Terrain Visibility with Multiple Viewpoints *

Ferran Hurtado[1], Maarten Löffler[2], Inês Matos[1,4], Vera Sacristán[1], Maria Saumell[5], Rodrigo I. Silveira[3,1], and Frank Staals[2]

[1] Dept. Matemàtica Aplicada II, Universitat Politècnica de Catalunya, Spain
[2] Dept. of Information and Computing Sciences, Universiteit Utrecht, The Netherlands
[3] Dept. de Matemática, Universidade de Aveiro, Portugal
[4] Dept. de Matemática & CIDMA, Universidade de Aveiro, Portugal
[5] Département d'Informatique, Université Libre de Bruxelles, Belgium

Abstract. We study the problem of visibility in polyhedral terrains in the presence of multiple viewpoints. We consider three fundamental visibility structures: the visibility map, the colored visibility map, and the Voronoi visibility map. We study the complexity of each structure for both 1.5D and 2.5D terrains, and provide efficient algorithms to construct them. Our algorithm for the visibility map in 2.5D terrains improves on the only existing algorithm in this setting.

1 Introduction

Visibility problems or, to be more specific, problems regarding whether two objects are *visible* from each other amidst a number of obstacles have been a hot topic in computational geometry. In this paper we are interested in visibility on *terrains*. A *2.5D terrain* is an xy-monotone polyhedral surface in $\mathbb{R}^3$. We also study 1.5D terrains: x-monotone polygonal lines in $\mathbb{R}^2$. The obstacles we consider are the terrain edges or triangles themselves. A fundamental aspect of visibility in terrains is the *viewshed* of a point (i.e. the *viewpoint*): the (maximal) regions of the terrain that the viewpoint can see.

In a 1.5D terrain, the viewshed is almost equivalent to the *visibility polygon* of a viewpoint, so well-known linear-time algorithms can be applied. In 2.5D the viewshed is more complex (see Fig. 1). In an n-vertex terrain, the viewshed of a viewpoint can have $\Theta(n^2)$ complexity. The best algorithms known to compute it take $O((n + k) \log n \log\log n)$ time [13], and $O((n\alpha(n) + k) \log n)$ time [9], where k is the size of the resulting viewshed, and $\alpha(n)$ is the inverse of the Ackermann function.

While the computation of the viewshed from one viewpoint on a terrain has been thoroughly studied, it is surprising that a natural and important variant has been left open: What happens if instead of *one* single viewpoint, one has *many*, say $m > 1$, different

* F. H., V. S., and R.I. S. were partially supported by projects Gen. Cat. DGR 2009SGR1040, MINECO MTM2012-30951 and ESF EuroGIGA-ComPoSe - MICINN EUI-EURC-2011-4306. M. L. and F. S. were supported by Netherlands Organisation for Scientific Research (NWO) under project no 639.021.123 and 612.001.022. I. M. was supported by FCT project PEst-C/MAT/UI4106/2011 with COMPETE number FCOMP-01-0124-FEDER-022690. M. S. was supported by projects F.R.S.-FNRS - EUROGIGA NR 13604 and GAČR GIG/11/E023. R.I. S. was partially supported by the FP7 Marie Curie Actions Individual Fellowship PIEF-GA-2009-251235, and by FCT through grant SFRH/BPD/88455/2012.

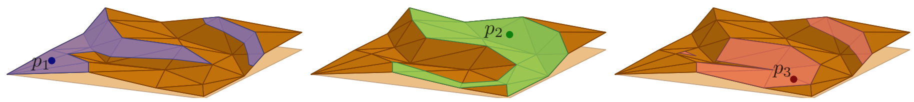


Fig. 1. The viewsheds of three viewpoints on a 2.5D terrain.

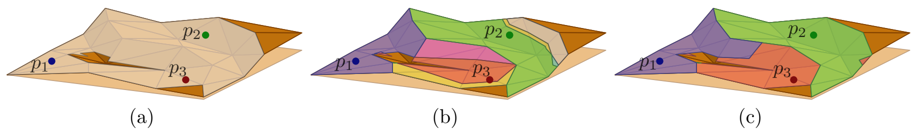


Fig. 2. The visibility map (a), the colored visibility map (b), and the Voronoi visibility map (c).

viewpoints on the terrain? The *common viewshed*, or *visibility map* can then be defined as the regions of the terrain that can be seen from *at least one* viewpoint. Computing the viewshed from each single viewpoint and then taking the union of the m viewsheds is a straightforward solution, but it has a high running time that does not take the final size of the visibility map into account. Obtaining more efficient algorithms for this and other related problems is the main focus of this paper.

To the best of our knowledge, there are no other studies in computational geometry on the visibility map of multiple viewpoints. We are not aware of any work for 1.5D terrains, whereas for 2.5D terrains we can only mention [6], where they essentially overlay the m individual viewsheds without studying the complexity of the visibility map. This results in the high running time of $O(m^2n^4)$. In addition, a few papers deal with the computation of viewsheds for multiple viewpoints for rasterized terrains [6, 11].

We would like to highlight the fact that it is not due to its lack of interest that visibility from multiple viewpoints has been overlooked up to now. Visibility in 1.5D terrains has been thoroughly studied from related perspectives, and in particular the problem of *placing* a minimum number of viewpoints to cover a terrain has received a lot of attention (e.g. [2, 3, 5, 7, 10]). Their theoretical interest and the fact that 1.5D terrains already pose a difficult challenge are the main motivation behind our work in that dimension.

