International Journal of Computational Geometry & Applications

Computing the Set of All the Distant Horizons of a Terrain*

Daniel Archambault, Willam Evans, and David Kirkpatrick
Department of Computer Science, The University of British Columbia, 201-2366 Main Mall
Vancouver, British Columbia, V6T 1Z4, Canada
{*archam, will, kirk*}*@cs.ubc.ca*

Received September 23, 2004
Revised March 17, 2005

We study the problem of computing the set of all distant horizons of a terrain, represented as either: (1) the set of all edges that appear on the distant horizon from at least one viewing direction; (2) for every edge e, the set of direction intervals for which e appears on the distant horizon; or (3) a search structure to query for the edges on the distant horizon, or the precise distant horizon, from a fixed viewing direction. We describe an algorithm that solves the first and second forms of the problem in $O(n^{2+\epsilon})$ time for any constant $\epsilon > 0$ where n is the number of edges of the terrain. This algorithm can be extended to compute a search structure for (3) in $O(n^{2+\epsilon})$ time. The search structure can return the s edges on the distant horizon in $O(\log n + s)$ time. We show solving problem (1) is 3SUM hard. Furthermore, we construct a terrain with a single local maximum in which $\Theta(n)$ edges each have $\Theta(n)$ direction intervals, showing that our solution to (2) cannot be significantly improved, in the worst case, even for such restricted terrains. This takes advantage of a novel construction in which the convex hull of a set of n linearly moving points, whose trajectories do not intersect, changes $\Omega(n^2)$ times.

Keywords: horizons; orthographic views

1. Introduction

A *polyhedral terrain* is a surface $\{(x, y, z)|z = f(x, y)\}$ where $f(x, y)$ is a piecewise-linear function defined over some polygonal region in $\Re^2$. It is typically represented using a polyhedral mesh composed of faces, edges, and vertices. The value $f(x, y)$ is referred to as the *height* or *altitude* of the terrain at the point (x, y). Intuitively, a *distant horizon* of a terrain is the boundary between the sky and the terrain from an orthographic projection of the terrain specified by some horizontal direction. More formally, point p on an edge e appears on the distant horizon if and only if it supports a horizontal line l that does not contain any point below the terrain. Such a line l is called a horizontal *visual line* in the visual hull literature (Laurentini[14] and Petitjean[15]), so we will reuse the term in this paper.

*An earlier version of this paper[4] appeared in the proceedings of CCCG 2004.

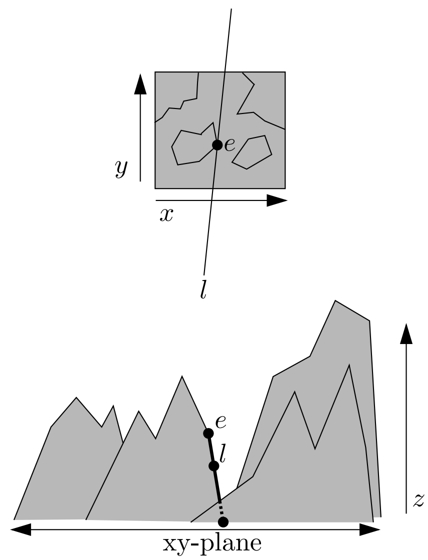

Fig. 1. The horizontal line l is tangent to the edge e (in bold) and does does not contain any point below the terrain. Thus, the edge e is on the distant horizon of the terrain from the horizontal viewing direction l.

Our goal is to compute a succinct representation for the set of all possible distant horizons of a terrain. One motivation is to obtain a representation of the terrain data that is faster to render[a], but that correctly reproduces the silhouette of the terrain in any distant orthographic projection. Identifying those edges that do not appear on any distant horizon, and hence never need to be rendered, provides a kind of global simplification. Alternatively, we might want to preprocess the terrain edges for distant horizon queries specified by a viewing direction. Specifically, we could predetermine, for each terrain edge e, the direction intervals for which e contributes to the distant horizon. By recovering and rendering the appropriate edges, we generate an accurate orthographic view for any specified horizontal projection direction. We therefore consider three versions of the distant horizon problem:

(1) The problem of computing, for every terrain edge e, whether or not e contributes to the distant horizon for at least one horizontal viewing direction.
(2) The problem of computing, for every terrain edge e, the set of direction intervals, the *distant horizon direction intervals*, for which e appears on the distant horizon[b].
(3) The problem of computing a query data structure that returns the set of terrain

[a]The rendering of edges could take the form of the rendering of trapezoidal strips extending downward from each edge.
[b]A solution to this problem would consist of a list of direction intervals for each edge, for which it appears on the distant horizon.

edges that appear on the distant horizon from a specified horizontal viewing direction.

The paper is structured as follows. In section 2, we present some previous and related work on terrains and horizons. In section 3, we present an algorithm that runs in $O(n^{2+\epsilon})$ time for any constant $\epsilon > 0$ where n is the number of edges of the terrain that solves version 2 (and version 1, since it is less general) of our problem. In section 4, we describe a solution to version 3 of our problem that can be constructed in $O(n^{2+\epsilon})$ time for any constant $\epsilon > 0$, which can return the set of s edges that form the distant horizon from a given horizontal viewing direction in $O(\log n + s)$ time. In section 5, we show that even version 1 of our problem is 3SUM hard, which suggests that sub-quadratic algorithms for this version are unlikely to exist. In section 6, we show that there exists a terrain with only a single local maximum and $\Omega(n^2)$ distant horizon direction intervals in total, demonstrating that sub-quadratic algorithms for version 2 of our problem do not exist, even for such restricted terrains.

2. Previous and Related Work

In this section, we discuss some of the previous and related work on extracting horizons from polyhedral terrains from single and multiple view points. We also describe work that has been done on bounding the number of topologically distinct views of a terrain from all viewing directions above the terrain.

