Region Intervisibility in Terrains

Esther Moet

Marc van Kreveld

René van Oostrum

institute of information and computing sciences, utrecht university

technical report UU-CS-2005-028

www.cs.uu.nl

Region Intervisibility in Terrains

Esther Moet Marc van Kreveld René van Oostrum

June 2005

Abstract

A polyhedral terrain is the graph of a continuous piecewise linear function defined over the triangles of a triangulation in the xy-plane. Two points on or above a terrain are visible to each other if the line-of-sight does not intersect the space below the terrain. In this paper, we look at three related visibility problems in terrains. Suppose we are given a terrain T with n triangles and two regions R_1 and R_2 on T, i.e., two simply connected subsets of at most m triangles. First, we present an algorithm that determines, for any constant $\epsilon > 0$, within $O(n^{1+\epsilon}m)$ time and storage whether or not R_1 and R_2 are completely intervisible. We also give an $O(m^3n^4)$ time algorithm to determine whether every point in R_1 sees at least one point in R_2. Finally, we present an $O(m^2n^2 \log n)$ time algorithm to determine whether there exists a pair of points $p \in R_1$ and $q \in R_2$, such that p and q see each other.

1 Introduction

Visibility computations in 3D, especially those with respect to terrain models, have their main application in geographic information systems (GIS). Examples are computations regarding horizon pollution and signal transmission, e.g., for mobile phone networks. Other application areas in which visibility problems in 3D arise are computer graphics and game design. This research area has received increasing attention from computational geometers over the last two decades.

The most common terrain model is the *polyhedral terrain*, which is the graph of a continuous piecewise linear function defined over the n triangles of a triangulation in the xy-plane. Three surveys of a variety of visibility problems and their solutions are [8, 11, 28]. Interest in algorithmic exploration of visibility in terrains is growing. For instance, there is a number of recent articles that study how the shortest pair of watchtowers that guards a terrain can be found efficiently [1, 3, 6].

Given a viewpoint on or above a terrain, the visibility map for that point is defined as the subdivision of the terrain into visible and invisible connected components. The most efficient and output sensitive algorithms to compute the visibility map of a terrain from a given viewpoint use hidden surface removal techniques. Reif and Sen [26] developed an $O((n + k) \log n \log \log n)$ time algorithm, where k is the size of the resulting visibility map, which can be quadratic. A few years later, Overmars, Katz and Sharir [20] reported an $O((n\alpha(n) + k) \log n)$ time algorithm, where $\alpha(n)$ is the extremely slowly growing inverse of the Ackermann–function.

In GIS, visibility problems on terrains are referred to as *viewshed analysis*. Efficient computation of the viewshed, i.e., the visibility map from a given viewpoint, has been studied extensively over the years [14, 15, 18, 27, 29]. In addition, some studies addressed the relation of the height of (points on) the terrain to visibility [18, 23]. The computation of visibility in GIS is usually performed with heuristical algorithms and provides approximate solutions [13, 17]. Recently, a method of terrain simplification that aims to preserve inter-point visibility was proposed [4]. In the algorithms community, the goal is to establish the computational complexity of GIS-inspired visibility problems and the worst-case running times of exact solutions are analyzed [5, 16]. This paper falls into the last category.

Previous algorithmic intervisibility studies only address either point-to-point or point-to-region visibility. In contrast, we study region-to-region visibility. A region is a simply connected subset of $O(m)$ triangles of the terrain. We introduce three ways to define region-to-region visibility: complete, semi-complete, and partial intervisibility. In the case of complete intervisibility, every point in one region sees every point in the other region and vice versa. Partial intervisibility signifies that there is a point in the one region that sees at least one point in the other region. Semi-complete intervisibility is a hybrid: every point in one region sees at least one point of the other region, and vice versa.

In this paper, we present three algorithms for the above mentioned problems. First, we determine whether two regions on a terrain are completely intervisible within time and storage bounds of $O(n^{1+\epsilon}m)$. We show that this problem is 3SUM-hard, so it is likely that our solution is close to optimal. The second algorithm we present tests for semi-complete intervisibility in $O(m^3n^4)$ time. Finally, we give an $O(m^2n^2 \log n)$ time algorithm to determine partial intervisibility.

The paper is structured as follows. In Section 2 we give definitions which we use in the rest of the paper. In Sections 3, 4, and 5, we consider complete, semi-complete and partial region intervisibility, respectively. In the last section, we summarize our results and give our views on future research. A preliminary abstract of this paper appeared earlier, containing only the results of Section 3.1 to 3.3 [30].

2 Preliminaries

In this section, we give several definitions regarding terrains and visibility that we use throughout this paper.

We define a *terrain* $T \subset \mathbb{R}^3$ to be a triangulated polyhedral surface with n vertices $V(T) = \{v_1, v_2, ..., v_n\}$. Each vertex v_i is specified by three real numbers (x_i, y_i, z_i), which are its cartesian coordinates. Furthermore, T can be intersected by any vertical line at most once. Therefore, T can also be viewed as the image of a piecewise linear function on $\mathbb{R}^2$. A *region* is a simply connected subset of $O(m)$ triangles of T.