Regarding 2.5D, the applications are too numerous to be detailed here, so we only present a few concrete examples. For instance, evaluating the effectiveness of a set of fire lookout towers [4], or identifying locations for placing wind turbines so they are not visible from "sensitive sites" like touristic points [12]. Finally, our results also apply to other contexts like sensor networks, in which wireless devices have to be placed on a terrain, and we have to measure the quality of the chosen device placement scheme [14]. The structures we study are particularly interesting within this context.

Problem Statement. A 2.5D terrain $\mathcal{T}$ consists of a set $V(\mathcal{T})$ of n vertices, a set $E(\mathcal{T})$ of $O(n)$ edges, and a set $F(\mathcal{T})$ of $O(n)$ faces. A 1.5D terrain $\mathcal{T}$ consists of a set $V(\mathcal{T})$ of n vertices and a set $E(\mathcal{T})$ of $n-1$ edges.

For any point p on the terrain $\mathcal{T}$ (either a 2.5D terrain or a 1.5D terrain), the *viewshed* of p on $\mathcal{T}$, denoted by $\mathcal{V}_{\mathcal{T}}(p)$, is the maximal set of points on $\mathcal{T}$ that are *visible* from p. A point q is visible from p if and only if the line segment $\overline{pq}$ does not intersect any point strictly below the terrain surface (intuitively, this corresponds to placing the viewpoints some small $\varepsilon > 0$ above the terrain). Note that our definition of visibility is symmetric, and that viewpoints have unlimited sight. The viewshed $\mathcal{V}_{\mathcal{T}}(\mathcal{P})$ of a set of viewpoints $\mathcal{P}$ is the set of points visible from at least one viewpoint in $\mathcal{P}$.

Given a set of viewpoints $\mathcal{P}$, we define the *Voronoi viewshed* $\mathcal{W}_{\mathcal{T}}(p, \mathcal{P})$ of a viewpoint $p \in \mathcal{P}$ as the set of points in the viewshed of p that are closer to p than to any other viewpoint that can see them. More precisely, $\mathcal{W}_{\mathcal{T}}(p, \mathcal{P}) = \mathcal{V}_{\mathcal{T}}(p) \cap \{x \mid x \in \mathcal{T} \wedge \mathit{closest}_{\mathcal{T}}(x, \mathcal{P}) = p\}$, where $\mathit{closest}_{\mathcal{T}}(x, \mathcal{P})$ denotes the closest (in terms of the Euclidean distance) viewpoint in $\mathcal{P}$ that can see a point x on $\mathcal{T}$.

We study three fundamental terrain visibility structures regarding multiple viewpoints for 1.5D and 2.5D terrains. These structures are illustrated in Fig. 2.

The *visibility map* $\mathrm{Vis}(\mathcal{T}, \mathcal{P})$ is a subdivision of the terrain $\mathcal{T}$ into a visible region $R_V = \mathcal{V}_{\mathcal{T}}(\mathcal{P}) = \bigcup_{p \in \mathcal{P}} \mathcal{V}_{\mathcal{T}}(p)$ and an invisible region $R_I = \mathcal{T} \setminus R_V$.

The *colored visibility map* $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$ is a subdivision of the terrain $\mathcal{T}$ into maximally connected regions R, each of which is covered by exactly the same subset of viewpoints $\mathcal{P}' \subseteq \mathcal{P}$. Each region R is a (maximally connected) subset of $\bigcap_{p \in \mathcal{P}'} \mathcal{V}_{\mathcal{T}}(p)$ and we have that $R \cap \bigcup_{p \in \mathcal{P} \setminus \mathcal{P}'} \mathcal{V}_{\mathcal{T}}(p) = \emptyset$.

The *Voronoi visibility map* $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$ is a subdivision of the terrain $\mathcal{T}$ into maximally connected regions, each of which is a subset of the Voronoi viewshed $\mathcal{W}_{\mathcal{T}}(p, \mathcal{P})$ of a viewpoint $p \in \mathcal{P}$.

We denote the *size*, that is, the total complexity of all its regions, of $\mathrm{Vis}(\mathcal{T}, \mathcal{P})$, $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$, and $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$, by k, k_c, and k_v, respectively.

For simplicity, we assume that $\mathcal{P}$ is a set of m viewpoints placed on terrain vertices, thus $m \leq n$. We consider this a reasonable assumption, since in most applications the number of terrain vertices is considerably larger than the number of viewpoints.

Results. We present a comprehensive study of the visibility structures defined above. We analyze the complexity of all the structures and propose algorithms to compute them. Our results are summarized in Table 1. Regarding 1.5D terrains, all our algorithms avoid computing individual viewsheds. $\mathrm{Vis}(\mathcal{T}, \mathcal{P})$ is computed in nearly optimal running time, while the algorithms for $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$ and $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$ are output-sensitive. Obtaining the latter algorithm, whose running time depends on k_c and k_v, was surprisingly challenging, and required using several subtle geometric properties of the problem.

