $p_0$, we compute the intersections of the triangles of $T$ with the triangle $\Delta_{ep_0}$. We project these intersections onto $e$; we now have a set $S$ of $O(n)$ intervals on $e$, in these intervals either every point illuminates $p_0$ or none of them does. Now move the point $p_t$, which initially is the point $p_0$, over the edge $f$, which corresponds to sweeping the triangle $\Delta_{ep_t}$ through $T$. In the meantime, we maintain the set $S$ of (projected) intervals on $e$. The *event points* of the sweep are the points where (i) a new interval (either illuminating or not) appears, (ii) an existing interval disappears, or (iii) two endpoints of different intervals change order. Because the intervals correspond directly to (sets of) triangles of $T$, it is easy to see that the event points are exactly the interval boundaries of $\mathcal{A}(R, f)$. An interval $i$ of $\mathcal{A}(R, f)$ is illuminated if there is a point on $e$ that illuminates any point $p \in i$, and is in the shadow otherwise.

Computing $\mathcal{A}(R, f)$ takes $O(n^2 \log n)$ time, as we compute and sort the intersections of a quadratic number of reguli with an edge, and every regulus intersects an edge a constant number of times. Computing the set $S$ of intervals on $e$ that illuminate $p_0$ takes $O(n \log n)$ time; see Lemma 15. The number of event points we encounter during the sweep is of $O(n^2)$ [6] and at every event point we need to update the sweepline status in a constant number of places, since there are exactly two edges of $T$ associated with each event point. With an appropriate data structure, e.g., an augmented binary search tree, this step, and thus the complete algorithm, takes $O(n^2 \log n)$ time.

**Lemma 19** *Let $T$ be a 3D scene with $n$ triangles, and let $e$ and $f$ be two edges of $T$. We can compute the part of* $\mathrm{WVM}(e, T)$ *restricted to $f$ in $O(n^2 \log n)$ time.*

The shadow of $T$ cast onto a triangle $t$ of $T$ by the illuminating edge $e$ can be computed by a sweep algorithm as well. First, we need an extra lemma. Since two algebraic curves of constant degree in the plane have at most a constant number of intersections, it is easy to verify the lemma below (the proof is basically similar to the proof of Lemma 16 and the analysis in [6]).

**Lemma 20** *Let $e$ be an edge of a 3D scene $T$ with $n$ triangles, and let $c$ be an algebraic curve of constant degree on a triangle $t$ of $T$. If $e$ acts as a light source, then the shadow that $T$ casts on $c$ has complexity $O(n^2)$ and can be computed in $O(n^2 \log n)$ time.*

The first step in computing the shadow map of an edge $e$ on a triangle $t$ is to generate the set $S(t)$ of curves of intersection of the reguli $\mathcal{R}(e, f, g)$ with $t$, for all ordered edge pairs $(f, g)$ of $T$. Next, we compute the subdivision of $t$ induced by $S(t)$, denoted by $\mathcal{A}(S(t))$. This arrangement can be computed in $O(n^4)$ randomized expected time [16], or in $O(n^4 \log n)$ time by a standard sweep algorithm. We execute the $O(n^2 \log n)$ algorithm that computes the shadow map on an algebraic curve *twice* for every curve $c$ in $S(t)$; we compute curves $c'$ and $c''$ that, symbolically, lie infinitesimally close to $c$ on either side. By computing the shadow maps on $c'$ and $c''$ we can determine whether the adjacent cells of $\mathcal{A}(S(t))$ are **illuminated** or **dark**. Since we have $O(n^2)$ curves in $S(t)$, the cell labeling takes $O(n^4 \log n)$ time.

**Lemma 21** *Let $T$ be a 3D scene with $n$ triangles, let $e$ be an edge of $T$, and let $t$ be a triangle of $T$. We can compute the part of the weak visibility map of $e$ on $t$ in $O(n^4 \log n)$ time.*

Now it is trivial to extend the above procedure to compute the complete weak visibility map of $e$ on $T$:

```
Algorithm COMPUTE_WEAK_VM :
  INPUT: An edge $e$ and a 3D scene $T$.
  OUTPUT: $\mathrm{WVM}(e, T)$
  1. For every triangle $t$ of $T$, compute the shadow induced by $e$ on $t$ in the way described above.
  2. Return the set of all illuminated connected components, which is $\mathrm{WVM}(e, T)$.
```

We now have obtained the following result.

