New Results on Shortest Paths in Three Dimensions*

Joseph S. B. Mitchell† Micha Sharir‡

August 13, 2004

Abstract

We revisit the problem of computing shortest obstacle-avoiding paths among obstacles in three dimensions. We prove new hardness results, showing, e.g., that computing Euclidean shortest paths among sets of "stacked" axis-aligned rectangles is NP-complete, and that computing L_1-shortest paths among disjoint balls is NP-complete. On the positive side, we present an efficient algorithm for computing an L_1-shortest path between two given points that lies on or above a given polyhedral terrain. We also give polynomial-time algorithms for some versions of stacked polygonal obstacles that are "terrain-like" and analyze the complexity of shortest path maps in the presence of parallel halfplane "walls."

Keywords: Shortest path, NP-hardness, motion planning, terrain

1 Introduction

The problem of computing shortest paths within a geometric domain has been well studied in computational geometry; see the surveys [18, 19]. The two-dimensional problem of computing a shortest path among a set of obstacles is relatively well understood, and there are algorithms [14], based on the continuous Dijkstra paradigm, that compute a shortest path in the Euclidean metric (or any L_p metric, $p \geq 1$) among a set of polygonal obstacles having a total of n vertices, in worst-case optimal running time $O(n \log n)$.

In three dimensions, one is interested in computing a shortest path among a set of polyhedral obstacles, or other obstacles (balls, cylinders, etc). The general problem is known to be hard, with hardness arising from two sources:

Algebraic hardness: Unlike shortest paths among obstacles in the plane, shortest paths in a polyhedral domain need not lie on a discrete graph. Shortest paths in a polyhedral domain

*Work on this paper by J. Mitchell has been supported by grants from the National Science Foundation (CCR-00-98172), NASA Ames Research (NAG2-1620), Metron Aviation, Honda Fundamental Research Lab, and a grant from the U.S.-Israel Binational Science Foundation. Work on this paper by M. Sharir has been supported by National Science Foundation Grant CCR-00-98246, by a grant from the U.S.–Israel Binational Science Foundation, by a grant from the Israeli Academy of Sciences for a Center of Excellence in Geometric Computing at Tel Aviv University, and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University. A preliminary version of this paper has appeared in *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, 2004, pp. 124–133.

†Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794-3600, USA; jsbm@ams.sunysb.edu

‡School of Computer Science,Tel Aviv University, Tel Aviv 69978, Israel, and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA; michas@tau.ac.il

are polygonal, with bend points that are either obstacle vertices, or lie interior to obstacle edges and then they obey a simple unfolding property: The path reaches the edge and leaves it at equal angles. Thus, a locally optimal subpath that bends only at edges can be unfolded at each edge along its edge sequence, resulting in a straight segment. Given an edge sequence, this local optimality property uniquely identifies a shortest path from s to t through that edge sequence. However, Bajaj [1, 2] has shown that comparison of the lengths of two paths arising from distinct edge sequences may require exponentially many bits, since the algebraic numbers that describe the optimal path lengths may have exponential degree. (We note, though, that this is not the main difficulty in a combinatorial approach to the problem. In fact, similar issues arise also in the planar case, and usually one assumes a computational model with infinite precision real arithmetic, which we also adopt for the three-dimensional case.)

Combinatorial hardness: The number of combinatorially distinct shortest paths from s to t may be exponential in the input size. In particular, the *shortest path map* with respect to a source point s in a polyhedral domain having n vertices may have an exponential number ($\Omega(2^n)$) of cells, each corresponding to a distinct edge sequence for shortest paths from s to points in the cell. Canny and Reif [3] used this fact to prove the NP-hardness of the 3-dimensional L_p-shortest path problem, for any $p \geq 1$.

In light of the (double) difficulty of the general problem, research has focused on special cases and on approximation algorithms. Special cases having $n^{O(k)}$-time algorithms include Euclidean shortest paths among k convex polyhedral obstacles [22], or among vertical "buildings" (prisms) having k different heights [12].

While computing L_1-shortest paths among general polyhedral obstacles in $\mathbb{R}^3$ is NP-hard [3], if the obstacles are orthohedral, having faces orthogonal to the coordinate axes, a search of the grid graph suffices for computing optimal paths; see, e.g., [7, 9, 10] for efficient algorithms that exploit this special structure.

The special case of shortest paths on a polyhedral surface (i.e., the obstacles are the complement of the surface) can be especially efficiently solved, since the surface constraint effectively makes the problem two-dimensional; the continuous Dijkstra paradigm leads to an $O(n^2)$-time algorithm [4, 20]. (See also [15] for a recently announced improvement.)

Related to some of our work is the recent result of Dror et al. [11] on the *touring polygons problem* (TPP), which can be viewed as a special case of the 3-dimensional shortest path problem in which the obstacles are a "stack" of planes, each having a simple polygonal hole. That is, we have an *ordered* sequence $h_1, \ldots, h_m$ of planes that occupy the same physical location in 3-space (or, more intuitively, are infinitesimally separated from each other), and each of them contains some flat obstacles. The path has to start at $s \in h_1$, end at $t \in h_n$, and be a concatenation of subpaths $\pi_1, \ldots, \pi_m$, such that each π_i is an obstacle-free path in h_i, for $i = 1, \ldots, m$, and the terminal point of π_i is the initial point of π_{i+1}, for each $i = 1, \ldots, m-1$. Dror et al. show that the TPP can be solved exactly in polynomial time if the polygonal holes are convex, and that the problem is NP-hard if the holes are nonconvex (and overlapping).

Approximation algorithms are based on discretizing the space by selectively placing sample points (e.g., along the edges of polyhedral obstacles) and searching the visibility graph of the discrete set of points. Analysis of how well path lengths are preserved in the discretization shows that paths whose length is within $(1+\varepsilon)$ times optimal can be computed in time polynomial in the input complexity and $(1/\varepsilon)$ [8, 21]. Choi, Sellen, and Yap [5, 6] have addressed some inconsistencies in earlier work, drawn attention to the distinction between bit complexity and algebraic complexity, and introduced the notion of "precision-sensitivity." Har-Peled [13] has given discretization methods for computing *approximate shortest path maps* in polyhedral domains, computing, for fixed source s and $\varepsilon \in (0, 1)$, a subdivision of size $O(n^2/\varepsilon^{4+\delta})$ (for any $\delta > 0$), in time roughly $O(n^4/\varepsilon^6)$, so that, for any point $t \in \mathbb{R}^3$, a $(1 + \varepsilon)$-approximation of the length of a shortest s-t path can be reported in time $O(\log(n/\varepsilon))$.

Our Results

In this paper we reconsider the complexity of the shortest path problem among obstacles in $\mathbb{R}^3$ in an attempt to understand better the distinction between those instances that are combinatorially hard and those that are not.