As for 2.5D terrains, we prove with a careful analysis—interesting on its own—that the maximum complexity of $\mathrm{Vis}(\mathcal{T}, \mathcal{P})$ and $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$ is much less than the overlay

	1.5D Terrains		2.5D Terrains	
Structure	Max. size	Computation time	Max. size	Computation time
Vis	$\Theta(n)$	$O(n \log n)$	$O(m^3 n^2)$	$O(m(n\alpha(n) + k_c) \log n)$
ColVis	$\Theta(mn)$	$O(n + (m^2 + k_c) \log n)$	$O(m^3 n^2)$	$O(m(n\alpha(n) + k_c) \log n)$
VorVis	$\Theta(mn)$	$O(n + (m^2 + k_c) \log n + k_v(m + \log n \log m))$	$O(m^4 n^2)$	$O(m(n\alpha(n) + k_c) \log n)$

Table 1. Complexity and computation time of the three visibility structures.

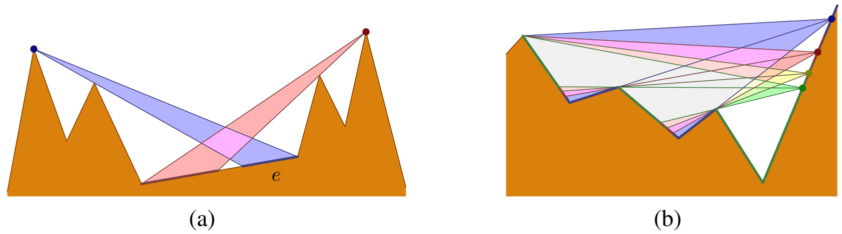


Fig. 3. (a) Edge e contains one invisible connected portion between two visible ones. (b) Every other edge has four different regions of $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$ and four different regions of $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$.

of the viewsheds, as implicitly assumed in previous work [6]. Using that, we show how a combination of well-known algorithms can be used to compute the visibility structures reasonably fast. Omitted proofs and details are given in the full version of this paper [8].

2 1.5D Terrains

2.1 Complexity of the Visibility Structures

In 1.5D our visibility structures can be seen as subdivisions of the x-axis into intervals.

Theorem 1. *Given a 1.5D terrain $\mathcal{T}$: $\mathrm{Vis}(\mathcal{T}, \mathcal{P})$ has maximum complexity $\Theta(n)$, and $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$ and $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$ both have maximum complexity $\Theta(mn)$.*

Proof (Sketch). There are two types of points of $\mathcal{T}$ that contribute to the complexity of $\mathrm{Vis}(\mathcal{T}, \mathcal{P})$: vertices of $\mathcal{T}$, and points where the $\mathcal{T}$ goes from visible to invisible or vice versa. There are n points of the first type. The points of the second type amount to $O(n)$, since it is easy to see that the interior of every edge $e \in E(\mathcal{T})$ contains at most two such points (see Fig. 3(a) for an example). Consequently, k is $\Theta(n)$.

As for $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$, notice that once a viewpoint sees a given point q on an edge $e \in E(\mathcal{T})$, it must see the whole segment from q to one of its endpoints. Hence, e can be split into at most $m + 1$ different regions of $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$. Therefore the complexity of $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$ is $O(mn)$. The example in Fig. 3(b) shows that this is tight.

Finally, let us focus on $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$. For a given edge $e \in E(\mathcal{T})$, $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$ restricted to e has at most $4m - 2$ regions (Lemma 1 of [8]), thus implying the upper bound. The lower bound can be achieved by a configuration of viewpoints on a particular terrain $\mathcal{T}$ that can be repeated so that every other edge of $\mathcal{T}$ has as many Voronoi regions as viewpoints, for arbitrary n and m. An example is shown in Fig. 3(b). $\square$

2.2 Algorithms to Construct the Visibility Structures

Construction of the Visibility Map. To construct the visibility map we first compute the left- and right-visibility maps, and then merge them. The *left(right)-visibility map* partitions $\mathcal{T}$ into two regions: the visible and the invisible portions of the terrain, where *visible* means visible from a viewpoint *to the left (right)* of that point of the terrain

In the following we explain the construction of the left-visibility map (thus, *visible* stands for *left-visible*). The algorithm uses the following property of 1.5D terrains, which is a consequence of the so-called *order claim* (See Claim 2.1 in [2]):

Observation 1 *Let $q \in \mathcal{T}$ be a point visible from the left by p_i and p_j, with p_i to the left of p_j. For any $r \in \mathcal{T}$ to the right of q, if p_i does not see r, then p_j cannot see r either.*

The algorithm sweeps the terrain from left to right while maintaining the leftmost visible viewpoint (if any), which we call the *active* viewpoint. The algorithm also stores a priority queue of events that comprises the x-coordinates of the vertices (*vertex events*) and the points of the terrain where a viewpoint becomes visible (*viewpoint events*). Initially we add an event for each terrain vertex and viewpoint, the latter corresponding to the position of the viewpoint on the terrain. We process the events sorted by their x-coordinate. When two events have the same x-coordinate, viewpoint events are processed first. Let p_a be the active viewpoint (if no viewpoint is visible, we set $p_a = \perp$).

(i) Viewpoint event, for a viewpoint p_i. If $p_a = \perp$, then a new visibility region starts. If $p_a = \perp$ or p_i is to the left of p_a, then p_i becomes the active viewpoint.