2.1. *Horizons*

From a single view point, computing the horizon of a terrain is equivalent to projecting its n edges onto a viewing plane and computing the upper envelope of the n resulting line segments. Cole and Sharir[7] showed that a single horizon has near linear, worst case complexity $\Theta(n\alpha(n))$ where $\alpha(n)$ is the inverse Ackermann function. Many algorithms already exist that compute the upper envelope of n line segments with near optimal and optimal, worst-case running times. Atallah[5] describes a divide-and-conquer scheme to compute the upper envelope of n line segments in $O(n\alpha(n)\log n)$ time. The algorithm recursively divides the set of line segments into halves and pairwise merges the resulting upper envelopes using a sweep line technique. Hershberger[13] improves the running time of the algorithm to $O(n \log n)$ by ensuring the line segments in the subsets of Atallah's algorithm produce upper envelopes of linear complexity at many levels of the recursion tree. De Floriani and Magillo[10] describe a randomized algorithm that computes the horizon of a terrain in $O(n\alpha(n)\log n)$ expected time. In this algorithm, edges are inserted in logarithmic time into a data structure that has an expected size of $O(n\alpha(n))$. The algorithm is on-line and can support the insertion and deletion of edges as the viewpoint changes or the terrain changes level of detail.

Stewart[17] describes an algorithm that can compute the approximate horizon from all n vertices of a terrain. These horizons can be used for terrain shading applications. The problem is solved by dividing the view around each of the n vertices into s sectors where the horizon within each sector is assumed to be constant and equal to the point of maximum elevation in the sector. This algorithm runs in $O(n \log^2 n + ns)$ time, however, it produces accurate horizons only if s is chosen sufficiently large.

2.2. *Distinct Views of a Terrain*

There has been some work done on bounding the number of combinatorially distinct views of a terrain from all viewing directions above the terrain. Two views are distinct if the two sets of edges, faces, vertices, and intersections visible in the two projections from the two viewing directions are distinct. The best known lower bound, presented by de Berg et al.[8], is $\Omega(n^5\alpha(n))$. The best known upper bound, presented by Agarwal and Sharir[3], is $O(n^{5+\epsilon})$ for any constant $\epsilon > 0$. The proof of the upper bound uses a technique originally presented by Halperin and Sharir[12] that is similar to a technique that we use and that we now discuss in detail.

For each of the n terrain edges, Halperin and Sharir dualize the other $n-1$ edges to form $n-1$ bivariate functions and then compute their lower envelope. For each edge of the terrain e_0, the dualization is performed by selecting all other edges $e \neq e_0$ and parameterizing the set of all lines l tangent to $\overline{e_0 e}$. The dual space is a (t, k, ζ)-space, where t is the distance from the point of tangency on e_0 to a designated end point on e_0, k is $\tan\theta$ where θ is the angle between the xy-projection of the line l and the x-axis, and ζ is $-\cot\phi$ where ϕ is the angle between l and the positive z-axis. A fixed value for t and k defines a unique vertical plane in primal and there is at most one line l in the plane (if there is one at all) between any (e, e_0) pair. This line has a single value for ζ, and, therefore, each tangent line can be expressed as a function of t and k in (t, k, ζ)-space. The crucial observation is that a point of the function is on the lower envelope in (t, k, ζ)-space if and only if its tangent line in primal lies above all other tangent lines for a fixed t and k. A vertex of the lower envelope corresponds to a line that is tangent to e_0 and three other edges of the terrain. These lines that are tangent to four edges simultaneously are important in bounding the number of distinct views of the terrain, and can be computed by n upper envelope computations each taking $O(n^{2+\epsilon})$ expected time using the algorithm of Boissonnat[6] or Sharir[16] for any constant $\epsilon > 0$ or deterministically with the same runtime using the algorithm of Agarwal et al.[2]. In this paper, we employ a similar but simpler duality transform. Since we only consider horizontal viewing directions, we only require a constant number of envelope computations.

For the upper envelope calculation with the stated bounds to apply, we need to ensure that the bivariate functions have certain properties. These properties, described by Halperin and Sharir[12], are as follows:

i) Each surface patch is monotone in the xy-direction (a vertical line intersects the surface in at most one point). The surface is a portion of an algebraic surface of constant maximum degree b.

ii) The vertical projection of the surface onto the xy-plane is a planar region bounded by a constant number of algebraic arcs of constant maximum degree (say b too).

iii) The relative interiors of any triple of surfaces intersect in at most two points.

iv) The surface patches are in general position.

3. Computing the Set of All Distant Horizon Direction Intervals

We now describe an algorithm that can compute the set of distant horizon direction intervals associated with each terrain edge. The algorithm transforms each of the n terrain edges into a bivariate function. This dual transform is similar to that of Halperin and Sharir[12] but, for our problem, only a constant number of envelope computations are required, since we only consider horizontal viewing directions.

Our dual space can be visualized as follows. If we intersect a terrain with an arbitrary vertical plane π then only the edge whose intersection point is of highest altitude supports a horizontal line in π that satisfies our definition of a horizontal visual line presented in the introduction (Fig. 2 right). All other horizontal tangents in π to the edges that intersect π must contain at least one point below the terrain. The xy-projection of π (Fig. 2 left) is a line with slope m and y-intercept b. Thus, we can compute the set of all distant horizons by first transforming each terrain edge e into a partial bivariate function $\{(m, b, h)|h = f_e(m, b)\}$ in a dual (m, b, h)-space, where $f_e(m, b)$ is the height of the point of intersection of e and the vertical plane π specified by m and b, and then taking the upper envelope of those functions[c]. We now derive the function $f_e(m, b)$ for a terrain edge e with vertices (x_1, y_1, h_1) and (x_2, y_2, h_2).

For a fixed slope m, there is exactly one horizontal line that passes through the point (x_1, y_1, h_1). The horizontal line is h_1 units above the xy-plane and has y-intercept $y_1 - x_1 m$. Thus, the vertex (x_1, y_1, h_1) maps to a line in dual (m, b, h)-space with points $(m, y_1 - x_1 m, h_1)$. Similarly, the point (x_2, y_2, h_2) maps to a line with points $(m, y_2 - x_2 m, h_2)$. Let $b_1(m) = y_1 - x_1 m$ and $b_2(m) = y_2 - x_2 m$ respectively. In primal, a vertical plane π, whose xy-projection has slope m and y-intercept b, cuts e if and only if $b \in [b_1(m), b_2(m)]$. Since an edge is a linear function, the height of the intersection between π and e (and hence the height in dual space) can be determined by linear interpolation of the height of the edge's vertices. Therefore, an edge e in primal is a partial bivariate function in dual. The function $f_e(m, b)$ is

[c]Potentially, problems could arise when $m = \infty$. One way to handle this special case is by computing the horizon at $m = \infty$ and reporting the edges or edge intervals when required. Another way would be to divide the set of horizontal viewing directions into two sets: one for $m \in [-1, 1]$ and another for all viewing directions outside this interval with the x-axis and the y-axis reversed in primal. We would then compute the upper envelope of two sets of functions.