We denote the plane that contains a given triangle t by $\mathcal{P}_t$. The half-space induced by $\mathcal{P}_t$ that contains the points above $\mathcal{P}_t$ is denoted by $\mathcal{P}_t^+$ and the other half-space by $\mathcal{P}_t^-$. The set of points above the entire terrain T is denoted by T^+, and the set of points below T is denoted by T^-. We assume that a triangle t of a terrain T is the union of its interior $int(t)$ and its boundary ∂t, i.e., t is a closed set. We denote the boundary of a region R by ∂R and the remainder of the points in R by $int(R)$. The closure of a (possibly open) set S is the union of S and ∂S, and is denoted by $cl(S)$. Throughout this paper, we use $O^*(f(n,m))$ as a shorthand for the bound $O(f(n,m) \cdot n^\epsilon)$, where $\epsilon > 0$ is an arbitrarily small constant. The $O^*(..)$-notation supresses polylogarithmic factors of n, and factors n^ϵ.

Given a terrain T in 3D and two points p and q, with $p, q \in T \cup T^+$, we say that p *sees* q

if the line segment pq does not intersect T^-, i.e. grazing contact of the line-of-sight with the terrain is permitted. Visibility is only defined on and above the terrain; so, if a point q in the interior of a triangle t sees a point p, then p must lie in $cl(\mathcal{P}_t^+)$.

Definition 1 *Let R_1 and R_2 be two regions on a terrain T. R_1 and R_2 are said to be* completely intervisible *if every point in R_1 sees every point in R_2. R_1 and R_2 are* partially intervisible *if there exists a point in R_1 that sees at least one point in R_2. Furthermore, R_1 is said to* weakly see *R_2 if every point in R_1 sees at least some point in R_2. If R_1 weakly sees R_2 and R_2 weakly sees R_1, then R_1 and R_2 are* semi-completely intervisible.

Observe that the above definitions are symmetric. Also note that if R_1 and R_2 have $O(m)$ straight line segments as edges, but do not consist of whole triangles of T, then we can refine T such that both regions do consist of whole triangles, while we add no more than $O(m)$ triangles.

Throughout this paper, we assume that two regions R_1 and R_2 on a terrain T are given, and we study complete, semi-complete, and partial intervisibility between R_1 and R_2 in Section 3, 4, and 5, respectively.

3 Complete Intervisibility

In this section, we look at complete visibility between two regions R_1 and R_2; in other words, we answer the following question:

Problem CompleteIntervisibility :
Does every point in R_1 see every point in R_2, and thus does every point in R_2 see every point in R_1 as well?

First, we obtain some preliminary results. In Section 3.2, we describe the algorithm and the data structures we use to test for complete intervisibility. Next, in Section 3.3, we derive an upper bound for the problem by analyzing the running time and storage of our algorithm, and we improve this bound by a time-memory trade-off. Finally, in Section 3.4, we prove that CompleteIntervisibility is 3sum-hard, which indicates that our algorithm is probably close to optimal.

3.1 Geometric Properties

To get started, we study the geometry of the problem. First, we investigate the orientations of R_1 and R_2 in space, i.e., for the time being, we ignore that other parts of T than the regions themselves can block visibility between R_1 and R_2; we remove this restriction later on. If R_1 and R_2 see each other completely, then all points in R_1 must lie in the intersection of the positive half-spaces induced by the triangles of R_2, and vice versa. Formally, the following condition has to be satisfied:

$$\forall p_1 \in R_1 : p_1 \in \bigcap_{t_2 \in R_2} cl(\mathcal{P}_{t_2}^+) \quad \wedge \quad \forall p_2 \in R_2 : p_2 \in \bigcap_{t_1 \in R_1} cl(\mathcal{P}_{t_1}^+) \tag{1}$$

Observe that if a point p in R_1 does not lie in the intersection of the half-spaces induced by the triangles of R_2, then at least one of the triangles of R_2 does not see p. Thus, condition (1) is necessary for complete visibility between two regions on a terrain. Note that it is not sufficient. We say that regions R_1 and R_2 are *facing each other* if condition (1) is satisfied. It is easy to verify that condition (1) need only be satisfied for the vertices of R_1 and R_2, in order for it to be satisfied for all points in the two regions, because $\bigcap_{t_2 \in R_2} cl(\mathcal{P}_{t_2}^+)$ and $\bigcap_{t_1 \in R_1} cl(\mathcal{P}_{t_1}^+)$ both are convex. Only when two regions R_1 and R_2 on a terrain T are facing each other, we have to consider the rest of the terrain as well to determine complete intervisibility. The visibility between points from R_1 and R_2 now depends on the rest of the triangles of T, i.e., those should not block the view.

Lemma 2 *Two regions R_1 and R_2 on a terrain T are completely intervisible if and only if*

1. *R_1 and R_2 are facing each other, and*
2. *∂R_1 and ∂R_2 are completely intervisible.*

Proof. The necessity of the first condition follows from the discussion above. The necessity of the second condition is easily understood. Because a region is a closed set, the boundary is a subset of it, and hence the boundaries of R_1 and R_2 have to see each other if the entire regions see each other.

To prove the sufficiency of the two conditions, we assume that 1. and 2. are satisfied, but for the sake of contradiction, we assume that there exist two points $p \in int(R_1)$ and $q \in int(R_2)$ that do not see each other, and show that this implies that there are points $p' \in \partial R_1$ and $q' \in \partial R_2$ that are not mutually visible.

Consider the vertical plane μ containing p and q. Let $T_\mu = T \cap \mu$ be the cross section of the terrain induced by μ; this is a lower-dimensional terrain itself, see Figure 1.

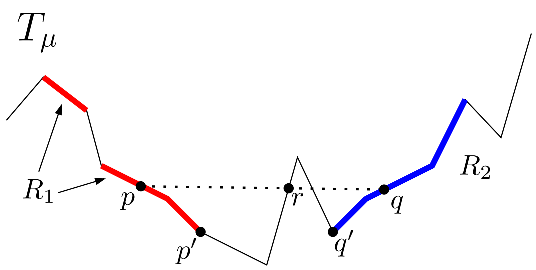


Figure 1: Illustration of the proof of Lemma 2.