(ii) Vertex event, for a vertex v. If the active viewpoint p_a becomes invisible after v, we compute where p_a becomes visible again by a ray-shooting query, and add a viewpoint event there. If there was a viewpoint event at v as well, this viewpoint becomes the active viewpoint. Otherwise, the current visibility region ends at v.

The correctness of the method follows from Obs. 1, which guarantees that it is enough to keep track of only the leftmost visible viewpoint. The following theorem is proved in the full version.

Theorem 2. *Given a 1.5D terrain $\mathcal{T}$, the visibility map* $\mathrm{Vis}(\mathcal{T}, P)$ *can be constructed in $O(n \log n)$ time.*

Construction of the Colored Visibility Map. The computation of the colored visibility map is similar to that of $\mathrm{Vis}(\mathcal{T}, P)$, with the extra complication of having to maintain all visible viewpoints during the sweep. We show in the full version that we can still handle each event in $O(\log n)$ time. In principle, the event processing time can be charged to the output size k_c –since each viewpoint is likely to generate a new region when it reappears. However, it can happen that several viewpoints reappear at exactly the same point, generating a single region in ColVis. With some analysis we show that the total number of these situations is $O(m^2)$, leading to the following result.

Theorem 3. *Given a 1.5D terrain $\mathcal{T}$, the colored visibility map* $\mathrm{ColVis}(\mathcal{T}, P)$ *can be constructed in $O(n + (m^2 + k_c) \log n)$ time.*

Construction of the Voronoi Visibility Map.

Divide and Conquer Approach. A way to construct $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$ consists in dividing the set of viewpoints into two subsets, computing the Voronoi visibility map of the two subsets recursively, and merging the two maps. This takes $O(mn \log m)$ time.

An Output-Sensitive Algorithm. Even though $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$ can have $\Theta(mn)$ complexity, it seems unlikely that such high complexity arises often in practical applications. In the following we present an alternative algorithm that essentially extracts the Voronoi visibility map from the colored visibility map. Its running time depends on the complexity of the two structures, and avoids the fixed $O(mn)$ term of the previous method.

The algorithm sweeps the terrain from left to right. During this sweep, we maintain three data structures: (i) a doubly-linked list with the vertices of $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$, ordered from left to right, (ii) a list $\mathcal{P}'$ with the currently visible viewpoints, and (iii) for each $p_i \in \mathcal{P}'$, the starting point a_i of the last region in which p_i is visible encountered so far in the sweep. We will use $\mathcal{T}[a, c]$, for a, c on $\mathcal{T}$ and $x(a) < x(c)$, to denote the closed portion of the terrain between a and c. The algorithm produces $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$ as a list of interval, viewpoint pairs $([a, c], p_i)$, such that p_i is the closest viewpoint to all points in $\mathcal{T}[a, c]$. If $\mathcal{T}[a, c]$ is not visible from any viewpoint, p_i is set to $\perp$.

Our algorithm uses the following two functions, whose implementation is described later. $\textsc{IsAlwaysCloser}([a, c], p_1, p_2)$ determines whether p_1 is always closer than p_2 in $\mathcal{T}[a, c]$, assuming both viewpoints are visible throughout $\mathcal{T}[a, c]$. $\textsc{FirstRegionChange}([a, c], p_1, \mathcal{P}')$ assumes that p_1 is visible throughout $\mathcal{T}[a, c]$ and is the closest visible viewpoint at a; it returns the leftmost point in $\mathcal{T}[a, c]$ where p_1 stops being the closest visible viewpoint from $\mathcal{P}'$ (or the end of the interval, if that never happens).

We process $\mathcal{T}$ in a number of iterations. Each iteration starts at the leftmost point u of a new Voronoi region, with $\mathcal{P}'$ containing the viewpoints that are visible from u.

If $\mathcal{P}' = \emptyset$, then the region starting at u and ending at the start point v of the next region in $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$ is not visible from any viewpoint. We report the region $[u, v]$ with $\perp$, and move forward (towards the right) until v, where a new Voronoi region, and thus a new iteration, starts.

If $\mathcal{P}' \neq \emptyset$, we compute the closest visible viewpoint in $O(m)$ time; if there is more than one, we move infinitesimally to the right of u, and compute the closest visible viewpoint there. Without loss of generality, we assume that the closest visible viewpoint is p_1. For all viewpoints $p_i \in \mathcal{P}'$, we set $a_i := u$. We now start traversing the terrain, from u towards the right. At a point q, we might find several events from ColVis:

1. A viewpoint p_j becomes visible. We update $\mathcal{P}'$, set $a_j := q$, and continue the sweep.
2. A viewpoint $p_j \neq p_1$ becomes invisible. We update $\mathcal{P}'$ and proceed depending on two subcases:
   (a) $\textsc{IsAlwaysCloser}([a_j, q], p_1, p_j)$ = $\textsc{True}$. Continue traversing the terrain.
   (b) $\textsc{IsAlwaysCloser}([a_j, q], p_1, p_j)$ = $\textsc{False}$. There is a point in $\mathcal{T}[a_j, q]$ at which p_j is closer than p_1, so at least one Voronoi region starts between u and q. We find the leftmost region change v by calling $\textsc{FirstRegionChange}([u, q], p_1, \mathcal{P}')$, and report $[u, v]$ as a Voronoi region with p_1 as closest point. We now backtrack our sweep, i.e. we traverse the terrain from right to left (updating $\mathcal{P}'$ as we encounter events), until we reach v, and start a new Voronoi region, and thus a new iteration of our algorithm at v.
3. Viewpoint p_1 becomes invisible. We update $\mathcal{P}'$, and compute $\textsc{IsAlwaysCloser}$ $([a_i, q], p_1, p_i)$, for all $p_i \in \mathcal{P}'$. If the answer is $\textsc{True}$ for *all* viewpoints in $\mathcal{P}'$, we report the region $[u, q]$ with p_1 as closest viewpoint, and start a new Voronoi region

and a new iteration at q. Otherwise, there is at least one Voronoi region that starts between u and q. We handle this analogously to case 2(b).