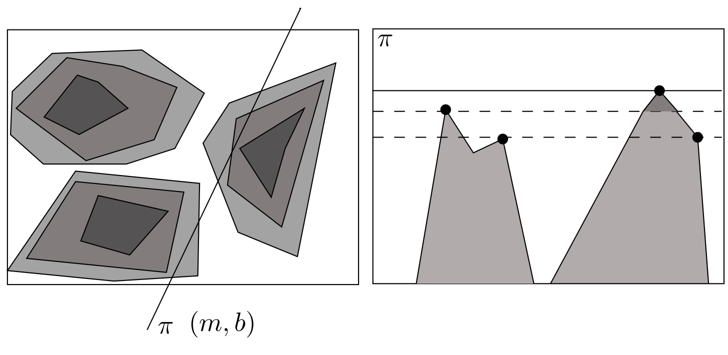

Fig. 2. The diagram shows two views of the terrain. Left: A view of the terrain from above. The concentric polygons of different colours represent the contours of the terrain at different altitudes. Right: The vertical slice π of the terrain.

given by:

$$f_e(m,b) = \begin{cases} (h_2 - h_1)\frac{b - b_1(m)}{b_2(m) - b_1(m)} + h_1 & b_1(m) \neq b_2(m), \\ & b \in [b_1(m), b_2(m)] \\ \max(h_1, h_2) & b_1(m) = b_2(m) = b \\ \text{undefined} & \text{otherwise} \end{cases} \quad (1)$$

The first case of equation (1) occurs when e is cut by the vertical plane π under the circumstances described above. The second case occurs when e is coplanar with π; we take the vertex of e that is of highest altitude. Finally, the last case occurs when the plane π does not intersect e at all. It should be noted that two of these functions can intersect along curves of at most degree two.

We now demonstrate that the surfaces produced by equation (1) satisfy the properties (i - iv) so we can use the algorithms of Halperin and Sharir[12] or Boissonnat[6] or the deterministic algorithm of Agarwal et al.[2] to compute the upper envelope of the functions. The first property is satisfied as any vertical plane π intersects a non-coplanar edge at most once at a unique height h. If the edge is coplanar with π, only the vertex of highest altitude is counted. Moreover, the surfaces are described by algebraic functions of at most degree two as the above derivation demonstrates. The second property is satisfied as there are exactly two linear functions ($b_1(m)$ and $b_2(m)$) that bound the bivariate function of an edge. The third property is satisfied because there are at most two horizontal lines tangent to any triple of edges. This can be seen by considering the points of intersection between an edge and a horizontal plane that is swept from highest to lowest altitude as moving points; in the horizontal plane, the intersection points move with linear trajectories and three such moving points can become collinear at most twice as we show in lemma 1. The fourth property is satisfied by ensuring that the edges of the terrain are in general position.

Lemma 1. *Three points moving with linear trajectories become collinear at most twice.*

Proof. Let $x_i(t)$ and $y_i(t)$ for $i = 1, 2, 3$ be the x and y coordinates of the three points at time t. The three points are collinear when:

$$\begin{vmatrix} x_1(t) & y_1(t) & 1 \\ x_2(t) & y_2(t) & 1 \\ x_3(t) & y_3(t) & 1 \end{vmatrix} = 0$$

As the coordinates of the three points vary linearly with t, the determinant is a quadratic form in t. Therefore, three points moving with linear trajectories become collinear at most twice. □

As these four properties are satisfied, the algorithm of Agarwal et al.[2] can be used to compute the upper envelope of the functions in $O(n^{2+\epsilon})$ time for any $\epsilon > 0$. It can be used to solve problem (1) as stated in the introduction, since a terrain edge e appears on some distant horizon if and only if its associated function forms part of the upper envelope. Problem (2) requires some additional processing as we must determine the intervals of direction for which each edge appears on a distant horizon. These intervals of direction can be determined by projecting the upper envelope down onto the mb-plane from above (see Fig. 3), and then considering the regions in the plane, where the bivariate function $f_e(m, b)$ for an edge e realizes the upper envelope or a distant horizon. We call these regions *patches*. It has been shown that there are at most $O(n2^{O(\sqrt{\log n})})$ of these patches in the plane if the underlying functions satisfy the four properties. Thus, to compute the solution to problem (2), we take each patch, and determine the maximum and minimum m-coordinate it realizes, creating a set of intervals. All the intervals that arise from the same function (i.e. the same terrain edge) are sorted, and overlapping intervals are merged to create the intervals of viewing directions from which this edge appears on the distant horizon.

4. Query Data Structure for a fixed Distant Horizon

We can use the upper envelope computed by the algorithm in section 3 to solve the third, query, version of the problem. In particular, once the set of intervals that solves problem (2) is known, we load the intervals into an interval tree[9]. If s edges appear on the horizon, the query data structure returns the edges on the distant horizon in $O(\log n + s)$ time.

When given a viewing direction m, the s edges on the distant horizon can be determined in $O(\log n + s)$ time as the tree has depth of at most $O(\log n)$ and each edge on the distant horizon is reported exactly once. The precise distant horizon can be determined afterward from the s edges in $O(s \log s)$ time using Hershberger's algorithm[13]. The value of s should generally be small, making this computation relatively inexpensive. However, if the precise distant horizon is desired, the data