We answer some of the open questions that have been circulating in the community since the Canny-Reif hardness result was discovered in 1986: Is it NP-hard to find Euclidean shortest paths among a set of disjoint axis-aligned boxes, or even a set of "stacked" axis-aligned rectangles? What if the obstacles are disjoint balls or other "fat" sets? What if there is only a single-obstacle polyhedral terrain (and one is allowed to fly over it)?

We do not address issues of algebraic complexity; where needed, we assume that we have an "oracle" that exactly compares path lengths. (Again, this is not much different from the standard assumptions made in the planar case, except that this algebraic decision step is somewhat more complicated in $\mathbb{R}^3$.)

Our results include the following ones:

1. We present an $O(n^3 \log n)$-time algorithm for computing an L_1-shortest path between two given points, that lies on or above a given polyhedral terrain with n faces.

2. We analyze the complexity of shortest path maps in the presence of parallel halfplane "walls." MICHA SAYS: **There are problems with the analysis!**

3. We show that the Euclidean shortest path problem is NP-complete for the case of obstacles that are disjoint axis-aligned boxes, even if the obstacles are "stacked" axis-aligned quadrants each of which is unbounded to the northeast or to the southwest.

4. Complementing our hardness proof, we give polynomial-time algorithms for the case of stacked axis-aligned rectangles that are "terrain-like", each containing a ray in the $(-y)$-direction, and for other favorable classes of axis-aligned rectangular shapes.

5. We prove that it is NP-hard to decide if the length of an L_1-shortest path from s to t is at most l, for the following cases: (i) The obstacles are disjoint balls in $\mathbb{R}^3$, or a "stack of pancakes" (flat circular disks). (ii) The obstacles are a stacked set of squares each at angle $\pi/4$ with respect to the coordinate axes (in this case the problem is NP-complete).

2 L_1-Shortest Paths Above a Terrain

Let T be a polyhedral terrain with n faces, and let s and t be two points that lie on or above the terrain. We wish to compute the L_1-shortest path $\pi_1(s, t)$ from s to t which stays fully on or above

T. We note that $\pi_1(s,t)$ need not be unique. In fact, even in the absence of T, any path from s to t that is monotone in all three coordinate directions is an L_1-shortest path from s to t.

Lemma 1 *Let h be the maximal z-coordinate of a point on some L_1-shortest path $\pi_1(s,t)$ from s to t. Let $s(h)$ (resp., $t(h)$) be the point at height h that lies directly above s (resp., t). Let $\pi_1^{(h)}(s,t)$ denote the path obtained by moving from s directly upwards to $s(h)$, then proceeding from $s(h)$ to $t(h)$ along a shortest L_1-path $\pi_0^{(h)}(s,t)$ that lies fully in the plane $z=h$ and stays on or above T, and finally moving from $t(h)$ directly downwards to t. Then $\pi_1(s,t)$ and $\pi_1^{(h)}(s,t)$ have the same L_1-length.*

Proof: By definition, $\|\pi_1(s,t)\|_1 \le \|\pi_1^{(h)}(s,t)\|_1$. To prove the opposite inequality, let π' denote the path obtained by projecting $\pi_1(s,t)$ vertically upwards onto the plane $z=h$; that is,

$$\pi' = \{(x,y,h) \mid (x,y,z) \in \pi_1(s,t)\}.$$

Note that $\|\pi_1(s,t)\|_1$ is equal to $\|\pi'\|_1$ plus the total z-variation along $\pi_1(s,t)$. Since $\pi_1(s,t)$ starts at s, reaches a point at height h and then gets to t, the z-variation along this path is at least $(h-s_z)+(h-t_z)$. In other words, we have

$$\|\pi_1(s,t)\|_1 \ge \|\pi'\|_1 + (h-s_z)+(h-t_z) \ge \|\pi_0^{(h)}(s,t)\|_1 + (h-s_z)+(h-t_z) = \|\pi_1^{(h)}(s,t)\|_1.$$

This follows from the definition of $\pi_1^{(h)}(s,t)$ and from the fact that π' is a path from $s(h)$ to $t(h)$ that lies in the plane $z=h$ and stays on or above T (which follows since T is a terrain), whereas $\pi_0^{(h)}(s,t)$ is an L_1-shortest such path. This completes the proof of the lemma. □

We vary h and slice T with the plane $\varpi(h):\ z=h$, to obtain a polygonal partition of $\varpi(h)$ into a free region and an obstacle region, consisting of those points in $\varpi(h)$ that lie on or above T and below T, respectively. Let $T(h)$ denote the free portion of $\varpi(h)$, and let, as above, $s(h), t(h)$ denote the points at height h that lie vertically above s and t. Note that $T(h)$ expands as h increases. We also denote by $e(h)$, for each edge e of T, the point that lies on e at height h, if such a point exists. (We assume that T contains no horizontal edges.) Our goal is to compute the L_1-shortest path in $T(h)$ that connects $s(h)$ to $t(h)$, and analyze how it varies as h increases, say, from $h=\max\{s_z,t_z\}$ to $h=+\infty$. Refer to Figure 1. MICHA SAYS: **Please fix the figure: (i) In the bottom-left part: What is the purpose of the empty dot below t? (ii) On the right, the dashed path is not the shortest: Merge the middle obstacle and the right one into a common obstacle; otherwise there is a shorter path that goes through the gap between them.**

Put $L(h) = \|\pi_1^{(h)}(s,t)\|_1$, for $h \ge \max\{s_z,t_z\}$. Note that $L(h) = (2h-s_z-t_z)+\|\pi_0^{(h)}(s,t)\|_1$, where, as above, $\pi_0^{(h)}(s,t)$ is the horizontal portion of $\pi_1^{(h)}(s,t)$ (between $s(h)$ and $t(h)$). Put $L_0(h) = \|\pi_0^{(h)}(s,t)\|_1$.

Lemma 2 *$L_0(h)$ is a (weakly) monotone decreasing function of h.*

Proof: MICHA SAYS: **"weakly" only if T contains vertical edges?!** If $h_1 > h_2$ then $T(h_1) \supseteq T(h_2)$ and hence any path in $T(h_2)$ between $s(h_2)$ to $t(h_2)$ can be lifted to a path in $T(h_1)$ between $s(h_1)$ to $t(h_1)$, which readily implies that $L_0(h)$ is monotone decreasing. □

Let $v_1, v_2, \ldots, v_n$ be the vertices of T sorted in increasing z-order. Let I_i denote the closed interval $[(v_i)_z, (v_{i+1})_z]$, for each $i<n$, and let I_0 (resp., I_n) denote the halfline $(-\infty, (v_1)_z]$ (resp.,