After processing the events of type 2 at the rightmost vertex of the terrain, we have successfully computed $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$. Since we backtrack our sweep in step 2, it may be the case that we (unnecessarily) visit events from $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$ multiple times. We can avoid this, by augmenting this step as follows. Consider step 2a. We notice that there cannot be a Voronoi region of p_j between a_j and q (since at least p_1 is closer and visible). So we can remove the events of p_j becoming visible at a_j and invisible at q from $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$. We remove q in step 2a itself. Event a_j is removed if we encounter it while backtracking in step 2b: at each event of type 1, i.e. a viewpoint p_j becoming visible, we check if p_j is in $\mathcal{P}'$. If not, we must have removed its corresponding endpoint (i.e. q) from $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$. Thus we can also remove a_j.

As for the auxiliary functions, $\textsc{IsAlwaysCloser}([a_j, q], p_1, p_j)$ can be implemented to run in $O(\log n)$ time by doing a ray-shooting query, where the ray is the bisector of p_1 and p_j. However, it is possible to answer this question faster.

Lemma 1. *Consider two points r and t such that all of $\mathcal{T}[r, t]$ is visible from two viewpoints p_1 and p_2. We can decide whether there exists some point in $\mathcal{T}[r, t]$ that is closer to p_2 than to p_1 in $O(1)$ time.*

$\textsc{FirstRegionChange}([u, q], p_1, \mathcal{P}')$ can be implemented to run in $O(m \log n)$ time as follows: For every $p_i \in \mathcal{P}'$, and using a ray-shooting query, compute the leftmost point (if any) on $\mathcal{T}[a_i, q]$ that is closer to p_i than to p_1. Then keep the leftmost point u' among all the points encountered. Again, it is possible to do this faster:

Lemma 2. *Let $[u, q]$ be an interval such that $p_1 \in \mathcal{P}$ is visible in all $\mathcal{T}[u, q]$ and is the closest visible viewpoint at u. Let $\mathcal{P}'$ be a set of viewpoints such that for each $p_i \in \mathcal{P}'$, $\mathcal{T}[a_i, q]$ is visible from p_i, for some a_i such that $x(u) \leq x(a_i)$. Then in $O(m + \log n \log m + n')$ time we can find the leftmost point $u' \in \mathcal{T}[u, q]$ such that at u' there is a change of region in $\mathrm{VorVis}(\mathcal{T}, \mathcal{P}')$, for n' the number of vertices in $\mathcal{T}[u, u']$.*

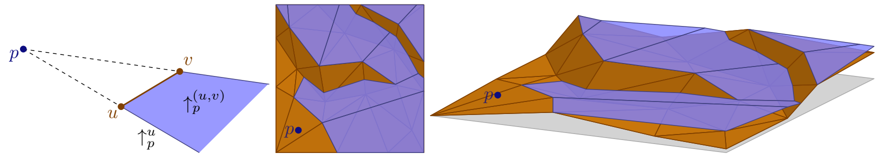
The proofs can be found in the full version. We obtain:

Theorem 4. *Given a 1.5D terrain $\mathcal{T}$, the Voronoi visibility map $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$ can be computed in $O(n + (m^2 + k_c) \log n + k_v(m + \log n \log m))$ time.*

3 2.5D Terrains

3.1 Complexity of the Visibility Structures

Proposition 1. *The visibility map $\mathrm{Vis}(\mathcal{T}, P)$ of a 2.5D terrain $\mathcal{T}$ can have complexity $\Omega(m^2 n^2)$.*



(a) (b) (c)

Fig. 5. (a) A ray and a vase. (b) The top-down view of a terrain $\mathcal{T}$ with a single viewpoint p. The domain is decomposed in the viewshed $\mathrm{Vis}(\mathcal{T}, p)$ and a collection of vases. (c) a 3D view of $\mathcal{T}$ and the vases of p.

Proof. We present a terrain that consists of a flat (horizontal) rectangle, the *courtyard*, surrounded by a thin wall. We make $O(n)$ (almost) vertical incisions, or *windows*, in the northern and western wall. We place half our viewpoints behind windows in the northern wall, and the other half behind windows in the western wall. Each viewpoint is placed so that it can see through $O(n)$ windows into the courtyard, see Fig. 4. It follows that the visibility map inside the courtyard forms an $O(mn) \times O(mn)$ grid. $\square$

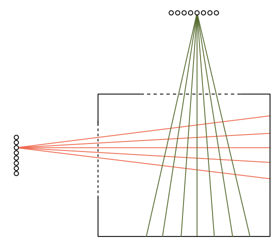


Fig. 4. Viewpoints are shown as white circles and rays indicate the part of the terrain visible from the viewpoint.