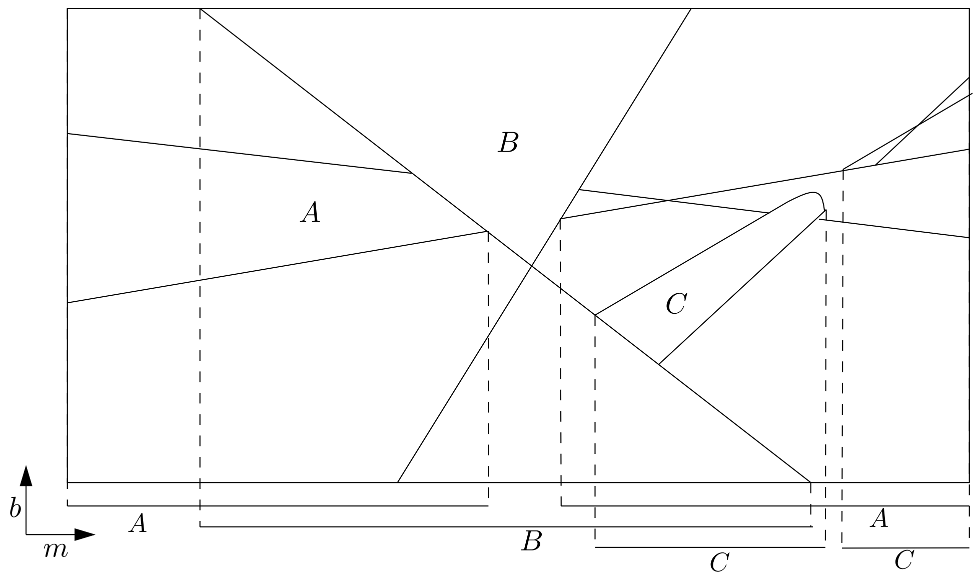

Fig. 3. Patches on the upper envelope transformed into intervals of viewing direction where the edges A, B, and C appear on the distant horizon.

structure can be modified to return it in $O(\log n + s\alpha(s))$ by storing the boundary of each patch with each interval and by not merging overlapping intervals.

5. 3SUM Hardness of Computing all Distant Horizon Edges

In this section, we show that the problem of determining the set of all terrain edges that appear on the distant horizon for at least one horizontal viewing direction (problem version 1) is 3SUM hard. We describe a $O(n \log n)$ time reduction from the *GeomBase* problem, which Gajentaan and Overmars[11] show to be equivalent to the classic 3SUM problem. Our reduction is similar to their reduction of *GeomBase* to *Separator 1*. Our reduction suggests that algorithms to solve the weakest version of the distant horizon problem with sub-quadratic run times are unlikely to exist. The *GeomBase* problem can be stated as follows:

GeomBase: Given a set of n points with integer coordinates on three horizontal lines $y = 0$, $y = 1$, and $y = 2$, determine whether there exists a non-horizontal line containing three points.

Our reduction is as follows. Sort the points on each of the three lines separately by x-coordinate. Let max and min be the largest and smallest x-coordinate of any point on any of the three lines. For each point (x, y) create four points $(x \pm \frac{1}{8}, y, 1)$

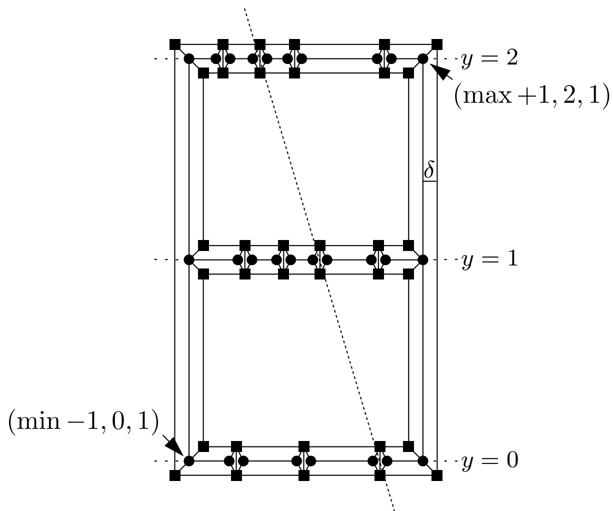

Fig. 4. Shows a terrain created for an instance of *GeomBase*. The solid vertices in the diagram are at height 1. The square vertices are at height 0.

and $(x, y \pm \delta, 0)$ and connect these vertices in a diamond shape (the s in Fig. 4) called a *slit*. A slit is triangulated by connecting its lower vertices. The value of δ can be any positive value less than $\frac{1}{2}$. For any two horizontally adjacent points (x_1, y) and (x_2, y) with $x_1 < x_2$, connect $(x_1 + \frac{1}{8}, y, 1)$ with $(x_2 - \frac{1}{8}, y, 1)$.

We now create what are referred to as the boxes in the diagram. Place six vertices to form *box* at $(x \in \{\min -1, \max +1\}, y \in \{0, 1, 2\}, 1)$ (see Fig. 4). Place three sets of four vertices at height 0 to form *inner box 1*, *inner box 2*, and *outer box*. The distance between an edge of *box* and any directly similar edge on *inner box 1*, *inner box 2*, and *outer box* is equal to δ. Triangulate the remaining faces of the terrain to complete the reduction. The reduction can be done in $O(n \log n)$ time as the time required to sort by x-coordinate dominates. The transformation of the output of the the distant horizon problem to the *GeomBase* problem is to scan the $O(n)$ edges adjacent to a slit and answer "yes" if and only if one of them lies on some distant horizon.

For the intersection of any horizontal plane with the terrain at an altitude in $[0, 1]$, we have a number of horizontal segments in the plane with a horizontal distance less than $\frac{1}{4}$ between any adjacent pair. It has been shown by Gajentaan and Overmars[11] that this horizontal distance is sufficiently small to ensure that a line passes through one pair of segments on each of $y = 0$, $y = 1$, and $y = 2$, if and only if three points were collinear in the original instance of the *GeomBase* problem. Thus, a line passes through three slits if and only if three points in the original instance of the *GeomBase* problem were collinear. This concludes the reduction.

Although the reduction demonstrates the problem of computing all distant horizon edges is 3SUM hard, the reduction requires $\Theta(n)$ local maxima. It is still unknown if the problem is 3SUM hard when restricted to terrains with only a constant number of local maxima. It is relatively straightforward to show that there exists a terrain of k local maxima with $\Omega(kn)$ distant horizon direction intervals. However, in the next section, we provide an example of a terrain with only a single local maximum and $\Omega(n^2)$ distant horizon direction intervals in total.