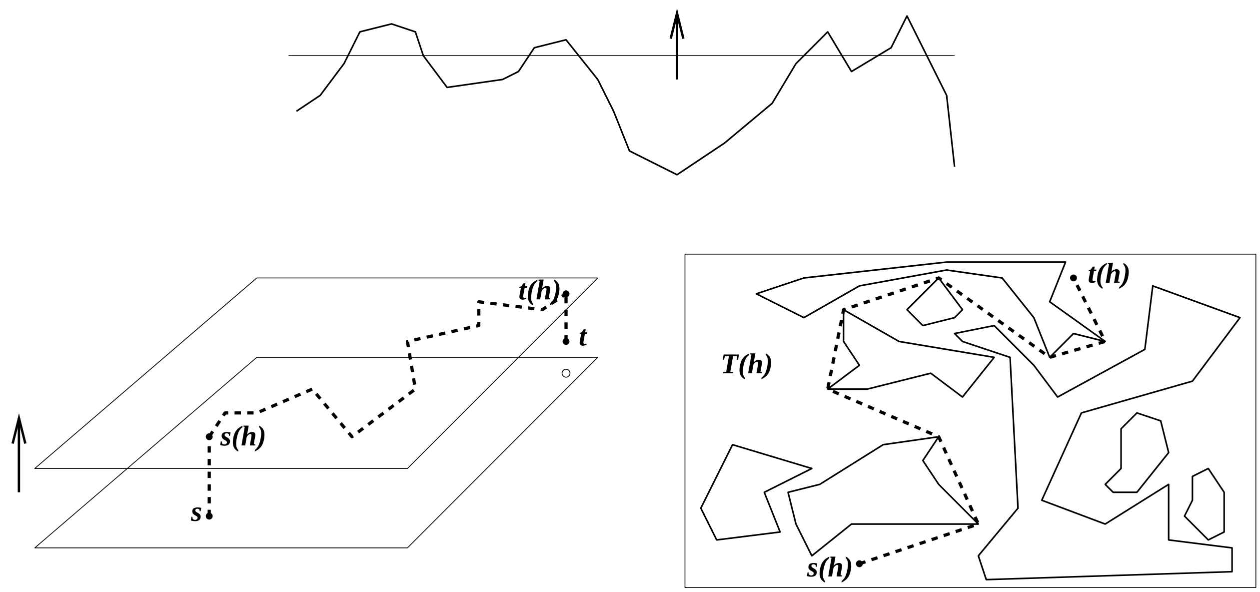

Figure 1: Sweeping a terrain upwards: As the height h varies, the slice of the terrain T yields a free space region $T(h)$, and the task is to compute a shortest path in $T(h)$ from $s(h)$ to $t(h)$.

$[(v_n)_z, +\infty))$. Partition each I_i further, at critical heights h at which there exist two edges e, e' of T such that $e(h)$ and $e'(h)$ have the same x- or y-coordinate. (More precisely, we are only interested in events of this form where the two points $e(h), e'(h)$ are *visible* from each other in $T(h)$.) let $\mathcal{I}$ denote the resulting collection of atomic intervals, over all the I_i's. Note that $|\mathcal{I}| = O(n^2)$, and that this bound is tight in the worst case.

Lemma 3 *For each interval $I \in \mathcal{I}$, the function $L_0(h)$ is a (weakly) monotone decreasing, concave, piecewise linear function of h over I. The function $L(h)$ is a concave piecewise linear function over I.*

Proof: For each h, we may assume that $\pi_0^{(h)}(s,t)$ is a polygonal path that bends only at some reflex vertices of $T(h)$ (convex obstacle vertices). For simplicity of the analysis, we will consider all such bends of $\pi_0^{(h)}(s,t)$, although the only bends that may affect the L_1-length of $\pi_0^{(h)}(s,t)$ are where it changes either its x-direction (from going left to going right or vice versa) or its y-direction (from going up to going down in the y-direction or vice versa), or both (see, e.g., the bends in Figure 5(i)). Other bends are inessential, because the path continues to be there both x- and y-monotone, and its L_1-length is not affected by the bend (see, e.g., the bends in Figure 5(iii)).

Let $G(h)$ denote the graph whose vertices are the points s, t and the edges of T that have points at $z = h$, and where two vertices e, e' of $G(h)$ (say, edges of T) are connected by an edge if the following holds: MICHA SAYS: **I made condition (ii) swallow condition (i).** Let $B(e, e', h)$ denote the axis-parallel rectangle that has $e(h)$ and $e'(h)$ as two opposite vertices. Then the portion of $B(e, e', h)$ that is visible from $e(h)$ contains $e'(h)$, but does not contain any other vertex of $G(h)$. See Figure 2. Edges of $G(h)$ incident to s or to t are defined analogously.

As is easily seen, $G(h)$ does not change combinatorially as h varies in I. Geometrically, $G(h)$ is embedded in the plane $\varpi(h)$ by mapping each vertex e (say, an edge of T) to the actual point $e(h)$ at height h on e (s is mapped to $s(h)$ and t to $t(h)$), and by mapping each edge (e, e') to the straight

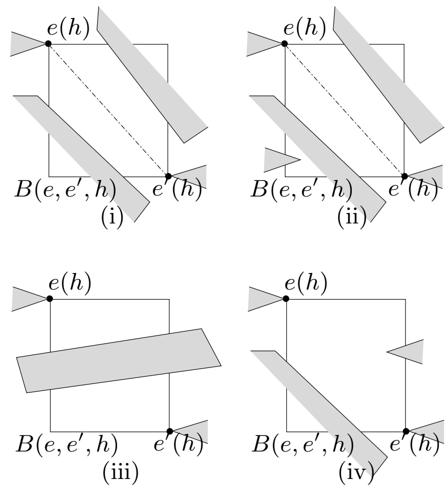

Figure 2: Edges (cases (i) and (ii)) and non-edges (cases (iii) and (iv)) of $G(h)$.

segment in $\varpi(h)$ that connects $e(h)$ and $e'(h)$ (including the cases where e and/or e' are s or t). Clearly, this embedding of $G(h)$ varies *continuously* as h varies in I. The path $\pi_0^{(h)}(s,t)$, under the structural assumption at the beginning of the proof, can be regarded as the embedding in $\varpi(h)$ of a path in $G(h)$ connecting s and t. Let $P(h)$ denote the finite set of all simple paths in $G(h)$ that connect s to t. Then clearly $L_0(h) = \min_{\pi \in P(h)} \|\pi(h)\|_1$, where $\pi(h)$ is the embedding of the path π in the plane $\varpi(h)$, as just described. Since the x- and y-coordinates of each vertex of $T(h)$ are linear functions of h over I, and since any two mutually visible vertices of $T(h)$ have a fixed x-order and a fixed y-order as h varies in I, it follows that $\|\pi(h)\|_1$ is a linear function of $h \in I$. Hence $L_0(h)$ is the lower envelope of a finite number of linear functions, and is thus concave and piecewise linear over I. By Lemma 2, it is also monotone decreasing. Since $L(h)$ is equal to $L_0(h)$ plus a linear function of h, it is also concave and piecewise-linear (but not necessarily monotone decreasing) over I. □