The set $R_1 \cap \mu$ consists of one or more connected components on T_μ, and p lies in the interior of one of them; the same holds for R_2 and q. We only need to consider the visibility in T_μ to show that there exist a point $p' \in T_\mu \cap \partial R_1$ and a point $q' \in T_\mu \cap \partial R_2$ that do not see each other. Because the two triangles on which p and q lie are facing each other, the line segment pq does not intersect T_μ^- in an ϵ-neighborhood of p, nor in an ϵ-neighborhood of q. Let r be the point on T_μ closest to p that is in the closure of $T_\mu^- \cap pq$. If p and r are in the same connected component of $R_1 \cap \mu$, then the triangle containing r is not facing the triangle containing q, which contradicts the assumption. Otherwise, there exists a point $p' \in \partial R_1 \cap T_\mu$ that lies between p and r on T_μ, such that p' cannot see q. We repeat this argument with q

and p', which gives us a point $q' \in \partial R_2 \cap T_\mu$ that cannot see p'. Since $p' \in \partial R_1$ and $q' \in \partial R_2$, we have a contradiction, which proves the lemma. □

Note that checking complete intervisibility only between the vertices of ∂R_1 and the vertices of ∂R_2 is not sufficient, because a skinny spike in the terrain can block two boundary edges from being completely intervisible, while their endpoints can indeed see each other.

Definition 3 *Let R_1 and R_2 be two regions on a terrain T. The* visibility triangle set VTS *is defined to be the set containing all triangles that are induced by either (i) a vertex of ∂R_1 and an edge of ∂R_2, or (ii) an edge of ∂R_1 and a vertex of ∂R_2.*

Because grazing contact with the terrain is permitted, intervisibility between ∂R_1 and ∂R_2 (and thus between R_1 and R_2 is blocked if and only if there is a triangle $t \in$ VTS for which the intersection $t \cap T^-$ is nonempty. Now the name *visibility triangles* becomes clear: if all triangles in VTS are free from intersections with T^-, then there is complete intervisibility between R_1 and R_2. The following lemma holds for an arbitrary triangle with its vertices in $V(T)$, so it holds for the triangles in VTS in particular.

Lemma 4 *Let T be a terrain, and let t be an arbitrary triangle with its vertices in $V(T)$. The intersection $t \cap T^-$ is nonempty if and only if at least one of the following two situations occurs:*

1. *a vertex of T lies strictly above t, or*
2. *an edge of T lies strictly above an edge of t.*

Proof. First, we observe that if $t \cap T^-$ is nonempty, then there cannot be a single terrain triangle that lies above every point in t, since the vertices of t are vertices of T, and every vertical line intersects T at most once.

Necessity: Suppose the intersection $t \cap T^-$ is nonempty, and let p be an arbitrary point in $t \cap T^-$. There is a unique point $q \in T$ that lies vertically above p; let t' be the triangle of T that contains q. Because a triangle is a closed set, we have that a vertex of t' (and of T) lies strictly above t, that an edge of t' (and of T) lies strictly above an edge of t, or both.

Sufficiency: If no vertex or edge of T is strictly above t, there are two situations: t is completely contained in T, or t intersects both T and T^+. In both cases, t does not intersect T^-. □

3.2 Algorithm

In this section, we describe the algorithm to determine complete visibility between two regions in a terrain, and the data structures it uses. Using the results from Section 3.1, this algorithm can be described as follows:

Algorithm Compute_Complete_Intervisibility :
INPUT: A terrain T and two regions R_1 and R_2 on T.
OUTPUT: **True** if R_1 and R_2 are completely intervisible, **False** otherwise.

1. Compute whether R_1 and R_2 satisfy condition (1), i.e., whether they are facing each other; return **False** if they are not.

2. Construct the set VTS as defined in Definition 3.
3. Determine whether ∂R_1 and ∂R_2 are intervisible in two steps:
   (a) Test whether there is a vertex of T that lies above some triangle of VTS.
   (b) Test whether there is an edge of T that lies above some edge of a triangle in VTS.
4. If both step 3(a) and 3(b) yield **False**, return **True**. Otherwise, return **False**.

In the first step, we determine whether the two regions are facing each other. For each region R_i, $i = 1, 2$, we compute $\bigcap_{t \in R_i} \mathcal{P}_t^+$, which is the intersection of $O(m)$ half-spaces and can be computed in $O(m \log m)$ time [10]. Next, we check for all $O(m)$ vertices of the other region whether they are contained in $\mathcal{C}_i = cl(\bigcap_{t \in R_i} \mathcal{P}_t^+)$. Since $\mathcal{C}_i$ is unbounded in the z-direction, point location for the 3D volume $\mathcal{C}_i$ corresponds to planar point location with respect to its xy-projection. Thus, we can preprocess $\mathcal{C}_i$ for point location in $O(m \log m)$ time and then query with $O(m)$ points in $O(\log m)$ time per query [10]. In conclusion, we can check if two regions on a terrain are facing each other in $O(m \log m)$ time.

Constructing VTS can trivially be done in $O(m^2)$ time. For the third step of the algorithm (as a whole), we have to check for intersections between a set of $O(n)$ terrain triangles and a set of $O(m^2)$ triangles induced by the edges and vertices of ∂R_1 and ∂R_2. In a brute-force algorithm, this leads to $O(nm^2)$ time. However, we can do better if we perform steps 3(a) and 3(b) separately.