6. An $\Omega(\mathbf{n}^2)$ Lower Bound for Computing All Distant Horizon Direction Intervals

We now present a terrain T with a single local maximum and $\Theta(n)$ edges with $\Theta(n^2)$ distant horizon direction intervals. It follows immediately from this example that any algorithm to return the set of all distant horizon direction intervals would take at least $\Omega(n^2)$ time. An overhead view of the terrain T is shown in Fig. 5. To demonstrate that this terrain has a quadratic number of direction intervals, we will consider the intersection of the terrain with a horizontal plane that is swept from highest to lowest altitude. The intersection T_h of the horizontal plane $z = h$ with the terrain T gives a simply connected region, whose vertices move with linear trajectories as the altitude of the plane is lowered. The points of intersection between the horizontal plane and any edge $\overline{a_j^+ a_j^-}$ or $\overline{b_i^+ b_i^-}$ are grouped into two chains of moving points parameterized by z. The moving points $a_j(z)$, $j \in \{0, 1 \ldots n\}$, which arise from the intersection of the horizontal plane with a $\overline{a_j^+ a_j^-}$ edge at an altitude of z, are grouped into *chain* A, while the moving points $b_i(z)$, $i = \{0, 1 \ldots n\}$, which arise from the intersection of the horizontal plane with a $\overline{b_i^+ b_i^-}$ edge at an altitude of z, are grouped into *chain* B. We show that each point on chain B supports $\Omega(n)$ distant horizon edge intervals. These intervals of viewing direction begin or end when the edge is placed on or removed from the convex hull of T_h. This example is the first example, to the authors' knowledge, of a set of n points with non-intersecting, linear trajectories whose convex hull changes $\Omega(n^2)$ times. Thus, it could be used in place of an example of Agarwal et al.[1], which shows a set of n points whose convex hull changes $\Omega(n^2)$ times, but whose trajectories intersect.

Let d be the distance indicated in Fig. 5. Let O be the origin of the coordinate system with the z-axis pointing out of the page. The edge $\overline{a_j^+ a_j^-}$ is the jth edge in the chain A:

$$a_j^+ = \left(-d - j, -\frac{(j+1)j(2d+j)}{(\sqrt{2}d - j)(\sqrt{2}d - (j+1))}, h^+\right)$$

$$a_j^- = \left(-d - j, \frac{\sqrt{2}d(2d+j)}{(\sqrt{2}d - (j+1))(\sqrt{2}d - j)}n - \frac{(j+1)j(2d+j)}{(\sqrt{2}d - j)(\sqrt{2}d - (j+1))}, h^-\right)$$

Similarly, the edge $\overline{b_i^+ b_i^-}$ is the ith edge in chain B:

$$b_i^+ = \left(d - \frac{d}{2^i}, -i\delta, h^+\right)$$

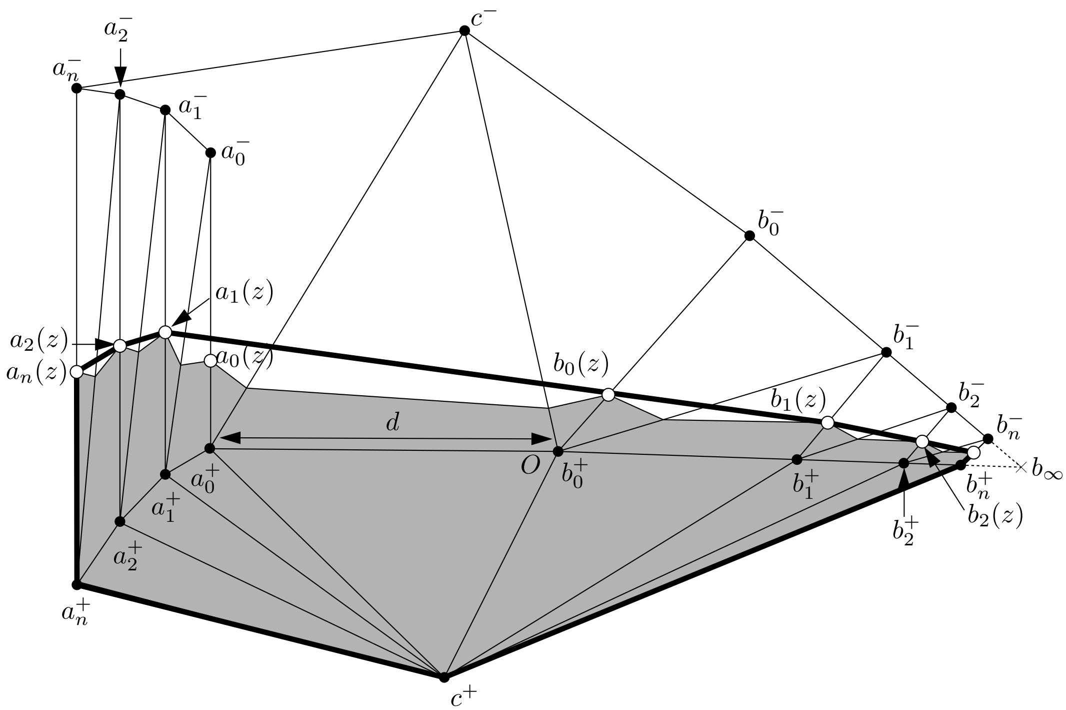

Fig. 5. Top down view of the terrain T with $\Omega(n^2)$ direction intervals. Its intersection T_h with a horizontal plane at an altitude $z \in [h^-, h^+]$ is shown in grey and the convex hull of the intersection is shown as a thick black line. The hollow vertices are the intersection points of the terrain edges with the horizontal plane.

$$b_i^- = \left(d - \frac{d}{2^i} + \frac{n}{2^i\sqrt{2}}, \frac{n}{2^i\sqrt{2}} - i\delta, h^-\right)$$

The vertices b_i^+ and b_i^- are perturbed by $-i\delta$ in the y direction so that they are not collinear but rather form a convex chain (see lemma 3). The terrain has a single local maximum with height h^{++} (the vertex labelled c^+). The vertices labelled a_j^+ and b_i^+ for $i = \{0, 1 \ldots n\}$ and $j = \{0, 1 \ldots n\}$ are at height h^+ and the vertices labelled a_j^- and b_i^- are at a height of h^- with $h^- = h^+ - n$. The vertex labelled c^- is at a height of h^{--}. Completing the triangulation with the edges $\overline{b_i^+ b_{i+1}^-}$ and $\overline{a_i^+ a_{i-1}^-}$ does not introduce points onto the convex hull as the edges lie inside concavities. The heights h^{++}, h^+, h^-, and h^{--} are chosen such that $h^{++} > h^+$ and $h^- > h^{--}$.