Corollary 1 *$L(h)$ attains its global minimum at an endpoint of some interval in $\mathcal{I}$.*

Proof: First, $L(h)$ has a global minimum: Let h_0 denote the infimum of all h for which $s(h)$ and $t(h)$ are visible. It is easily checked that (i) $L(h)$ is a strictly increasing linear function of h (of slope 2), for $h \geq h_0$, and (ii) $L(h)$ attains its minimum over the interval $[\max\{s_z, t_z\}, h_0]$. This minimum cannot be attained at an interior point of any $I \in \mathcal{I}$, because $L(h)$ is concave there. □

Corollary 1 provides the basis for a simple algorithm for constructing the shortest path, to be described momentarily. Before doing so, we first analyze how the functions $L_0(h)$, $L(h)$ change at endpoints of intervals of $\mathcal{I}$. This analysis could be useful in an attempt to improve the algorithm.

Let h^* be a height at which two mutually visible vertices $e(h^*), e'(h^*)$ of $T(h^*)$ have the same x- or y-coordinate, say $e(h^*)_x = e'(h^*)_x$. Let h_1 be slightly smaller, and h_2 be slightly larger, than h^*. Then every edge of $G(h_1)$ that is incident to neither e nor e' is also an edge of $G(h_2)$, and vice versa. Indeed, $G(h_1)$ (resp., $G(h_2)$) is fixed for all $h_1 < h^*$ (resp., $h_2 > h^*$) and sufficiently close to it. Hence changes in the graph can occur only at h^*. Let g, g' be two edges of T, neither of which is equal to e or e', such that (g, g') is an edge of $G(h_1)$. Assuming general position, it is clear that both conditions in the definition of an edge of $G(h)$ continue to hold for (g, g') as we increase h from h_1 to h_2 through h^*. Hence (g, g') is also an edge of $G(h_2)$. The reverse direction, from $G(h_2)$ to $G(h_1)$, is argued in exactly the same manner. Note also that the L_1-length of $(g(h), g'(h))$ is a linear function of h in the entire interval $[h_1, h_2]$.

Consider next the edge (e, e'). Note that if (e, e') is an edge of $G(h_1)$ then it is also an edge of $G(h_2)$, and vice versa. However, the L_1-length of this edge is a piecewise-linear *convex* function of h, with a bend at h^*. Specifically, assuming that $e(h)$ lies above $e'(h)$ in the y-direction, we have

$$\|e(h)e'(h)\|_1 = e(h)_y - e'(h)_y + |e(h)_x - e'(h)_x|.$$

Consider finally edges of the form (g, e), where $g \neq e'$. Suppose that such an edge belongs to $G(h_1)$. If $e'(h)$ enters the box $B(g, e, h)$ at h^* and violates condition (ii) for this box then (g, e) disappears from $G(h_2)$. In this case, though, it is easily checked that both edges (g, e') and (e, e') remain edges of $G(h_2)$. An important observation is that even though the edge (g, e) is no longer an edge of $G(h_2)$, the L_1-shortest distance from $g(h)$ to $e(h)$ in $T(h)$ remains equal to the L_1-length of the straight segment $g(h)e(h)$. Symmetric arguments apply when moving from $G(h_2)$ to $G(h_1)$ and for edges incident to e'.

The preceding discussion implies the following property: Let π be a path in $P(h_1)$ or in $P(h_2)$, which does not contain the edge (e, e'). Then the L_1-length $a(h)$ of $\pi(h)$ is a linear function over the domain where π exists. Moreover, $L_0(h) \leq a(h)$ over the entire interval $[h_1, h_2]$. On the other hand, if π contains (e, e') then $\|\pi(h)\|_1$ is a piecewise-linear *convex* function over $[h_1, h_2]$, with a bend at h^*.

In other words, if all the paths $\pi(h)$ that attain the minimum $L_0(h)$ at h^* do not use the edge $(e(h), e'(h))$ then $L_0(h)$ is concave and piecewise linear at h^* (it is linear there if only a single such path attains the minimum). If, on the other hand, all paths $\pi(h)$ that attain $L_0(h)$ at h^* use $(e(h), e'(h))$ then $L_0(h)$ is convex and piecewise-linear at h^*. If the set of these paths contains paths of both kinds, the actual behavior of $L_0(h)$ at h^* depends on the relative slopes of the path-length functions of these paths.

The case in which h^* is the height of a vertex v of T is more involved, since $L_0(h)$ can become discontinuous at h^*, as is illustrated in Figure 3. Informally, the topology of $T(h)$ may change as h increases through h^*, in the sense that a new passage in the vicinity of v may open up, allowing much shorter paths to connect $s(h)$ and $t(h)$. We note, though, that such a discontinuity can only happen at vertices that are *saddle points* of the height function. MICHA SAYS: **Agreed?!** Figure 4 depicts the possible behavior of $L_0(h)$ and $L(h)$. MICHA SAYS: **I need to add the figure. Will do it in the next iteration.**

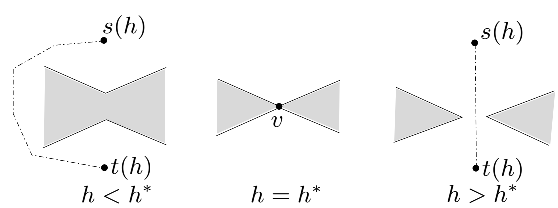

Figure 3: A discontinuity in the length of $\pi_0^{(h)}(s, t)$ as we sweep upwards through a vertex v of T.

Computing the L_1-shortest path. Recall that our goal is to compute the global minimum of $L(h)$, from which $\pi_1(s, t)$ is easily obtained. By Corollary 1, the minimum must be attained at

Figure 4: The possible behavior of $L_0(h)$ and $L(h)$.

an endpoint of some interval in $\mathcal{I}$. This suggests the following simple algorithm. Let H denote the set of endpoints of intervals in $\mathcal{I}$. We compute $\pi_1^{(h)}(s,t)$ for each $h \in H$, and select the path with the smallest length $L(h)$. Each of these computations is the construction of a planar L_1-shortest path in a polygonal environment, which can be accomplished in $O(n \log n)$ time and linear storage [16, 17]. Altogether, the algorithm runs in $O(n^3 \log n)$ time.