Step 3(a): No terrain vertex above a visibility triangle

To reduce the time complexity, it is helpful to take a different look at the geometry. For a given vertex v of the terrain, we want to check whether there is any triangle in VTS that lies strictly below v. Obviously, we want to limit the number of triangles to test against v.

If v lies vertically above a triangle t in VTS, then the projection of v onto the xy-plane lies in the projection of t. The set VTS contains $O(m^2)$ non-disjoint triangles; we want to retrieve exactly those triangles $S(v) \subseteq$ VTS that contain the projection of v in their projections. Thus, the remaining question is: does v lie below all triangles of $S(v)$? This question can also be simplified by looking at the geometry. If a point p in 3D does not lie below all triangles in a set $S(v)$, it lies outside of the polyhedron $\bigcap_{t \in S(v)} \mathcal{P}_t^-$.

To find the triangles from VTS that contain a given point in their projections on the xy-plane, we construct a partition tree of $O(m^2)$ size [10]. Given a query point p, it returns $O^*(m)$ canonical subsets of triangles from VTS that contain p in their projection. We give every node μ in the partition tree an associated data structure of linear complexity in the cardinality of the canonical subset $c(\mu)$. This structure is used to determine whether the query point lies in the intersection of the half-spaces induced by all triangles in $c(\mu)$. This query can be answered in $O(\log m)$ time after $O(m \log m)$ preprocessing time, by using the Dobkin–Kirkpatrick hierarchy [12, 24].

The construction time for a partition tree of size $O(m^2)$ is $O(m^{2+\epsilon})$ for any constant $\epsilon > 0$ [10]. Fortunately, the space required for the partition tree's associated structure does not increase the total storage space much, in particular, nothing at all in O^*-notation. The total data structure requires $O(m^2 \log m)$ space. The construction time is $O^*(m^2)$ and the total query time is $O(n)$ times $O^*(m)$, which is $O^*(mn)$.

Step 3(b): No terrain edge above a visibility edge

We want to limit the number of terrain edges against which we check the visibility edges. We project the terrain edges onto the xy-plane and for every edge from VTS, we want to find the terrain edges whose projections intersect the projection of the query edge. Note that no edge projects to a single point. When edges of the terrain and the query edge intersect in the projection, we can treat both these terrain edges and the query edge as full lines in 3D. The objective now is to find out whether the line supporting the query edge lies above all lines supporting the selected terrain edges.

The data structure that we use stores the projections of the terrain edges onto the xy-plane in a cutting tree of size $O(n^2)$ [10]. An ordinary cutting tree stores unbounded lines and given a query point, it returns those lines that lie above the query point. Because we want to query with a line segment s_q and would like to retrieve those line segments that intersect s_q, we create a four-level cutting tree of the same size. With the first two levels, we retrieve those line segments whose supporting lines lie above one of the endpoints of s_q and below the other. To narrow this set of line segments down to the ones that actually intersect with s_q, we add another two levels in the cutting tree; these are cutting trees on the dual arrangement of the endpoints of line segments that appear in canonical subsets of the first two levels.

We perform at most m^2 queries on this tree with edges from VTS, each taking $O(\log n)$ query time. The query returns all intersecting terrain edges in $O(\log n)$ canonical subsets. We give each node μ in the cutting tree an associated structure of size $O(n^{2+\epsilon})$ for the canonical subset $c(\mu)$ of edges as the supporting lines in space [7]. Consequently, we can decide whether a query line lies above all lines in the canonical subset in $O(\log n)$ query time.

The storage required for the cutting tree and its associated structure is $O(n^{2+\epsilon})$, and the total data structure can be constructed in $O^*(n^2)$ time. The total query time is $O(m^2 \log n)$, or $O^*(m^2)$, and thus the total time complexity of step 3(b) is $O^*(n^2 + m^2) = O^*(n^2)$.

3.3 Running Time Analysis

So far, we have achieved an upper bound of $O^*(n^2)$ for problem CompleteIntervisibility, which improves the $O(nm^2)$ time of a brute-force algorithm. In Table 1, we summarize the time and storage complexities for the partition tree with its associated structures (PT) and the cutting tree with its associated structures (CT).

	Preprocessing	Total queries	Storage
PT	$O^*(m^2)$	$O^*(mn)$	$O(m^2)$
CT	$O^*(n^2)$	$O^*(m^2)$	$O^*(n^2)$

Table 1: Summary of time and storage complexities

Now we perform a trade-off between the storage and query time to improve the upper bound to $O^*(mn)$ time.

In step 3(b), we determined whether any terrain edge lies above any visibility edge by constructing an $O^*(n^2)$ size cutting tree with an associated structure for lines in space on the terrain edges. To make the time complexity more dependent on m than on n, we create $k = n/m$ groups of terrain edges of size n/k each and build k of the augmented cutting trees described above, one for each group. This implies that every structure has size $O^*(\frac{n^2}{k^2}) =$

$O^*(m^2)$ and the total complexity of all structures is $O^*(\frac{n^2}{k}) = O^*(nm)$. If we query these k structures with a single visibility edge, this takes $O(k \log \frac{n}{k})$ time, so for m^2 queries, this takes $O(m^2 k \log \frac{n}{k}) = O^*(nm)$ time. This is the same time complexity as we need to perform step 3(a), so there is no need to improve that step as well (even though we could perform a time-memory trade-off there to achieve a running time of $O^*(n^{\frac{2}{3}} m^{\frac{4}{3}} + n)$).