In order to establish an upper bound on the complexity of the visibility maps, we start with the most general case, in which $\mathcal{T}$ is actually an arbitrary polyhedron.

Let $\mathcal{M}$ be a polyhedron, let v be a vertex of $\mathcal{M}$, and let $p \in \mathcal{P}$ be a viewpoint. We define the *ray* of p and v, denoted $\uparrow_p^v$, to be the half line that starts at v and has vector $\overrightarrow{pv}$. Similarly, let $p \in \mathcal{P}$ be a point and $e = \overline{uv}$ be an edge of $\mathcal{M}$. The *vase* of p and e, denoted $\uparrow_p^e$, is the region in $\mathbb{R}^3$ bounded by e, $\uparrow_p^u$, and $\uparrow_p^v$ (see Fig. 5(a)). The set of all vases originating from p is denoted $\Uparrow(p) = \{\uparrow_p^e \mid e \in E(\mathcal{M})\}$. Assuming general position, we have:

Observation 2 $\mathrm{Vis}(\mathcal{M}, \mathcal{P})$ *can have three types of vertices: (1) vertices of $\mathcal{M}$, (2) intersections between an edge of $\mathcal{M}$ and a vase, and (3) intersections between a triangle of $\mathcal{M}$ and two vases.*[6]

Theorem 5. *The visibility map* $\mathrm{Vis}(\mathcal{M}, \mathcal{P})$ *of a polyhedron* $\mathcal{M}$ *has complexity* $O(m^2n^3)$.

Proof. Each vase comes from a viewpoint in $\mathcal{P}$ and an edge in $E(\mathcal{M})$. Clearly, $|V(\mathcal{M})|$, $|E(\mathcal{M})|$, $|F(\mathcal{M})| \in O(n)$. So, the number of vertices of type (1), (2), and (3) is at most $O(n)$, $O(mn^2)$, and $O(m^2n^3)$, respectively. $\square$

Next, we show that if $\mathcal{M}$ is a terrain, then the number of vertices of type (3) can only be $O(m^3n^2)$. Given any object $B \in \mathbb{R}^3$, we denote by $\underline{B}$ the vertical projection of B to $\mathbb{R}^2$. Furthermore, we define $S_1 \oplus S_2$ to be the overlay of subdivisions S_1 and S_2. Let $\uparrow_s^e$

[6] It is worth noting that there is a fourth possible type of vertex: an intersection of three vases. However, such a vertex does not lie on $\mathcal{M}$, so it does not appear in the visibility map.

and $\uparrow_t^f$ be two vases. The intersection of these two vases is a line segment (or half line), which we denote by ${}_s^e\!\times_t^f$. We call this a *pyramid ray*.

Observation 3 *Consider k planar subdivisions $S_1, S_2, \ldots, S_k$, and let $\mathbb{S} = \bigoplus_{i=1}^{k} S_i$ be their overlay. Any line ℓ has at most $O(\sum_{i=1}^{k} |S_i|)$ intersections with $\mathbb{S}$.*

Lemma 3. *Let R be the set of pyramid rays created by $\mathcal{P}$ on a 2.5D Terrain $\mathcal{T}$. Every edge $\underline{e} \in \underline{E(\mathcal{T})}$ intersects at most $O(m^3 n)$ rays from $\underline{R}$.*

Proof. Let $\underline{X}_i$ be the subdivision of $\mathbb{R}^2$ that we obtain by vertically projecting the upper envelope of $\mathcal{T}$ and all vases in $\Uparrow(p_i)$. Fig. 5(b) shows an example. Any pyramid ray $r \in R$ is the intersection of one vase from $\Uparrow(p_i)$ and one vase from $\Uparrow(p_j)$, for some $i \neq j$. This means that $\underline{r}$ is contained in a cell of $\underline{X}_i \oplus \underline{X}_j$. Let $\mathbb{X} = \bigoplus_{i=1}^{m} \underline{X}_i$. Then each cell in $\mathbb{X}$ is contained in at most m different projected vases, hence, it contains at most $\binom{m}{2}$ (pieces of) projected pyramid rays.

There are $O(n)$ vases in $\Uparrow(p_i)$, so $\underline{X}_i$ has $O(n)$ vertices. From Obs. 3 it follows that any line —and therefore any edge $\underline{e} \in \underline{E(\mathcal{T})}$— intersects the edges of $\mathbb{X}$ at most $O(mn)$ times. This means $\underline{e}$ intersects at most $O(mn)$ cells, and therefore also at most $O(m^3 n)$ pyramid rays in $\underline{R}$. $\square$

Lemma 4. *$\mathrm{Vis}(\mathcal{T}, \mathcal{P})$ contains at most $O(m^3 n^2)$ vertices of type (3).*

Proof. We split the vertices of $\mathrm{Vis}(\mathcal{T}, \mathcal{P})$ of type (3) into two subtypes. Each vertex v (of type (3)) is associated with one pyramid ray r. Now, v is either of type (3)a, if it is the highest vertex on r, or of type (3)b otherwise. The number of vertices of type (3)a is at most $O(m^2 n^2)$, since there is at most one per ray and there are only $O(m^2 n^2)$ rays. We now show the number of vertices of type (3)b is at most $O(m^3 n^2)$.