Theorem 1 *Let T be a polyhedral terrain with n faces, and let s,t be two points on or above T. The L_1-shortest path between s and t that stays on or above T can be computed in $O(n^3 \log n)$ time (and linear storage).*

2.1 Discussion and Extensions

(1) The same machinery yields a slightly stronger result:

Theorem 2 *Let T be a polyhedral terrain with n faces, and let s be a fixed point on or above T. One can preprocess T and s into a data structure of size $O(n^3)$, in time $O(n^3 \log n)$, so that, given a query point t that lies on or above T, the L_1-shortest path between s and t can be computed in $O(n^2 \log n)$ time.*

Proof: Let H_0 denote the set of critical heights, which are either (i) heights of (saddle) vertices of T; or (ii) heights h where there exist two edges, e and e', of T whose corresponding points, $e(h)$ and $e'(h)$, have the same x-coordinate or the same y-coordinate and are visible in $T(h)$; or (iii) heights h at which there exists an edge e of T whose corresponding point $e(h)$ has the same x-coordinate or the same y-coordinate as the point $s(h)$, and these two points are visible in $T(h)$.

Each $h \in H_0$ is an endpoint of some interval in $\mathcal{I}$, but there exist additional endpoints that depend on t. Specifically, these are heights h where there exists an edge e of T whose corresponding point $e(h)$ have the same x-coordinate or the same y-coordinate as the point $t(h)$, and these two points are visible in $T(h)$. Let $H_1(t)$ denote the set of these additional heights. Fortunately, there are only $O(n)$ such heights. (In contrast, the size of H_0 can be $\Theta(n^2)$.)

The preprocessing algorithm proceeds as follows. For each $h \in H_0$, we compute the *shortest path map* $M(s,h)$ that represents all L_1-shortest paths in $T(h)$ from $s(h)$ to the other points of $T(h)$. As shown in [16, 17], this can be done in $O(n \log n)$ time and $O(n)$ storage. We further process each $M(s,h)$ for efficient (logarithmic-time) point location queries. This too can be done in $O(n \log n)$ time and $O(n)$ storage. In total, preprocessing takes $O(n^3 \log n)$ time and produces a data structure of size $O(n^3)$.

A query with a point t is processed as follows. We first compute the set $H_1(t)$ of additional critical heights. To simplify matters, we simply compute, in $O(n)$ time, the set of all critical heights where $t(h)$ and some $e(h)$ have the same x- or y-coordinate. (This is a superset of $H_1(t)$, because we ignore the requirement of visibility between these two points.) We compute $\pi_1^{(h)}(s,t)$ for each h in this superset, and compute the minimum λ_1 of their L_1-lengths. In addition, for each $h \in H_0$, we locate $t(h)$ in $M(s,h)$, in $O(\log n)$ time. This determines the length of $\pi_1^{(h)}(s,t)$, and we compute the minimum λ_2 of these lengths. The path that attains $\min\{\lambda_1, \lambda_2\}$ is the L_1-shortest path from s to t. The total query time is clearly $O(n^2 \log n)$. $\square$

(2) Can one analyze the changes in the shortest-path map $M(s, h)$ as h varies continuously? The critical heights are those h at which a vertex of $T(h)$ has two homotopically different shortest paths from $s(h)$. How many times can this happen?

(3) One can also consider the all-shortest-paths extension of the problem: Given a set Q of k points on or above T, find all $\binom{k}{2}$ L_1-shortest paths between pairs of points in Q. Using the algorithm provided by Theorem 2, this can be solved in time $O((kn^3 + k^2n^2) \log n)$.

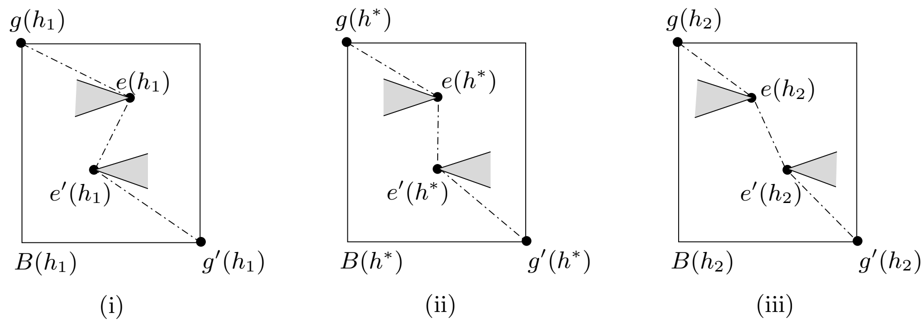

Figure 5: A criticality in the structure of $\pi_0^{(h)}(s, t)$.

3 L_2-Shortest Paths Over Walls

Let $e_1, \ldots, e_n$ be n lines in 3-space, all orthogonal to the y-axis, so that the equation of each e_i is of the form $y = a_i$, $z = b_i x + c_i$, and so that $a_1 < a_2 < \cdots < a_n$. Each e_i defines a *wall* W_i, which is the vertical halfplane lying below e_i. Let s and t be a source and target points, so that $s_y < a_1$ and $t_y > a_n$. The problem is to find the L_2-shortest path from s to t that does not meet the relative interior of any wall W_i. For any point $\zeta \in \mathbb{R}^3$, denote by $\pi(\zeta)$ the shortest path from s to ζ, and by $L(\zeta)$ the length of that path. (We remark that the walls W_i are *not* "stacked", in the sense discussed in the introduction, because they are non-infinitesimally separated in the y-direction.)

We assume that the lines e_i are in general position, except for the "built-in" degeneracy that they lie in planes parallel to the xz-plane. in the strong sense that the coefficients a_i, b_i, c_i and the coordinates of s and t are *algebraically independent*: no multivariate polynomial with integer corefficients vanishes when we substitute for the variables distinct values taken from this coefficient set. MICHA SAYS: **And even that does not solve our problems!**

In this section we provide analysis of the structure of the paths $\pi(\zeta)$ and of the resulting shortest path map, culminating in Theorem 3, which shows that the shortest path map has $O(n^2)$ complexity. Unfortunately, we do not yet have a polynomial-time algorithm that computes the map.

3.1 Properties of $\pi(\cdot)$ and $L(\cdot)$

The first three properties are easy, and are given without proofs.

(1) For any ζ, $\pi(\zeta)$ is a polygonal path that bends only at points that lie on some of the edges e_i.