By putting the results of Sections 3.3 to 3.4 together, we get the following result.

Theorem 5 *Let T be a terrain, and let R_1 and R_2 be two regions on T, with $O(m)$ triangles each. Complete intervisibility of R_1 and R_2 can be determined in $O^*(nm)$ time and space.*

3.4 3SUM-hardness

In this section, we prove that COMPLETEINTERVISIBILITY is at least as difficult as the well-known problem 3SUM.

Problem 3SUM :
Given a set S of m integers, are there three elements of S that add up to 0?

The best known algorithm for this problem takes $\Theta(m^2)$ time. If a problem is at least as difficult to solve as 3SUM, we say that it is 3SUM-*hard*, which implies that there is little hope of finding a subquadratic time algorithm for that problem. In [19], a large collection of problems is described, with proofs that all of these problems are 3SUM-hard. One of these problems is GEOMBASE.

Problem GEOMBASE [19] :
Given a set of m points S with integer coordinates on three horizontal lines $y = 0$, $y = 1$, and $y = 2$ in the plane, determine whether there exists a non-horizontal line containing three of the points.

We give a reduction from GEOMBASE to our problem and thus prove that COMPLETEINTERVISIBILITY is 3SUM-hard. To achieve this, we first assume that grazing contact of the line-of-sight between two points is not permitted; later, we show how we can relax this restriction to obtain the reduction in our original setting. Assume we are given an instance S of the problem GEOMBASE. We transform this into an instance of COMPLETEINTERVISIBILITY, such that the answer we obtain by solving COMPLETEINTERVISIBILITY is the same as the answer we would have obtained had we solved GEOMBASE directly. Let A be those points of S that lie on the line $y = 0$, let B be those on the line $y = 1$, and C those on $y = 2$. The aim is to create a terrain T in which there is *no complete intervisibility* if and only if *there exists a line* through a point in A, one in B and one in C.

We construct T by placing two copies of a steep slope opposite each other. We place the two regions R_1 and R_2 on these slopes; the regions are both similar, namely a rectangular shape with a number of small protrusions at the bottom. This construction is displayed in Figure 2. The vertices of the protrusions of both R_1 and R_2 are located at the same height value, say $z = 1$. The vertices of R_1 are located at the x- and y-coordinates of the $O(m)$ points in A and the vertices of R_2 at the x- and y-coordinates of the $O(m)$ points in C. In between R_1 and R_2, we place a number of skinny peaks with their summits at the x- and y-coordinates of the $O(m)$ points in B and height $z = 1$. Finally, we connect all subconstructions such that they form a terrain.

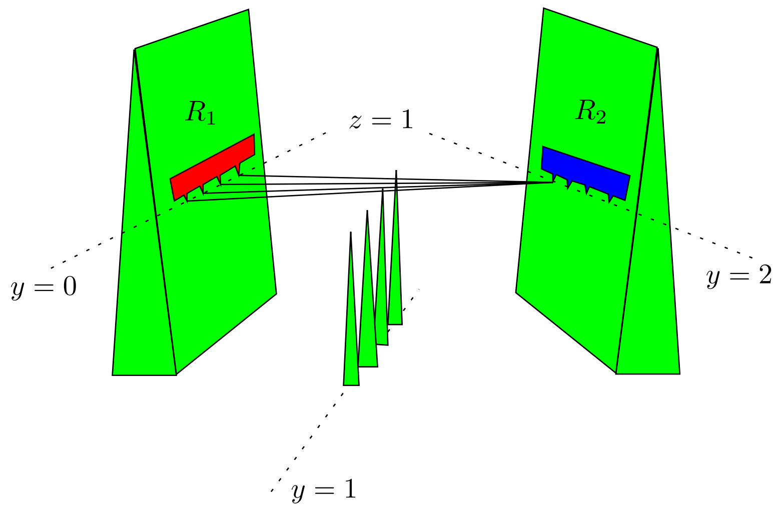
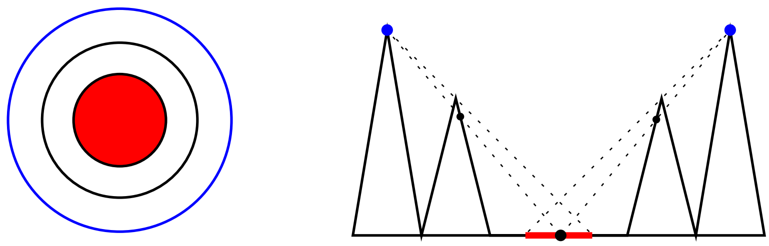


Figure 2: Reduction of GEOMBASE to COMPLETEINTERVISIBILITY.

Now it is clear that if and only if the answer to GEOMBASE is "no", the answer to COMPLETEINTERVISIBILITY is "yes", i.e., the regions R_1 and R_2 are completely intervisible. The reduction takes $O(n \log n)$ time, since we connect everything to obtain a single terrain, which involves triangulating the different parts of the construction. Because we did not allow grazing contact of the line-of-sight in this reduction, we have to adjust our construction such that it also is valid if grazing contact is permitted. To achieve this, we symbolically say that the summits of the peaks that correspond to the points in B are infinitesimally higher whenever we compare them in height to another feature of the terrain. Therefore, we have a valid reduction from GEOMBASE and our problem is 3SUM-hard. Since $\Omega(n)$ is also an (obvious) lower bound for COMPLETEINTERVISIBILITY, we conclude that it is unlikely that an optimal algorithm to determine complete region intervisibility has a running time of $o(m^2 + n)$.