Let v be a vertex of $\mathrm{Vis}(\mathcal{T}, \mathcal{P})$ of type (3)b. It is the intersection of a ray r and a triangle $t \in F(\mathcal{T})$. Since v is not the highest vertex on r, there must be another vertex w on r. Clearly, w cannot lie on t, so $\underline{w}$ must lie outside $\underline{t}$, while $\underline{v}$ lies inside $\underline{t}$. Thus there must be an edge $e \in E(P)$ such that $\underline{r}$ crosses $\underline{e}$. We charge v to this intersection between $\underline{r}$ and $\underline{e}$. Clearly, any such intersection gets charged at most once. By Lemma 3, there are at most $O(m^3 n^2)$ such intersections in total. Hence, the number of vertices of type (3)b is also at most $O(m^3 n^2)$. $\square$

Theorem 6. *$\mathrm{Vis}(\mathcal{T}, \mathcal{P})$, for $\mathcal{T}$ a 2.5D terrain, has complexity $O(m^3 n^2)$.*

The visibility map $\mathrm{Vis}(\mathcal{T}, \mathcal{P})$ corresponds to the union over $\mathcal{P}$ of the viewsheds of the individual viewpoints. Similarly, the colored visibility map corresponds to the overlay of the viewsheds of the individual viewpoints in $\mathcal{P}$. Therefore, Obs. 2 also holds for the vertices of $\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$. This implies the following result.

Theorem 7. *$\mathrm{ColVis}(\mathcal{T}, \mathcal{P})$, for $\mathcal{T}$ a 2.5D terrain, has complexity $O(m^3 n^2)$.*

Finally, we are interested in the Voronoi visibility map. $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$ can have additional types of vertices: intersections of Voronoi edges with terrain triangles. We use *power diagrams*: Let $\mathcal{C} = C_1, .., C_m$ be a set of m circles in $\mathbb{R}^2$, and let c_i and r_i denote the center and radius of C_i, respectively. The (2D) power diagram $\mathrm{PD}(\mathcal{C})$ is the subdivision of $\mathbb{R}^2$ into m regions, one for each circle, such that $R_i = \{x \in \mathbb{R}^2$ s. t., for all $j \in \{1,..,m\}, pow(C_i,x) \leq pow(C_j,x)\}$, where $pow(C_i,x) = d_2(c_i,x)^2 - r_i^2$ (and $d_2(\cdot,\cdot)$ denotes the Euclidean distance in $\mathbb{R}^2$). The (2D) power diagram of m circles has complexity $O(m)$ and can be computed in $O(m \log m)$ time [1].

Let $\mathrm{VD}(\mathcal{P})$ denote the 3-dimensional Voronoi diagram of $\mathcal{P}$. We observe that the restriction of $\mathrm{VD}(\mathcal{P})$ to any single plane H in $\mathbb{R}^3$ corresponds to a *power diagram* $\mathrm{PD}(\mathcal{C}_\mathcal{P})$ in $\mathbb{R}^2$: Assume without loss of generality that H is a horizontal plane at $z = 0$, and let $\xi \geq \max_{p \in \mathcal{P}} p_z^2$ be some large value. Any point $a \in H$ is closer to $p \in \mathcal{P}$ than to $q \in \mathcal{P}$ if (and only if) $d(a,p) = d_3(a,p) \leq d_3(a,q)$, and hence if $d_3(a,p)^2 \leq d_3(a,q)^2$. Using that $a_z = 0$ we can rewrite this to $d_2(a,\underline{p})^2 - (\xi - p_z^2) \leq d_2(a,\underline{q}) - (\xi - q_z^2)$. So if we introduce a circle C_p in $\mathcal{C}_\mathcal{P}$ for every viewpoint p with center $\underline{p}$ and radius r_p such that $r_p^2 = \xi - p_z^2$ then we get that a is closer to p than to q if and only if $pow(C_p,a) \leq pow(C_q,a)$. Thus, we can prove:

Theorem 8. $\mathrm{VorVis}(\mathcal{T},\mathcal{P})$*, for* $\mathcal{T}$ *a 2.5D terrain, has complexity* $O(m^4n^2)$.

3.2 Algorithms to Construct the Visibility Structures

Computing the (Colored) Visibility Map. Katz et al. [9] developed an $O((n\alpha(n)+k)\log n)$ time algorithm to compute the viewshed of a single viewpoint, where k is the output complexity and $\alpha(n)$ is the extremely slowly growing inverse of the Ackermann function. Coll et al. [6] use this algorithm to compute the visibility map of a 2.5D terrain in $O(m^2n^4)$ time and space. Essentially they project the individual viewsheds onto $\mathbb{R}^2$, and construct the overlay $\mathbb{V} = \bigoplus_{p \in \mathcal{P}} \underline{\mathcal{V}_\mathcal{T}(p)}$ (see Fig. 6). It is then easy to construct the (colored) visibility map from $\mathbb{V}$. We use the same approach. However, using our observations from the previous section, we show that even if the viewsheds have complexity $\Theta(n^2)$, we can compute the (colored) visibility map in $O(m^4n^2 \log n)$ time.

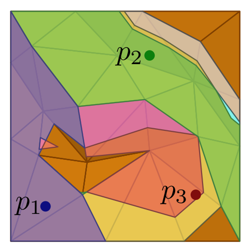


Fig. 6. Overlay $\mathbb{V}$.