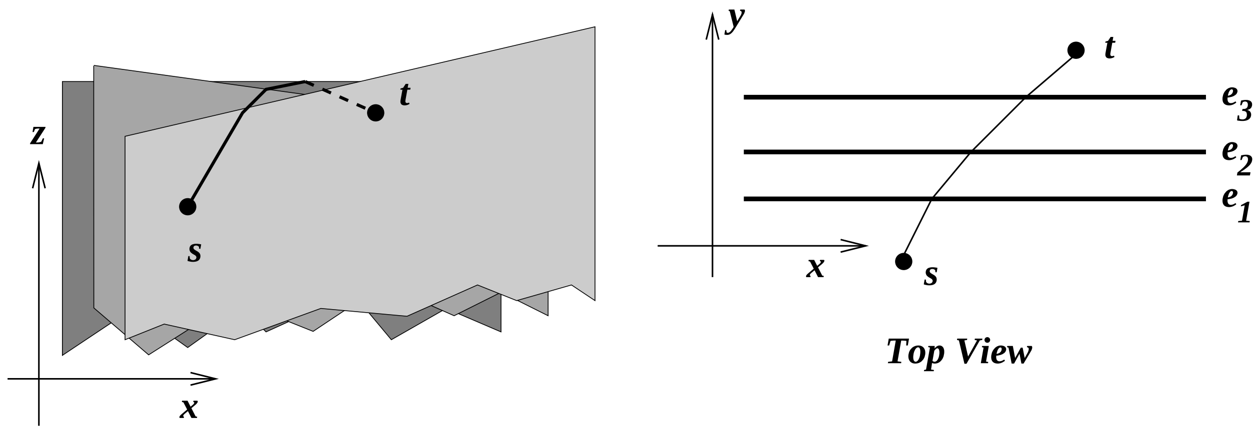

Figure 6: The shortest path problem over a set of walls, each orthogonal to the y-axis, defined by lines $e_1, \ldots, e_n$.

(2) For any ζ, $\pi(\zeta)$ is y-monotone.
(3) $\pi(\zeta)$ is the concatenation of two subpaths $\pi_1 \| \pi_2$ (either of which may be empty), so that π_1 is ascending in the z-direction, and π_2 is descending.
(4) For any ζ, the path $\pi(\zeta)$ is unique.
Proof: For simplicity, assume that $\zeta_y > a_n$, so all walls W_i are "in-between" s and ζ. Consider any y-monotone polygonal path that connects s and ζ, does not intersect the relative interior of any W_i, and bends only at points that lie in the planes $y = a_i$; clearly, $\pi(\zeta)$ is such a path. Let (x_i, a_i, z_i) be the point where our path crosses the plane $y = a_i$. Its length is thus

$$F(x_1, z_1, \ldots, x_n, z_n) = \sum_{i=0}^{n} \sqrt{(x_{i+1} - x_i)^2 + (a_{i+1} - a_i)^2 + (z_{i+1} - z_i)^2},$$

where (x_0, a_0, z_0) are the coordinates of s and $(x_{n+1},\, a_{n+1},\, z_{n+1})$ are the coordinates of ζ. Clearly, F is a convex function, being the sum of convex functions. Moreover, since we have assumed general position, it is easily seen that F is strictly convex. By definition, $L(\zeta)$ is the minimum value of F over the convex polyhedral domain P, defined by $z_i \geq b_i x_i + c_i$, for each $i = 1, \ldots, n$. The minimum of a strictly convex function over a convex domain is unique. $\square$

This proof has several additional implications:
(5) Suppose that ζ varies in some convex domain K. Then $L(\zeta)$ is a convex function of ζ over K.
Proof: Consider the function F defined in the preceding proof, and add to its domain the coordinates of ζ as three additional variables, so it is a function of the $2n + 3$ variables $x_1, z_1, \ldots, x_n, z_n$, ζ_x, ζ_y, ζ_z. By definition,

$$L(\zeta) = \min_{(x_1, z_1, \ldots, x_n, z_n) \in P} F(x_1, z_1, \ldots, x_n, z_n, \zeta_x, \zeta_y, \zeta_z).$$

In other words, the graph of $L(\zeta)$ is the lower boundary of the projection of the graph of F onto the subspace $K \times \mathbb{R}$. Since the projection is a convex region, its lower boundary is the graph of a convex function over the convex domain K. $\square$

(6) Suppose that ζ varies along some line ℓ. Then

(i) $L(\zeta)$ is a convex function of ζ over ℓ.

(ii) As we slide ζ along ℓ, $\pi(\zeta)$ varies continuously (in the Hausdorff metric).

(iii) The combinatorial structure of $\pi(\zeta)$ changes only when it passes through three collinear and mutually visible points that lie on three edges e, e', e'', in which case $\pi(\zeta)$ connects these three points along the line segment that they span.

Proof: Property (i) is a special case of (5). Property (ii) follows by observing that its violation would imply the existence of a point on ℓ with two distinct shortest paths to it, which contradicts the uniqueness of shortest paths (Property (4)). Since $\pi(\zeta)$ varies continuously, the only combinatorial changes that it may encounter occur when it gains a new bend or loses a bend. At any such criticality, the condition in property (iii) must occur. $\square$

3.2 Combinatorial Complexity of the Shortest-Path Map

The structure of $\pi(\zeta)$ in the vicinity of a generic point.

Lemma 4 *Assume that ζ varies along the last line $e = e_n$. Let $\zeta_0 \in e$ be a point for which no collinear criticality of type 6(iii) occurs along $\pi(\zeta_0)$. We vary ζ along e from a point slightly to the left of ζ_0 to a point slightly to its right. Then, for each $i < n$ for which $\pi(\zeta_0)$ bends at e_i, the contact point $\pi(\zeta) \cap e_i$ moves monotonically, either to the left or to the right; these directions depend on ζ_0, and different directions may arise for different lines e_i.*

Proof: For simplicity of presentation, assume that $\pi(\zeta_0)$ is in contact with all lines $e_1, \ldots, e_n$. Put $\zeta_0^{(i)} = \pi(\zeta_0) \cap e_i$, so $\zeta_0 = \zeta_0^{(n)}$. By the continuity of $\pi(\zeta)$, if $\zeta \in e$ is sufficiently close to ζ_0 then $\pi(\zeta)$ continues to touch (and bend at) each line e_i.

Let u_i denote the unit vector along e_i, oriented to the right. Let $\xi_i \in \mathbb{R}$ be such that $\pi(\zeta) \cap e_i = \zeta_0^{(i)} + \xi_i u_i$. We need to show that $\mathrm{sign}(\xi_i)/\mathrm{sign}(\xi_n)$ is fixed when ζ is sufficiently close to ζ_0. See Figure 7.

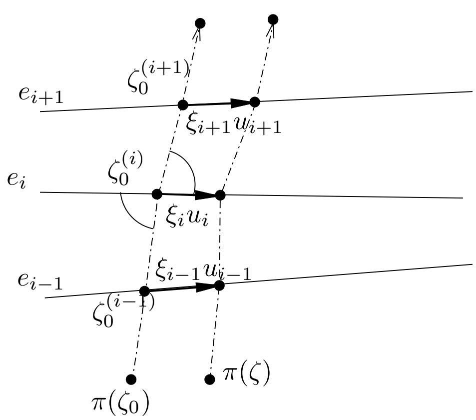

Figure 7: The structure of the shortest paths near a generic point.