**Theorem 22** *Let $T$ be a 3D scene with $n$ triangles, and let $e$ be an edge of $T$. The weak visibility map* $\mathrm{WVM}(e, T)$ *can be computed in $O(n^5 \log n)$ time.*

# 5 The Triangle Visibility Map

We also consider the problem where the light source is neither a point nor a line segment, but a triangle in 3D. Using the techniques and ideas of the previous two sections, we show bounds on the worst-case combinatorial complexity of the two types of visibility maps for a triangle in 3D.

## 5.1 The Strong Visibility Map of a Triangle

The strong visibility map of a triangle $t$ in a 3D scene $T$ is the subdivision of the triangles of $T$ into points that completely see $t$ and points that do not. Like in Section 3, we bound the combinatorial complexity of the strong visibility polyhedron $\mathrm{SVP}(t, T)$, which also gives us an upper bound on the complexity of the strong visibility map $\mathrm{SVM}(t, T)$.

Let $e_1$, $e_2$, and $e_3$ be the three edges of $t$. We define $S(t, T)$ to be the set $\bigcup_{i=1..3} S(e_i, T)$, where $S(e_i, T)$ is defined as in Definition 8. Let $\mathcal{A}(S(t, T))$ be the spatial subdivision induced by the triangles of $S(t, T)$. By a similar argumentation as the proofs of Lemma 10 and Theorem 12, we have the following result.

**Theorem 23** *Let $T$ be a 3D scene with $n$ triangles, and let $t$ be a triangle of $T$. Then* $\mathrm{SVP}(t,T)$ *and* $\mathrm{SVM}(t,T)$ *have complexity $\Omega(n^2)$ and $O(n^2 \log n)$ and can be computed in $O(n^2 \log^3 n)$ randomized expected time.*

## 5.2 The Weak Visibility Map of a Triangle

As we did in Section 4, we distinguish the two cases where the scene $T$ is a terrain, and where $T$ is a general 3D scene.

### 5.2.1 Polyhedral Terrains

The lower bound construction where the visibility map of an edge has $\Omega(n^4)$ vertices (see Figure 5) trivially yields the same lower bound for a triangle light source.

The $O(n^5)$ upper bound on the complexity of the weak visibility map of an edge is also an upper bound for the weak visibility map of a triangle $t$. To verify this, suppose there is a point $p$ in the weak visibility map of $\mathrm{WVM}(t)$, and let $q$ be a point on $t$ that sees $p$. Then we can rotate the line-of-sight $pq$ vertically upwards around $p$, until we obtain a line-of-sight $pq'$ that is free from intersections with $T$, such that the point $q'$ lies on one of the edges of $t$. Hence, $\mathrm{WVM}(t) \subseteq \bigcup_{i=1..3} \mathrm{WVM}(e_i)$. Trivially, it also holds that $\mathrm{WVM}(t) \supseteq \bigcup_{i=1..3} \mathrm{WVM}(e_i)$. Because on every triangle of $T$, the complexity of $\bigcup_{i=1..3} \mathrm{WVM}(e_i)$ is bounded by the complexity of the arrangement induced by three sets of $O(n^2)$ reguli, $\mathrm{WVM}(t,T)$ is $\Theta(n^4)$ on a single triangle and $O(n^5)$ overall.

**Theorem 24** *Let $T$ be a polyhedral terrain, and let $t$ be a triangle of $T$. Then* $\mathrm{WVM}(t,T)$ *has complexity $\Omega(n^4)$ and $O(n^5)$.*

### 5.2.2 General 3D Scenes

Now, we consider the complexity of the weak visibility map of a triangle $t$ on a general 3D scene $T$. Recall from Definition 4 in Section 2 that the aspect graph is the subdivision of space into connected regions where the view is combinatorially the same. In other words, all points in a single cell of the aspect see exactly the same features of $T$. Obviously, in every cell the triangle $t$ is either visible or invisible. Thus, the complexity of the aspect graph *restricted to the triangles* of $T$ is an upper bound on $\mathrm{WVM}(t,T)$. The complete aspect graph, i.e., the subdivision of $\mathbb{R}^3$, is defined by $O(n^3)$ critical surfaces [8]. For every triangle, this gives us a set of $O(n^3)$ critical curves, which in turn induces an arrangement of complexity $O(n^6)$ on a single triangle, and thus $O(n^7)$ on $T$. Hence, we have the following lemma.