4 Semi-Complete Intervisibility

We next present an algorithm to determine whether two regions R_1 and R_2 are semi-completely intervisible. Semi-complete intervisibility is much less restricted than complete intervisibility. For example, the triangles in the set VTS (from Definition 3) can be intersected by the terrain many times while the two region boundaries are still semi-completely intervisible. We expect an algorithm to determine semi-complete intervisibility to be more complex and time-consuming than an algorithm to determine complete intervisibility. Furthermore, contrary to the approach of the previous section, computing only intervisibility between the boundaries of the two regions is not sufficient; see Figure 3(a) for a schematic display of the xy-projection of a counterexample, and Figure 3(b) for a cross section. Let the flat disk (red) be the first region, and the upper ridge (blue) the second region. While the boundaries of the two regions are semi-completely intervisible, even completely, there is a point in the interior of the disk that does not see any point on the ridge.



(a) (b)

Figure 3: The boundaries of two regions can be semi-completely intervisible, while the regions themselves are not semi-completely intervisible.

4.1 Aspect Graph Approach

To start, we compute whether every point in R_1 sees at least some point in R_2, i.e., we compute whether R_2 is weakly visible from every point in R_1. The other way around can be computed analogously. To obtain some more intuition for the problem at hand, we first describe a brute-force approach to compute whether R_2 is weakly visible from every point in R_1. In Section 4.2, we significantly improve on the $O(mn^8 \log n)$ brute-force time bound.

The aspect graph, first proposed by Koenderink and van Doorn [22], is an important structure for studying visibility in 3D. For our setting of a polyhedral terrain, this structure is defined as the graph in which the nodes represent the set of combinatorially different views of the terrain, and the arcs represent the transitions between these views due to continuous movement of the viewpoint. In this definition, a *view* refers to a view of the terrain such that an infinitesimal movement will yield a view of the object that has the same combinatorial structure. We do not use the aspect graph itself, but the subdivision of space it induces. This subdivision of $\mathbb{R}^3$ consists of cells in which all points in the same cell have the same view combinatorially. A shared boundary between two cells indicates a transition between the views of the corresponding cells. Although this is a slightly different structure than the original graph from [22], we call it an aspect graph in this paper.

There are $O(n^3)$ transitions between different views of the terrain, and these can be subdivided into three types: $O(n^3)$ transitions are defined by three edges of the terrain, $O(n^2)$ transitions are defined by a vertex and an edge, and $O(n)$ are defined by a single triangle [9]. In Figure 4 these three types are illustrated.

All of these transitions define either a plane or an algebraic surface of constant degree. If the viewpoint is on one side of such a plane or surface, the view is combinatorially different than if the viewpoint is on the other side. Because grazing contact of the line-of-sight with the terrain is allowed, the boundary between two cells has the same combinatorial view as the cell that 'sees the most' of the two. These $O(n^3)$ planes and surfaces and their intersections subdivide space into cells and thus the aspect graph has complexity $O(n^9)$. However, in [2], the complexity of this structure for a polyhedral terrain is proven to be $O(n^{8+\epsilon})$. If we compute the aspect graph of the terrain on R_1, we get a partition of R_1 into a finite set of aspect graph cells inside which all points have the same view of the terrain. We compute the intersection of the visibility map and R_2 for one sample point in each such cell. This intersection is nonempty for each cell if and only if R_2 is weakly visible from R_1.

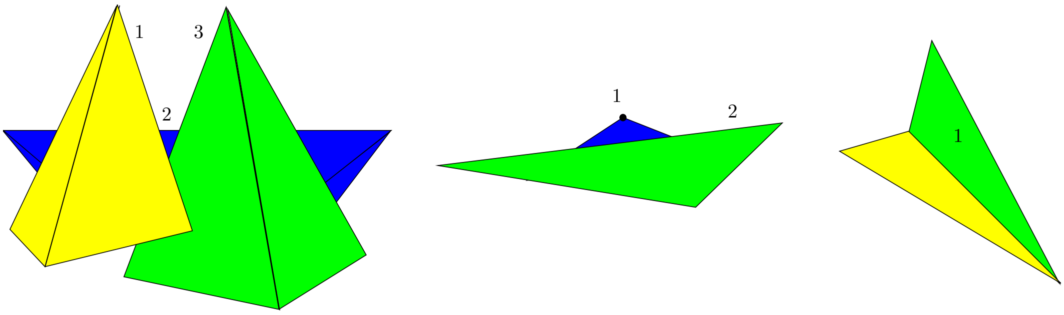


Figure 4: The three types of transitions between different views.

Because the $O(n^3)$ planes and surfaces partition a single triangle into $O(n^6)$ cells, we have $O(n^6)$ sample points for every triangle of R_1. For every sample point, we can compute the visibility map of the terrain in $O(n^2 \log n)$ worst-case time [20], so for a region of $O(m)$ triangles, the total time complexity will be $O(mn^8 \log n)$. In the next section, we improve on this bound by taking a different approach.

4.2 Using the Edge Visibility Map

In this section, we take a conceptually different look at semi-complete intervisibility between two regions. In general, visibility and illumination problems are closely related; therefore, let us assume that every point in R_2 emits light, and then consider which points on the terrain, and in particular on R_1, are illuminated. These are exactly the points that see at least one point in R_2. In the terminology of Definition 1, we now have that R_1 weakly sees R_2 if and only if R_1 is completely illuminated. The following observation follows directly from the property that every vertical line intersects a terrain at most once.