$\pi(\zeta)$ satisfies the *local optimality* criterion that asserts that, for each $i < n$, the incoming and outgoing angles that $\pi(\zeta)$ subtends at e_i are equal. More precisely, the criterion is

$$\frac{(\zeta_0^{(i)} + \xi_i u_i - \zeta_0^{(i-1)} - \xi_{i-1}u_{i-1}) \cdot u_i}{\|\zeta_0^{(i)} + \xi_i u_i - \zeta_0^{(i-1)} - \xi_{i-1}u_{i-1}\|} = \frac{(\zeta_0^{(i+1)} + \xi_{i+1}u_{i+1} - \zeta_0^{(i)} - \xi_i u_i) \cdot u_i}{\|\zeta_0^{(i+1)} + \xi_{i+1}u_{i+1} - \zeta_0^{(i)} - \xi_i u_i\|}. \tag{1}$$

Moreover, all these equations also hold when $\xi_1 = \xi_2 = \cdots = \xi_n = 0$. We have

$$\|\zeta_0^{(i)} + \xi_i u_i - \zeta_0^{(i-1)} - \xi_{i-1}u_{i-1}\| =$$

$$\|\zeta_0^{(i)} - \zeta_0^{(i-1)}\| + \frac{(\zeta_0^{(i)} - \zeta_0^{(i-1)}) \cdot (\xi_i u_i - \xi_{i-1}u_{i-1})}{\|\zeta_0^{(i)} - \zeta_0^{(i-1)}\|} + O\left((|\xi_i| + |\xi_{i-1}|)^2\right),$$

and similarly for the other norm. Using the fact that equality holds when the ξ_i's are all 0, it is easily seen that the first-order approximation of (1) is a system of $n-1$ linear homogeneous equations in $\xi_1, \ldots, \xi_n$. Regard this system as equations in $\xi_1, \ldots, \xi_{n-1}$. Our strong general position assumption allows us to assume that the system is nonsingular. It then follows that, up to first-order approximation, each ξ_i is a multiple of ξ_n, which implies the property asserted above, and thus completes the proof of the lemma. $\square$

MICHA SAYS: **Problems and notes: (1) Even when the system is nonsingular, some ξ_i's may solve to 0, and then we need to modify the statement.**

MICHA SAYS: **(2) The coefficients of the system depend on the ζ_i's. If ζ_0 is a GENERIC point, the Lemma is probably OK, but perhaps at some degenerate locations, the system can become singular, in spite of the string general position assumption.**

MICHA SAYS: **(3) So what happens if the system is singular? We simply can express the ξ_i's in terms of several (more than one) free members of this set. Can this happen?! Suppose we express everything in terms of ξ_n and ξ_{n-1}. This means that, up to first order, there is a continuum of shortest paths to the fixed nearby point ξ_n. Can this happen? Maybe some strict convexity rules this out?! Not all hope is lost yet...**

The structure of $\pi(\zeta)$ in the vicinity of a triple collinearity.

Lemma 5 *Suppose that $\zeta_0 \in e = e_n$ and that $\pi(\zeta_0)$ contains a critical triple collinearity, occurring between three consecutive contact points, say $\zeta_0^{(i-1)} \in e_{i-1}$, $\zeta_0^{(i)} \in e_i$, and $\zeta_0^{(i+1)} \in e_{i+1}$. Then, when $\zeta \in e$ is on one side of ζ_0 and sufficiently close to it, $\pi(\zeta)$ bends at e_i, and when $\zeta \in e$ is on the other side and sufficiently close to ζ_0, $\pi(\zeta)$ passes over e_i without touching it.*

Proof: We vary ζ along e through ζ_0, as above, and use the same notations as in the preceding lemma.

Note that the local optimality conditions are now insufficient to characterize $\pi(\zeta)$. In addition, we have to require that if $\pi(\zeta)$ touches e_i then it bends there *downwards*, for otherwise, a local shortcut near the contact point with e_i would result in a shorter path that passes (strictly) above e_i. The condition for bending downwards can be expressed algebraically as follows. Let H_i be the plane spanned by e_i and by the segment $\zeta^{(i-1)}\zeta^{(i)}$. Then $\zeta^{(i)}\zeta^{(i+1)}$ must lie below H_i. Equivalently, the scalar product of $\zeta^{(i)}\zeta^{(i+1)}$ with the upward-directed normal v_i of H_i should be negative. See Figure 8.

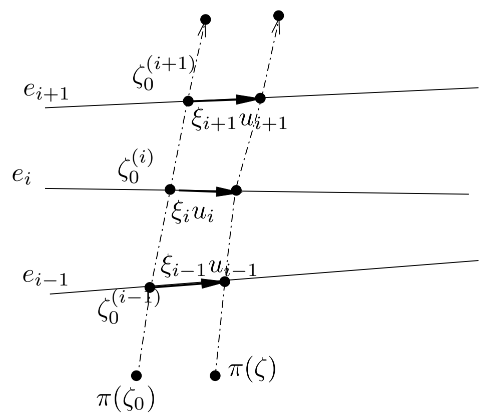

Figure 8: The structure of the shortest paths near a critical triple collinearity.

We can take

$$v_i = u_i \times (\zeta_0^{(i)} - \zeta_0^{(i-1)} + \xi_i u_i - \xi_{i-1}u_{i-1}).$$

Hence, the condition for bending downwards is

$$u_i \times (\zeta_0^{(i)} - \zeta_0^{(i-1)} + \xi_i u_i - \xi_{i-1}u_{i-1}) \cdot (\zeta_0^{(i+1)} - \zeta_0^{(i)} + \xi_{i+1}u_{i+1} - \xi_i u_i) < 0. \tag{2}$$

Since (i) $u_i \times u_i = 0$, (ii) $\zeta_0^{(i)} - \zeta_0^{(i-1)}$ is parallel to $\zeta_0^{(i+1)} - \zeta_0^{(i)}$, and (iii) $u_{i-1} \cdot u_i \times u_{i+1} = 0$ (because all three vectors are parallel to the xz-plane), it follows that (2) is equivalent to

$$\xi_{i+1}u_i \times (\zeta_0^{(i)} - \zeta_0^{(i-1)}) \cdot u_{i+1} - \xi_{i-1}u_i \times u_{i-1} \cdot (\zeta_0^{(i+1)} - \zeta_0^{(i)}) < 0, \tag{3}$$

which is a linear inequality in ξ_{i-1}, ξ_{i+1}.

Now drop the line e_i from the input family, and apply Lemma 4 to the remaining lines. This implies that ξ_{i-1} and ξ_{i+1} are (up to first-order approximation) multiples of ξ_n. Hence, the left-hand side of (3) reverses its sign as ζ passes from one side of ζ_0 to the other side. Hence, when ζ is on one side of ζ_0, $\pi(\zeta)$ bends at e_i, and when ζ is on the other side, $\pi(\zeta)$ passes over e_i without touching it. This completes the proof of the lemma. □

MICHA SAYS: **Again, all the daemons of Lemma 4 raise their ugly heads here too.**

Lemma 6 *For each $i < n$, the set*

$$C_i = \{\zeta \in e_n \mid \pi(\zeta) \cap e_i \neq \emptyset\}$$

is connected.