The intersection point of each edge in chain A moves in the sweep plane with a linear trajectory parallel to the y-axis. The intersection point on each edge of chain B moves in the sweep plane with a linear trajectory forming an angle of $\frac{\pi}{4}$ with the x-axis. Let $a_j(z)$ be the intersection point of the edge $\overline{a_j^+ a_j^-}$ with the horizontal plane at altitude $z \in [h^-, h^+]$:

$$a_j(z) = \left(-d - j, \frac{\sqrt{2}d(2d+j)}{(\sqrt{2}d-(j+1))(\sqrt{2}d-j)}(h^+ - z) - \frac{(2d+j)j(j+1)}{(\sqrt{2}d-(j+1))(\sqrt{2}d-j)} \right)$$

Let $b_i(z)$ be the intersection point of the edge $\overline{b_i^+ b_i^-}$ with this same plane:

$$b_i(z) = \left(d - \frac{d}{2^i} + \frac{h^+ - z}{2^i\sqrt{2}}, \frac{h^+ - z}{2^i\sqrt{2}} - i\delta \right)$$

Let b_∞ be the point $(d, 0)$. An edge $\overline{b_i^+ b_i^-}$ on chain B supports a horizontal visual line when its intersection point $b_i(z)$ is on the convex hull of a horizontal slice of the terrain (Fig. 5). It fails to support a horizontal visual line when $b_i(z)$ is inside of the convex hull. The appearance or disappearance of $b_i(z)$ on the convex hull corresponds to the start or end of an edge interval. We now analyze how all $b_i(z)$ for $i \in \{0, 1 \ldots n\}$ appear on or disappear from the convex hull for $z \in [h^-, h^+]$ (Fig. 6).

At altitude $z = h^+ - j$ where $j \in \{0, 1 \ldots n\}$, every $b_i(z)$ is on the convex hull and both $a_j(z)$ and $a_{j-1}(z)$ are on the line $\overline{b_0(z)b_\infty}$. For an altitude $z < h^+ - j$, $a_{j-1}(z)$ is inside the convex hull, but $a_j(z)$ is on the convex hull for $z \in (h^+ - (j+1), h^+ - j)$. As z decreases within this interval, $b_0(z)$, $b_1(z) \ldots b_{n-1}(z)$ disappear from the convex hull until all of them lie under the line $\overline{a_j(z)b_n(z)}$ ($b_n(z)$ does not disappear from the convex hull). As z continues to decrease in $(h^+ - (j+1), h^+ - j)$, $b_{n-1}(z)$, $b_{n-2}(z) \ldots b_0(z)$ appear on the convex hull, until $z = h^+ - (j+1)$ when every $b_i(z)$ is again on the convex hull and $a_{j+1}(z)$ and $a_j(z)$ are on the line $\overline{b_0(z)b_\infty}$. As every $a_j(z)$ undergoes this process in a unique interval $(h^+ - (j+1), h^+ - j)$, n edges of chain B have $\Omega(n)$ direction intervals. Thus, T has a total of $\Omega(n^2)$ direction intervals. To prove that this actually occurs, we need to show that $a_j(z)$ is the only intersection point on chain A that is a positive distance above the line $\overline{b_0(z)b_\infty}$ in the interval $z \in (h^+ - (j+1), h^+ - j)$. Then, we need to show that the chain B is a convex chain for all $z \in [h^-, h^+]$ so that a sufficiently small, non-zero δ can be chosen such that all $b_i(z)$ disappear from the convex hull and then reappear on it in the interval $z \in (h^+ - (j+1), h^+ - j)$. Finally, we need to prove that the distant horizon direction intervals on each edge are disjoint, demonstrating that there are truly $\Omega(n^2)$ distant horizon direction intervals.

Lemma 2. *The intersection point $a_j(z)$ is the only intersection point on chain A which is a positive distance above the line $\overline{b_0(z)b_\infty}$ in the interval $z \in (h^+ - (j + 1), h^+ - j)$.*

Proof. We derive an expression for the distance y between $a_j(z)$ and the intersection of $\overline{b_0(z)b_\infty}$ with the line $x = -j - d$ and determine for what values of z it is positive. The distance y is:

$$y = \frac{(2d + j)(\sqrt{2}d(h^+ - z - j(j + 1))}{(\sqrt{2}d - j)(\sqrt{2}d - (j + 1))} - \frac{(2d + j)(h^+ - z)}{\sqrt{2}d - (h^+ - z)}$$

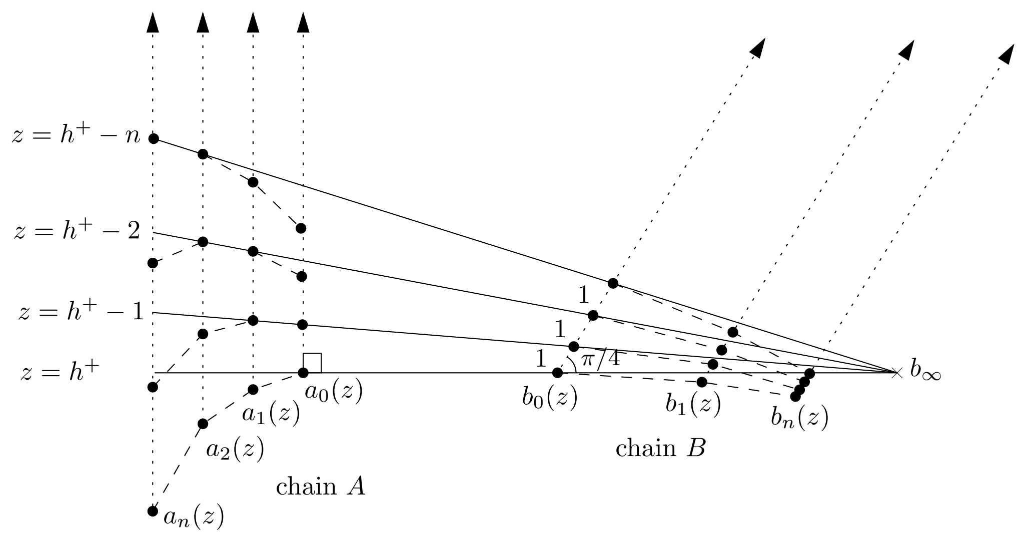

Fig. 6. The trajectories (dotted lines) of the intersection points in the sweep plane showing how chains A and B (dashed lines) evolve through $z \in [h^-, h^+]$.

which simplifies to:

$$y = \frac{\sqrt{2}d(2d+j)}{(\sqrt{2}d - j)(\sqrt{2}d - (j+1))} \frac{-(h^+ - z)^2 + (2j+1)(h^+ - z) - j(j+1)}{\sqrt{2}d - (h^+ - z)}$$