Observation 6 *Let p be a point and R a region, both on a terrain T. If p is illuminated by some point in R, it is illuminated by a point on an edge of one of the triangles in R.*

This observation gives us a new way to test for semi-complete intervisibility between R_1 and R_2. First, for every illuminating edge of R_2, we compute the subdivision of T into illuminated and shadow points. Next, we overlay these $O(m)$ subdivisions. Finally, we determine whether or not R_1 is completely illuminated.

It has recently been shown that an illuminating edge induces a *shadow map* of complexity $\Theta(n^2)$ onto another terrain edge and a shadow map of complexity $\Theta(n^4)$ onto a terrain triangle [21]. The computation of these shadows takes $O(n^2 \log n)$ and $O(n^4 \log n)$ time, respectively. Note that the shadow map of a point light source is equivalent to the visibility map of that point acting as a viewpoint.

Fortunately, we do not need to overlay all the separate shadow maps explicitly, which would very time-consuming. The $O(n^4 \log n)$ time algorithm that computes the shadow map of a single illuminating edge e onto a triangle t of T first computes an arrangement on t that is induced by $O(n^2)$ critical surfaces and then labels each cell of this arrangement as being either illuminated or in the shadow [21]. A multi-edge shadow map for $O(m)$ illuminating edges can be computed in the same way, with the difference that there now are $O(mn^2)$ critical surfaces that induce the arrangement, which thus has complexity $O(m^2n^4)$. This arrangement $\mathcal{A}$ can be computed in $O(m^2n^4)$ randomized expected time [25]. If we compute the $O(m)$ individual shadow maps in advance, in $O(mn^4 \log n)$ time, we can walk through each of these maps and simultaneously label the cells in $\mathcal{A}$ that are contained in an illuminated cell of one of the shadow maps as illuminated. This labeling can be performed in $O(mn^4 \log n)$ time, since we walk through $O(m)$ arrangements of complexity $O(n^4)$. Thus, we can compute which parts of a terrain triangle t are illuminated by the edges of the triangles in R_1 in $O(m^2n^4)$ time, and for the entire terrain in $O(m^3n^4)$ time. Once we have this information, it is easy to determine whether R_2 is completely illuminated within the same time bound.

Theorem 7 *Let T be a terrain, and let R_1 and R_2 be two regions on T, with $O(m)$ triangles each. Semi-complete intervisibility of R_1 and R_2 can be determined in $O(m^3n^4)$ time.*

5 Partial Intervisibility

Finally, we consider the problem of determining whether there exists a point in region R_1 that sees any point in R_2. The proof of the following lemma is straightforward and in fact very similar to the proof of Lemma 2.

Lemma 8 *Two regions R_1 and R_2 on a terrain are partially intervisible if and only if there exists a point p on an edge of a triangle in R_1 and a point q on an edge of a triangle in R_2 such that p sees q.*

Proof. The if-part of the proof is immediate from Definition 1.

For the only if-part, let p be a point in R_1 and let q be a point in R_2, such that p sees q. Let μ be the vertical plane through the line segment pq, and let $T_\mu = T \cap \mu$ be the cross section of T induced by μ. We only need to consider the visibility in T_μ to show that there exist a point v on an edge of R_1 and a point w on an edge of R_2 that see each other. Because T_μ is a lower-dimensional terrain, we can rotate the ray emanating from p and directed towards q upwards around p, until we hit a vertex v of T_μ, such that v and p are mutually visible. Note that v is a point on an edge of R_2; recall that R_2 consists of complete triangles of T. We repeat this process by rotating the ray from v towards p upwards, to find a vertex w of T_μ such that v sees w. Now we have that w is a point on an edge of R_1, and the fact that v and w see each other completes the proof. □

Using the same concept of illumination instead of visibility as in the previous section, this problem turns out to be quite easy to tackle. It is easy to verify that if the edges of R_1 act as light sources, and R_1 and R_2 are partially intervisible, then some edge in R_2 contains an illuminated point. As we showed in the previous section, the shadow map induced by an illuminating edge onto another edge has complexity $\Theta(n^2)$ and can be computed in $O(n^2 \log n)$ time. Since the shadow map is a one-dimensional structure, there is no need to actually overlay m shadow maps on every edge; we simply compute them sequentially and stop as soon as we find an illuminated point. In particular, we run the $O(n^2 \log n)$ time edge-to-edge shadow map algorithm of [21] for all $O(m^2)$ different edge pairs of one edge in R_1 and one in R_2. This gives us the following theorem:

Theorem 9 *Let T be a terrain, and let R_1 and R_2 be two regions on T, with $O(m)$ triangles each. Partial intervisibility of R_1 and R_2 can be determined in $O(m^2n^2 \log n)$ time.*

6 Concluding Remarks

In this paper, we considered three problems related to the intervisibility of two regions of at most m triangles on a terrain with n triangles. We first developed an algorithm that, for any $\epsilon > 0$, determines in $O(n^{1+\epsilon}m)$ time whether the two regions in a terrain are completely intervisible, whereas a brute-force algorithm would require $O(m^2n)$ time. The algorithm uses a partition tree and a cutting tree, both with associated structures, leading to a total of $O^*(mn)$ storage. Moreover, we proved that computing complete visibility between two regions is 3SUM-hard.

Secondly, we developed an algorithm to determine in $O(m^3n^4)$ time whether two regions in a terrain are semi-completely intervisible. The algorithm involves computing the shadow maps of the edges of the first region onto the second region, and checks whether the second region is completely illuminated (and vice versa). Finally, we presented a similar algorithm to determine partial region intervisibility, which runs in $O(m^2n^2 \log n)$ time.