Lemma 5. *Given a 2.5D terrain* $\mathcal{T}$ *with* n *vertices and a set* $\mathcal{P}$ *of* m *viewpoints. The planar subdivision* $\mathbb{V}$ *can be constructed in* $O(m(n\alpha(n) + k_c)\log n)$ *time.*

Theorem 9. *Both the visibility* $\mathrm{Vis}(\mathcal{T},\mathcal{P})$ *and the colored visibility map* $\mathrm{ColVis}(\mathcal{T},\mathcal{P})$*, for* $\mathcal{T}$ *a 2.5D terrain can be computed in* $O(m(n\alpha(n) + k_c)\log n)$ *time.*

Computing the Voronoi Visibility Map. Let F be a face of the colored visibility map $\mathrm{ColVis}(\mathcal{T},\mathcal{P})$, and let $\mathcal{P}_F$ denote the set of viewpoints that can see F. For each such face F we compute the intersection of F with the $\mathrm{VD}(\mathcal{P}_F)$. We do this via the power diagram: i.e. consider the plane H containing F, and compute the power diagram on H with respect to the the viewpoints in $\mathcal{P}_F$. This takes $O(k_c m \log m)$ time in total, since $\mathrm{ColVis}(\mathcal{T},\mathcal{P})$ has $O(k_c)$ faces, and each power diagram can be computed in $O(m \log m)$ time. Each power diagram is constrained to a single face, so we glue all of them together and project the result onto $\mathbb{R}^2$. This yields a subdivision $\mathbb{W}$ of size $O(k_c m)$. We now compute $\mathbb{V}$ in $O(m(n\alpha(n) + k_c)\log n)$ time (as described above), and overlay it with $\mathbb{W}$ in $O(k_c m + k_c + k_v) = O(k_c m)$ time. Hence:

Theorem 10. *The Voronoi visibility map* $\mathrm{VorVis}(\mathcal{T}, \mathcal{P})$*, for* $\mathcal{T}$ *a 2.5D terrain, can be computed in* $O(m(n\alpha(n) + k_c)\log n)$ *time.*

4 Final remarks

We studied visibility with multiple viewpoints on polyhedral terrains for the first time. Our results show that considering multiple viewpoints converts classical visibility problems into much more challenging ones, even for 1.5D terrains.

Moreover, our results lead to many intriguing questions. For 1.5D terrains, is there an efficient algorithm to construct the Voronoi visibility map whose running time does not depend on k_c? In 2.5D, the worst-case complexities are not tight; it would be interesting to close those gaps. Algorithmically, in 2.5D the main challenge is to find an algorithm to construct the structures directly, avoiding the computation of the individual viewsheds. Finally, an interesting and realistic extension is when viewpoints have *limited* sight (i.e. can only see up to a certain distance). We discuss this extensively in the full version.

References

1. Aurenhammer, F.: Power diagrams: Properties, algorithms and applications. SIAM J. Comput. 16(1), 78–96 (1987)
2. Ben-Moshe, B., Katz, M., Mitchell, J.: A constant-factor approximation algorithm for optimal 1.5D terrain guarding. SIAM J. Comput. 36(6), 1631–1647 (2007)
3. Bose, P., Shermer, T., Toussaint, G., Zhu, B.: Guarding polyhedral terrains. Comput. Geom. 7(3), 173–185 (1997)
4. Catry, F., Rego, F., Santos, T., Almeida, J., Relvas, P.: Fires prevention in Portugal - using GIS to help improving early fire detection effectiveness. In: 4th Int. Wildland Fire Conf. (2007)
5. Cole, R., Sharir, M.: Visibility problems for polyhedral terrains. J. Symbolic Comput. 7(1), 11–30 (1989)
6. Coll, N., Madern, N., Sellarès, J.A.: Good-visibility maps visualization. Vis. Comput. 26(2), 109–120 (2010)
7. Gibson, M., Kanade, G., Krohn, E., Varadarajan, K.: An approximation scheme for terrain guarding. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, LNCS, vol. 5687, pp. 140–148. Springer (2009)
8. Hurtado, F., Löffler, M., Matos, I., Sacristán, V., Saumell, M., Silveira, R.I., Staals, F.: Terrain visibility with multiple viewpoints (2013), arXiv:1309.4323 [cs.CG]
9. Katz, M.J., Overmars, M.H., Sharir, M.: Efficient hidden surface removal for objects with small union size. Comput. Geom. 2, 223–234 (1992)
10. King, J., Krohn, E.: Terrain guarding is NP-hard. SIAM J. Comput. 40(5), 1316–1339 (2011)
11. Lv, P., Zhang, J., Lu, M.: An optimal method for multiple observers sitting on terrain based on improved simulated annealing techniques. In: Advances in Applied Artificial Intelligence, LNCS, vol. 4031, pp. 373–382. Springer (2006)
12. Möller, B.: Changing wind-power landscapes: regional assessment of visual impact on land use and population in Northern Jutland, Denmark. Appl. Energ. 83(5), 477–494 (2006)
13. Reif, J.H., Sen, S.: An efficient output-sensitive hidden surface removal algorithm and its parallelization. In: Proc. 4th Symp. Computational Geometry. pp. 193–200. ACM (1988)
14. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw. 52(12), 2292 – 2330 (2008)