**Lemma 25** *Let $T$ be a 3D scene and let $t$ be a triangle of $T$. Then* $\mathrm{WVM}(t,T)$ *has complexity $O(n^7)$.*

In the remainder of this section, we show that $O(n^7)$ is tight, by extending the $\Omega(n^5)$ lower bound construction of Figure 6. The basic idea of the extension is the following: first, we 'replace' $A$ by a row giving $\Omega(n^2)$ gaps, which produces $\Omega(n^3)$ illuminated vertical planes, and thus we get $\Omega(n^6)$ illuminating vertical lines that each intersect $\Omega(n)$ triangles. In this way, we get a weak triangle visibility map with $\Omega(n^7)$ vertices. We describe this construction in more detail.

In order to construct a row with $\Omega(n^2)$ gaps that let the light that is emitted by $t$ pass through without using more than $O(n)$ triangles, we first make the following observation. Each of the illuminated vertical planes of the lower bound of Figure 6 is uniquely determined by two gaps: one in $A$ and one in $B$. Besides the obvious restriction that the light source should be located on the opposite side of $A$ than $B$, these illuminated planes do not depend on the location of the light source. Because we now have an extra degree of freedom, we have the opportunity to make $\Omega(n^2)$ gaps by building a grid $G$ instead of the row $A$. This idea is illustrated in Figure 8(a). The important fact here is that, if the holes in the grid degenerate to points, every hole has a unique $x$-coordinate.

If we allow enough distance between $G$ and $B$, the holes in $G$ basically act as point light sources, emitting cones of light. Now, just like in the old construction, each of the holes in $G$ interacts with each of the gaps in $B$, producing $\Omega(n^3)$ illuminated vertical planes in total; see Figure 8(b).

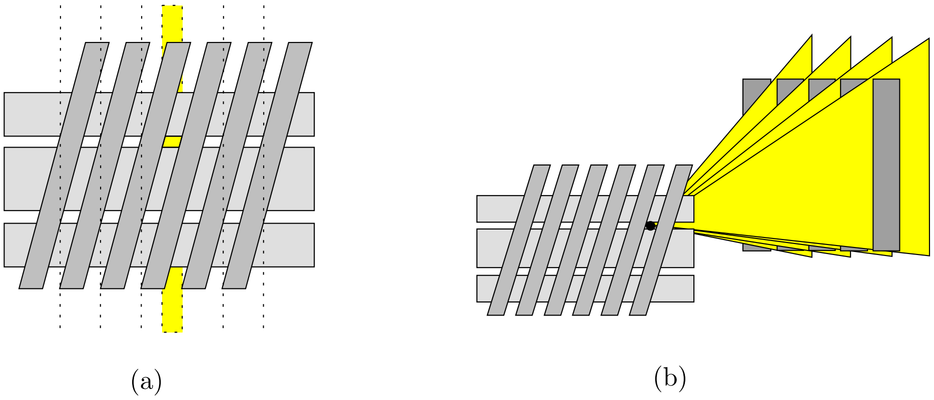

Figure 8: (a) A vertical grid with $\Omega(n^2)$ holes that allow light to pass through. (b) Every combination of a hole in the grid and a gap between two vertical slabs induces $\Omega(n)$ illuminated planes. (c) The total lower bound construction. The copy of $G$ and $B$ that induces another $\Omega(n^3)$ illuminated planes is not shown.

The total construction is illustrated schematically in Figure 9. First, we have the illuminating triangle $t$, followed by the grid $G$, then the row of vertical slabs $B$, and finally the set $C$ of vertically stacked triangles. To ensure that we indeed get $\Omega(n^6)$ illuminated vertical lines, we place a copy of $G$ and $B$ on the $xy$-plane such that they generate $\Omega(n^3)$ illuminated vertical planes from a different direction; the light source must be large enough to provide light for both constructions. This gives us another set of $\Omega(n^3)$ illuminated vertical planes, intersecting the $\Omega(n^3)$ planes that are produced by the first copy.