It should be noted that within the same time bounds, the different kinds of intervisibility between a constant number of regions can be determined, because every extra region requires a constant number of repetitions of the algorithm which does not increase the execution time asymptotically.

A number of other intervisibility problems are still unexplored, especially those that concern partial visibility. A possible topic for future research is partial visibility from a watchtower of, for example, a path through a terrain or a collection of other watchtowers.

Acknowledgements

The authors thank Christian Knauer for useful suggestions.

References

[1] P. K. Agarwal, S. Bereg, O. Daescu, H. Kaplan, Ntafos, and B. Zhu. Guarding a terrain by two watchtowers. *Proceedings of the 21st ACM Annual Symposium on Computational Geometry*, pages 346–355, 2005.

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

[3] B. Ben-Moshe, P. Carmi, and M. J. Katz. Computing all large sums-of-pairs in $\mathbb{R}^n$ and the discrete planar two-watchtower problem. *Information Processing Letters*, 89:137–139, 2004.

[4] B. Ben-Moshe, M. J. Katz, J. S. B. Mitchell, and Y. Nir. Visibility preserving terrain simplification—an experimental study. *Comput. Geom. Theory Appl.*, 28:175–190, 2004.

[5] M. Bern, D. Dobkin, D. Eppstein, and R. Grossman. Visibility with a moving point of view. *Algorithmica*, 11:360–378, 1994.

[6] S. Bespamyatnikh, Z. Chen, K. Wang, and B. Zhu. On the planar two-watchtower problem. In *Proc. 7th Ann. Internat. Conf. Computing and Combinatorics*, pages 121–130, 2001.

[7] B. Chazelle, H. Edelsbrunner, L. J. Guibas, M. Sharir, and J. Stolfi. Lines in space: Combinatorics and algorithms. *Algorithmica*, 15:428–447, 1996.

[8] R. Cole and M. Sharir. Visibility problems for polyhedral terrains. *J. Symbolic Comput.*, 7:11–30, 1989.

[9] 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.

[10] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, Germany, 2nd edition, 2000.

[11] L. De Floriani and P. Magillo. Intervisibility on terrains. In P.A.Longley, M.F.Goodchild, D.J.Maguire, and D.W.Rhind, editors, *Geographic Information Systems: Principles, Techniques, Managament and Applications*, chapter 38, pages 543–556. John Wiley & Sons, 1999.

[12] D. P. Dobkin and D. G. Kirkpatrick. Determining the separation of preprocessed polyhedra – a unified approach. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, volume 443 of *Lecture Notes Comput. Sci.*, pages 400–413. Springer-Verlag, 1990.

[13] P. F. Fisher. Algorithm and implementation uncertainty in viewshed analysis. *Internat. J. Geogr. Inform. Syst.*, 7:331–347, 1993.

[14] P. F. Fisher. Stretching the viewshed. In *Proc. 6th Internat. Sympos. Spatial Data Handling*, pages 725–738, 1994.

[15] P. F. Fisher. Reconsideration of the viewshed function in terrain modelling. *Geogr. Syst.*, 3:33–58, 1996.

[16] L. D. Floriani, B. Falcidieno, C. Pienovi, D. Allen, and G. Nagy. A visibility-based model for terrain features. In *Proc. 2nd Internat. Sympos. Spatial Data Handling*, pages 235–250, 1986.

[17] L. D. Floriani, G. Nagy, and E. Puppo. Computing a line-of-sight network on a terrain model. In *Proc. 5th Internat. Sympos. Spatial Data Handling*, pages 672–681, 1992.

[18] W. R. Franklin and C. K. Ray. Higher isn't necessarily better: Visibility algorithms and experiments. In *Proc. 6th Internat. Sympos. Spatial Data Handling*, pages 751–763, 1994.

[19] A. Gajentaan and M. H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory Appl.*, 5:165–185, 1995.

[20] M. J. Katz, M. H. Overmars, and M. Sharir. Efficient hidden surface removal for objects with small union size. *Comput. Geom. Theory Appl.*, 2:223–234, 1992.

[21] C. Knauer, M. van Kreveld, and E. Moet. Visibility maps of edges and triangles in 3d. *Manuscript*, 2005. `http://www.cs.uu.nl/~esther/visibilitymaps.pdf`

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

[23] J. Lee. Analyses of visibility sites on topographic surfaces. *Internat. J. Geogr. Inform. Syst.*, 5:413–430, 1991.

[24] D. Mount. Geometric intersection. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 33, pages 615–632. CRC Press LLC, Boca Raton, FL, 1997.

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

[26] J. H. Reif and S. Sen. An efficient output-sensitive hidden surface removal algorithm and its parallelization. In *Symposium on Computational Geometry*, pages 193–200, 1988.

[27] P. Sorensen and D. Lanter. Two algorithms for determining partial visibility and reducing data structure induced error in viewshed analysis. *Photogrammatic Engineering and Remote Sensing*, 28:1129–1132, 1993.

[28] J. Urrutia. Art gallery and illumination problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 973–1027. North-Holland, 2000.

[29] M. van Kreveld. Variations on sweep algorithms: efficient computation of extended viewsheds and classifications. In *Proc. 7th Internat. Sympos. Spatial Data Handling*, pages 13A.15–13A.27, 1996.

[30] M. van Kreveld, E. Moet, and R. van Oostrum. Region inter-visibility in terrains. In *Abstracts of the 20th European Workshop on Computational Geometry*, pages 155–158, 2004.