We need to know when $a_j(z)$ is above the line $\overline{b_0(z)b_\infty}$ i.e. when $y > 0$. This occurs if and only if:

$$(h^+ - z - j)(-(h^+ - z) + (j+1)) > 0$$

which holds when $z > h^+ - \sqrt{2}d$, which is true for sufficiently large d, and $z \in (h^+ - (j+1), h^+ - j)$. Therefore, the point $a_j(z)$ is the only point that is a positive distance above the line $\overline{b_0(z)b_\infty}$ in the interval $z \in (h^+ - (j+1), h^+ - j)$. □

Since $a_j(z)$ passes a positive distance above $\overline{b_0(z)b_\infty}$, sufficiently small, non-zero δ can be chosen such that $a_j(z)$ passes above and then falls below all $\overline{b_i(z)b_{i+1}(z)}$ in the interval $(h^+ - (j+1), h^+ - j)$. We now prove the convexity of chain B in all intervals.

Lemma 3. *Chain B is a convex chain for all $z \in [h^-, h^+]$.*

Proof. Chain B will be convex for all $z \in [h^-, h^+]$ if it can be shown that the slopes of $\overline{b_i(z)b_{i+1}(z)}$ decrease with increasing i or:

$$\frac{\frac{h^+-z}{2^{(i-1)}\sqrt{2}} - (i-1)\delta - \frac{h^+-z}{2^i\sqrt{2}} + i\delta}{-\frac{d}{2^{(i-1)}} + \frac{h^+-z}{2^{(i-1)}\sqrt{2}} + \frac{d}{2^i} - \frac{h^+-z}{2^i\sqrt{2}}} - \frac{\frac{h^+-z}{2^i\sqrt{2}} - i\delta - \frac{h^+-z}{2^{i+1}\sqrt{2}} + (i+1)\delta}{-\frac{d}{2^i} + \frac{h^+-z}{2^i\sqrt{2}} + \frac{d}{2^{i+1}} - \frac{h^+-z}{2^{i+1}\sqrt{2}}} > 0$$

which simplifies to:

$$\frac{2^{i+1}\delta}{2d - \sqrt{2}(h^+ - z)} > 0$$

which is positive for all $z > h^+ - \sqrt{2}d$. Therefore the chain B will be convex for all $z \in [h^-, h^+]$. □

Now we show that the distant horizon direction intervals on each edge are disjoint, demonstrating that there are $\Omega(n)$ distant horizon direction intervals on each edge of chain B.

Lemma 4. *There are $\Omega(n)$ distant horizon direction intervals for an edge $\overline{b_i^+ b_i^-}$.*

Proof. Let h_j^- and h_j^+ be the values of z for which $a_j(z)$, $b_i(z)$, and $b_{i+1}(z)$ are collinear, with $h_j^- < h_j^+$. The terrain edge $\overline{b_i^+ b_i^-}$ supports a horizontal tangent only for z in one of the intervals

$$(h^-, h_n^-), (h_n^+, h_{n-1}^-), \ldots, (h_1^+, h_0^-), (h_0^+, h^+)$$

since at altitudes z between h_j^- and h_j^+, $b_i(z)$ is not on the convex hull of T_z.

To establish the lemma, it suffices to show that the slopes (in the xy-plane) of horizontal visual lines tangent to $b_i(z)$ for z in two different intervals are different. At any altitude z, the slope of a horizontal visual line tangent to $b_i(z)$ must be greater than the slope of $\overline{b_i(z)b_{i+1}(z)}$ and less than the slope of $\overline{a_j(z)b_i(z)}$, for any j. By Lemmas 5 and 6, the slopes of $\overline{b_i(z)b_{i+1}(z)}$ and $\overline{a_j(z)b_i(z)}$ increase for increasing z. Thus the slopes of the visual lines tangent to $b_i(z)$ for $z \in (h_{j+1}^+, h_j^-)$ must be greater than the slope of $\overline{b_i(h_{j+1}^+)b_{i+1}(h_{j+1}^+)}$ and less than the slope of $\overline{a_j(h_j^-)b_i(h_j^-)}$. Since $a_j(z)$, $b_i(z)$, and $b_{i+1}(z)$ are collinear at $z = h_j^-$ and again at $z = h_j^+$, by Lemma 5 (or 6), the slopes of the visual lines tangent to $b_i(z)$ for $z \in (h_{j+1}^+, h_j^-)$ are less than the corresponding slopes for $z \in (h_j^+, h_{j-1}^-)$. Therefore, there are $\Omega(n)$ direction intervals for every edge $\overline{b_i^+ b_i^-}$ of the terrain. □

Lemma 5. *The slope of $\overline{b_i(z)b_{i+1}(z)}$ increases for increasing z.*

Proof. The slope of $\overline{b_i(z)b_{i+1}(z)}$ is given by:

$$m(z) = \frac{\frac{h^+ - z}{2^i\sqrt{2}} - i\delta - \frac{h^+ - z}{2^{i+1}\sqrt{2}} + (i+1)\delta}{-\frac{d}{2^i} + \frac{h^+ - z}{2^i\sqrt{2}} + \frac{d}{2^{i+1}} - \frac{h^+ - z}{2^{i+1}\sqrt{2}}}$$

and its derivative simplifies to:

$$m'(z) = \frac{\sqrt{2}(d + 2^{i+1}\delta)}{(h^+ - z - \sqrt{2}d)^2}$$

which is always positive. Therefore, the slope of $\overline{b_i(z)b_{i+1}(z)}$ increases for all increasing z. □

Lemma 6. *For $d > (n+1)/\sqrt{2}$ and $\delta < 3$, the slope of $\overline{a_j(z)b_i(z)}$ increases for increasing z.*