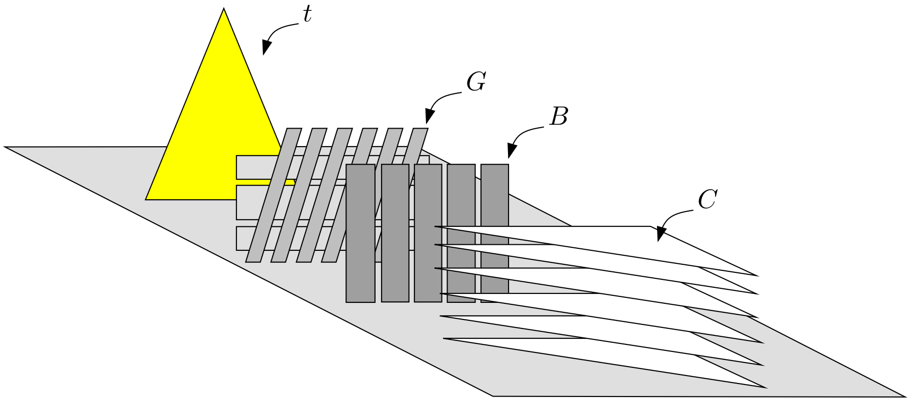

Figure 9: The total lower bound construction.

Note that we have to make sure that all the vertices we create on the white triangles of Figure 9 are indeed vertices of the weak visibility map of $t$. To achieve this, we add a constant number of very large triangles (not shown in Figure 9) to block all light coming from the light source that does not pass through the holes in $G$ and $B$ or the corresponding holes in the orthogonal copy of the construction. Concluding, we have the following theorem.

**Theorem 26** *Let $T$ be a 3D scene with $n$ triangles, and let $t$ be a triangle of $T$. Then* $\mathrm{WVM}(t, T)$ *has complexity* $\Theta(n^7)$ *in the worst case.*

### 5.3 Algorithm

We present an $O(n^7 \log n)$ time algorithm to compute $\mathrm{WVM}(t, T)$. Take one triangle $\Delta$ from $T$ and select and clip all triangles of $T$ to the convex hull of $t$ and $\Delta$. This gives a set $T'$ of triangles. Next, we compute the arrangement $\mathcal{A}$ of all reguli $\mathcal{R}(e, f, g)$ on $\Delta$, where $(e, f, g)$ is an ordered triple of edges from triangles in $T' \cup t$. Choose a point $p$ in some cell $c$ in $\mathcal{A}$, and compute the arrangement $\mathcal{E}$ of edges of $T'$ projected onto the plane supporting $t$. Because $\mathcal{A}$ corresponds directly to the aspect graph on $\Delta$, the combinatorial structure of $\mathcal{E}$ is the same for any point in $c$. It only changes when we move $p$ in $\mathcal{E}$ such that crosses a regulus $\mathcal{R}(e, f, g)$ and enters a cell $c'$ that is adjacent to $c$. The projected edges of $e$, $f$, and $g$ must have had subedges that formed a triangle in $\mathcal{E}$, which collapses when $p$ reaches $\mathcal{R}(e, f, g)$. When $p$ enters $c'$, an inverted version of this triangle appears.

After computing $\mathcal{E}$ for the point $p$ in $c$, we compute for every cell in $\mathcal{E}$ how many triangles of $T'$ cover it, and store this information with the cell. We also maintain whether $c$ lies inside or outside $t$; recall that $\mathcal{E}$ lies in the plane supporting $t$. Finally, we maintain a global count of the number of cells of $\mathcal{E}$ that lie inside $t$ and are not covered by any of the projected triangles. If for a cell $c$ of $\mathcal{A}$ the global count is zero, then $c$ is completely in the shadow; $c$ is illuminated if the global count is positive.

All of the above information can be updated when we go from cell $c$ to a neighboring cell $c'$ in $O(\log n)$ time, with the aid of a balanced binary search tree for each projected edge, which provides efficient access to $\mathcal{E}$. We have to make local changes to $\mathcal{E}$ for the collapsing and appearing triangle, and we must determine the number of projected triangles that cover the new triangle in $\mathcal{E}$. This is the number of triangles that covered the collapsing triangle, plus or minus 1 or 3; this can be determined from the edges $e$, $f$, and $g$ of $T'$. Since $\mathcal{A}$ has $O(n^6)$ cells, the total update time if we traverse all cells of $\mathcal{A}$ is $O(n^6 \log n)$. After we visited every cell of $\mathcal{A}$, we have computed the weak visibility map of $t$ on $\Delta$.

**Theorem 27** *Let $T$ be a 3D scene with $n$ triangles, and let $t$ be a triangle of $T$. The weak visibility map* $\mathrm{WVM}(t, T)$ *can be computed in* $O(n^7 \log n)$ *time.*

## 6 Concluding Remarks

We presented lower and upper bounds on the combinatorial complexities of various types of visibility maps of edges and triangles in a 3D scene. An interesting extension of this study, especially with computer graphics applications in mind, would be not just computing which points on $T$ are illuminated and which are dark, but also computing to which extent the points of $T$ are illuminated, i.e., what is the fraction of the points on $e$ that illuminate them. Also, as can be seen in Table 1, there are a few gaps left between lower and upper bounds. Finally, it would be interesting to look into output-sensitive algorithms to compute the different types of visibility maps.

## References

[1] P. K. Agarwal and M. Sharir. On the number of views of polyhedral terrains. *Discrete Comput. Geom.*, 12:177–182, 1994.

[2] B. Aronov and M. Sharir. Castles in the air revisited. *Discrete Comput. Geom.*, 12:119–150, 1994.

[3] M. Bern, D. Dobkin, D. Eppstein, and R. Grossman. Visibility with a moving point of view. In *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 107–117, 1990.

[4] H. Brönnimann, H. Everett, S. Lazard, F. Sottile, and S. Whitesides. Transversals to line segments in $\mathbb{R}^3$. In *Proc. 15th Canad. Conf. Comput. Geom.*, pages 174–177, 2003.

[5] N. Chin and S. Feiner. Fast object-precision shadow generation for areal light sources using BSP trees. *Comput. Graph.*, 25:21–30, Mar. 1992. Proc. 1992 Sympos. Interactive 3D Graphics.

[6] N. Coll, M. Fort, and J. Sellarès. Approximate multi-visibility map computation. In *Abstr. Europ. Workshop Comput. Geom.*, pages 97–100, 2005.

[7] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discrete Comput. Geom.*, 14:261–286, 1995.

[8] M. de Berg, D. Halperin, M. Overmars, and M. van Kreveld. Sparse arrangements and the number of views of polyhedral scenes. *Internat. J. Comput. Geom. Appl.*, 7:175–195, 1997.

[9] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9(1):66–104, 1990.

[10] M. L. Fredman and B. Weide. On the complexity of computing the measure of $\bigcup[a_i, b_i]$. *Commun. ACM*, 21:540–544, 1978.

[11] S. Ghali, E. Fiume, and H.-P. Seidel. Object-space, connectivity-preserving, shadow computation. In *Proc. Conf. Vis. Mod. Visualiz. (VMV-00)*, pages 281–289, 2000.

[12] Z. Gigus, J. Canny, and R. Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6):542–551, June 1991.

[13] J. Heintz, T. Reico, and M.-F. Roy. Algorithms in real algebraic geometry and applications to computational geometry. In J. E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 137–163. American Mathematical Society, Association for Computing Machinery, Providence, RI, 1991.

[14] J. J. Koenderink and A. J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[15] M. McKenna. Worst-case optimal hidden-surface removal. *ACM Trans. Graph.*, 6:19–28, 1987.

[16] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[17] J. O'Rourke. *Handbook of discrete and computational geometry*, chapter 28. Visibility. J.E. Goodman and J. O'Rourke, editors. CRC Press, Inc., Boca Raton, FL, USA, 2004.

[18] H. Plantinga and C. R. Dyer. Visibility, occlusion, and the aspect graph. *Internat. J. Comput. Vision*, 5(2):137–160, 1990.

[19] H. Pottmann and J. Wallner. *Computational Line Geometry.* Springer Verlag, Berlin, 2001.

[20] J.-R. Sack and J. Urrutia, editors. *Handbook of Computational Geometry.* North-Holland, Amsterdam, 2000.

[21] S. Suri and J. O'Rourke. Worst-case optimal algorithms for constructing visibility polygons with holes. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 14–23, 1986.

[22] B. Tagansky. A new technique for analyzing substructures in arrangements of piecewise linear surfaces. *Discrete Comput. Geom.*, 16:455–479, 1996.

[23] S. J. Teller. Computing the antipenumbra of an area light source. *Comput. Graph.*, 26(4):139–148, July 1992. Proc. SIGGRAPH '92.

[24] C. A. Wang and B. Zhu. Three-dimensional weak visibility: Complexity and applications. *Theor. Comput. Sci.*, 234(1-2):219–232, 2000.

[25] A. Woo, P. Poulin, and A. Fournier. A survey of shadow algorithms. *IEEE Comput. Graph. Appl.*, 10(6):13–32, 1990.