Proof. The slope of $\overline{a_j(z)b_i(z)}$ is

$$m(z) = \frac{\frac{\sqrt{2}d(2d+j)(h^+-z)}{(\sqrt{2}d-(j+1))(\sqrt{2}d-j)} - \frac{(2d+j)(j+1)j}{(\sqrt{2}d-(j+1))(\sqrt{2}d-j)} - \frac{h^+-z}{2^i\sqrt{2}} + i\delta}{-2d - j + \frac{d}{2^i} - \frac{h^+-z}{2^i\sqrt{2}}}$$

and its derivative is

$$m'(z) = \frac{\left(2d+j-\frac{d}{2^i}\right)\left(\frac{\sqrt{2}d(2d+j)}{(\sqrt{2}d-(j+1))(\sqrt{2}d-j)} - \frac{1}{2^i\sqrt{2}}\right) + \frac{1}{2^i\sqrt{2}}\left(\frac{(2d+j)(j+1)j}{(\sqrt{2}d-(j+1))(\sqrt{2}d-j)} - i\delta\right)}{\left(-2d-j+\frac{d}{2^i}-\frac{h^+-z}{2^i\sqrt{2}}\right)^2}.$$

For $d > (n+1)/\sqrt{2}$,

$$\frac{(2d+j)(j+1)j}{(\sqrt{2}d-(j+1))(\sqrt{2}d-j)} > 0 \qquad \text{and} \qquad 2d+j-\frac{d}{2^i} > 1$$

Thus for such a choice of d, $m'(z)$ is positive if

$$\left(2d+j-\frac{d}{2^i}\right)\left(\frac{\sqrt{2}d(2d+j)}{(\sqrt{2}d-(j+1))(\sqrt{2}d-j)} - \frac{1+i\delta}{2^i\sqrt{2}}\right) > 0,$$

which holds, since j ranges from 0 to n, if

$$\frac{1+i\delta}{2^i\sqrt{2}} < \sqrt{2}.$$

Since i ranges from 0 to n, this holds if $\delta < 3$. Therefore, for $d > (n+1)/\sqrt{2}$ and $\delta < 3$, the slope of $\overline{a_j(z)b_i(z)}$ increases for increasing z. □

By lemmas 2, 3, and 4, sufficiently small positive, non-zero δ can be chosen such that all $b_i(z)$ for $i \in \{0, 1 \ldots n-1\}$ disappear from the convex hull and then reappear on it in the interval $z \in (h^+ - (j+1), h^+ - j)$. These appearances or disappearances of points from the convex hull correspond to disjoint distant horizon direction intervals for each edge $\overline{b_i^+ b_i^-}$ for $i = \{0, 1 \ldots n-1\}$. Thus, as n edges on chain B each experience n distant horizon direction intervals, and terrain T has $\Omega(n^2)$ distant horizon direction intervals.

7. Conclusions and Future Work

In this paper, we have described an algorithm to compute all distant horizon edges in $O(n^{2+\epsilon})$ time for any constant $\epsilon > 0$ where n is the number of edges of the terrain. To return the set of direction intervals for every edge of the terrain, we require only an additional $O(\log n)$ factor. We have shown that the problem of computing all distant horizon edges is 3SUM hard, and we have provided an example of a terrain with a constant number of local maxima and $\Omega(n^2)$ distant horizon edge intervals. We have also described a query data structure that can be constructed in $O(n^{2+\epsilon})$ time for any constant $\epsilon > 0$ that can report the edges on a distant horizon from a viewing direction in $O(\log n + s)$ time for a distant horizon of s edges. The value of s is at most n.

Future work consists of determining for a terrain of a constant number of local maxima if the problem of computing the set of edges that appear in the set of all distant horizons is still 3SUM hard. The 3SUM hardness argument requires $O(n)$ local maxima and, thus, it is conceivable that algorithms could be constructed with running times of $O(kn)$ where k is the number of local maxima on the terrain.

8. Acknowledgements

The first author would like to thank Ciarán Llachlan Leavitt for her help with editing this paper into its final form. We gratefully acknowledge the support of DARPA grant F30620-00-2-0560 and the National Sciences and Engineering Research Council of Canada for funding of this project. A huge thanks to Raimund Seidel for his help with this project in the summer of 2003. This paper would not have been possible without it. Finally, we would like to thank the reviewers of our CCCG 2004 paper[4] and the reviewers of The International Journal of Computational Geometry and Applications for providing insightful comments, which helped us improve this paper.

References

1. P. K. Agarwal, L. Guibas, J. Hershberger, and E. Veach. Maintaining the extent of a moving point set. In *The 5th Workshop on Algorithms and Data Structures*, pages 31–34, 1997.
2. P. K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes in three dimensions and its applications. In *Proceedings of the 11th Annual Symposium on Computational Geometry*, pages 182–189, 1995.
3. P. K. Agarwal and M. Sharir. On the number of views of polyhedral terrains. *Discrete and Computational Geometry*, 12:177–182, 1994.
4. D. Archambault, W. Evans, and D. Kirkpatrick. Computing the set of all the distant horizons of a terrain. In *Proceedings of the 16th Canadian Conference on Computational Geometry*, pages 76–79, 2004.
5. M. Atallah. Dynamic computational geometry. In *Proceedings of the 24th IEEE Symposium on the Foundations of Computer Science*, pages 92–99, 1983.
6. J. Boissonnat and K. Dobrindt. On-line construction of the upper envelope of triangles and surface patches in three dimensions. *Computational Geometry: Theory and Applications*, 5(6):303–320, 1995.
7. R. Cole and M. Sharir. Visibility problems for polyhedral terrains. *Journal of Symbolic Computation*, 7(1):11–30, 1989.
8. M. de Berg, D. Halperin, M. Overmars, and M. van Kreveld. Sparse arrangements

and the number of views of polyhedral scenes. *International Journal of Computational Geometry*, 7(3):175–195, 1997.

9. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*, chapter 5, page 212. Springer-Verlag, 2nd edition, 2000.
10. L. de Floriani and P. Magillo. Horizon computation on a hierarchical terrain model. *The Visual Computer: An International Journal of Computer Graphics*, 11:134–149, 1995.
11. A. Gajentaan and M. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Computational Geometry: Theory and Applications*, 5(3):165–185, 1995.
12. M. Halperin and M. Sharir. New bounds for lower envelopes in three dimensions with applications to visibility in terrains. In *Proceedings of the 9th Annual Symposium on Computational Geometry*, pages 11–18, 1993.
13. J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Information Processing Letters*, 33(4):169–174, 1989.
14. A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.
15. S. Petitjean. A computational geometric approach to visual hulls. *International Journal of Computational Geometry and Applications*, 8(4):407–436, 1998.
16. M. Sharir. Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete and Computational Geometry*, 12(1):327–345, 1994.
17. A. Stewart. Fast horizon computation at all points of a terrain with visibility and shading applications. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):82–93, 1998.