Proof: Suppose to the contrary that, for some i, C_i is disconnected. Since C_i is closed (as follows from Property 6), this implies that there exist two points $\zeta_1, \zeta_2 \in C_i$ such that the open interval $(\zeta_1, \zeta_2) \subseteq e_n$ is disjoint from C_i. That is, for each $\zeta \in (\zeta_1, \zeta_2)$, $\pi(\zeta) \cap e_i = \emptyset$. Let V_i be the vertical plane containing e_i, and consider the continuous arc

$$\gamma_i = \{\pi(\zeta) \cap V_i \mid \zeta \in [\zeta_1, \zeta_2]\}.$$

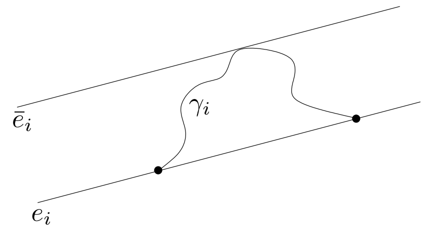

Figure 9: The arc γ_i.

Then γ_i starts and ends on e_i, but all its other points lie above e_i. See Figure 9.

Let $\bar{e}_i$ be the line in V_i that is parallel to e_i, tangent to γ_i, and containing γ_i in the closed halfplane below it. Let $\zeta_0 \in (\zeta_1, \zeta_2)$ be a point for which $\pi(\zeta_0) \cap V_i \in \bar{e}_i$. Consider a new input in which e_i is replaced by $\bar{e}_i$, and we are interested in shortest paths $\bar{\pi}(\zeta)$ from s to points $\zeta \in e_n$ which do not pass below any of the lines $e_1, \ldots, e_{i-1}, \bar{e}_i, e_{i+1}, \ldots, e_{n-1}$. By construction, it is easily checked that $\bar{\pi}(\zeta_0) = \pi(\zeta_0)$. On the other hand, for any point $\zeta \in e_n$ in the vicinity of ζ_0, the path $\pi(\zeta)$ passes below $\bar{e}_i$, and thus $\bar{\pi}(\zeta)$ must bend at $\bar{e}_i$. This however contradicts Lemma 5, and thus completes the proof of the lemma. $\square$

MICHA SAYS: **The proof is plain wrong!! The line $\bar{e}_i$ is in degenerate position, simply by being tangent to γ_i. Then the first-order approximation is meaningless. If the lemma is correct, which it may well be, we need another argument.**

As a corollary, we obtain the following main result of this section,

Theorem 3 *The number of combinatorial changes in the structure of $\pi(\zeta)$, as ζ moves along e_n, is $O(n)$. Thus, the complexity of the shortest path map, restricted to the lines e_i, is $O(n^2)$.*

Proof: As argued above, the combinatorial structure of $\pi(\zeta)$ changes only when the path has a critical triple collinearity. By Lemma 5, as this criticality is encountered, either $\pi(\zeta)$ starts making contact with the middle line e_i of the collinearity, or stops making such a contact. By Lemma 6, once such a contact is lost, it will not materialize again. This is easily seen to imply that the number of combinatorial changes in $\pi(\zeta)$ is only $O(n)$. $\square$

We leave it as an open problem whether Theorem 3 can be exploited to yield a polynomial-time algorithm for computing the shortest-path map on e_n, say.

MICHA SAYS: **We had an unfinished section on "The Case of Two Orthogonal Slopes"**

MICHA SAYS: **If we ever straighten this mess, we should go back to getting an algorithm, based on some sort of parametric searching, to find all the triple critical collinearities.**

4 Hardness Proofs

We show that several special instances of the shortest path problem are NP-hard or NP-complete, thereby extending and sharpening the earlier result of Canny and Reif **??**. We begin by addressing the case of "stacked" rectangles.

We redefine the notion of stacked obstacles, briefly discussed in the introduction. We consider an ordered *stack* of flat polygonal obstacles $\mathcal{S} = (P_1, P_2, \ldots, P_n)$, with P_1 on the bottom and P_n on the top. We allow obstacles that are unbounded polygons. We think of each of the polygons P_i as lying in a plane (a *layer*) parallel to the xy-plane, with an infinitesimal separation in z-coordinate between the plane of P_i and the plane of P_{i+1}. We are given a source point s infinitesimally below the plane of P_1 and a goal point t infinitesimally above the plane of P_n. We seek a Euclidean shortest path from s to t avoiding the obstacles.[1]

We show that the L_2-shortest path problem is NP-complete for a stack $\mathcal{S}$ of axis-aligned rectangles. In fact, we show hardness for a special class of axis-aligned rectangles, namely "q-rectangles" of "types" I and III. An axis-aligned rectangle is a *q-rectangle* ("quadrant-rectangle") if it is semi-infinite in x and semi-infinite in y. There are four distinct types of q-rectangles, depending on the pair of directions in which the rectangle is unbounded. We say that a q-rectangle is of type I (resp., II, III, IV) if it is unbounded in the $+x$ and $+y$ (resp., $-x$ and $+y$, $-x$ and $-y$, $+x$ and $-y$) directions.

Our proof is based on a careful adaptation of the Canny-Reif proof [3] of NP-hardness of the three-dimensional shortest path problem, using several new gadgets.

Theorem 4 *It is NP-complete to decide if there exists an obstacle-avoiding path of Euclidean length at most L among a set of stacked axis-aligned rectangles. In fact, it is NP-complete for the special case where the axis-aligned rectangles are q-rectangles of types I and III.*

Proof: It is easy to see that the problem is in NP, by encoding a solution by the sequence of rectangle edges it "flips over". Note that in this case, the squared length of the path is a quadratic polynomial in the input data; in particular, testing whether the length is at most L can be done in linear time, in the standard real (or rational) exact arithmetic model of computation. We refer to our instance of the shortest path problem for stacked rectangles as SP-STACKED, and our hardness proof is by reduction from 3-SAT. Given an instance of 3-SAT, having n variables $b_1, \ldots, b_n$ and m clauses $C_i = (l_{i1} \vee l_{i2} \vee l_{i3})$, we construct a stack of q-rectangles, $(R_1, \ldots, R_M)$, where $M = O(n+m)$, along with points s and t, such that solving the SP-STACKED problem on this instance permits us to determine whether the 3-SAT formula $\wedge_i C_i$ has a satisfying truth assignment: There exists an obstacle-avoiding path from s to t of length L if and only if the given formula $\wedge_i C_i$ has a satisfying truth assignment.

While our proof attempts to follow that of [3], there are important new aspects that arise. First, we must be careful to make the layout a stacked instance in the plane, so we must modify the clause filter of [3], which exploited spacing of obstacles in the third dimension to have the path classes pass through three literal filters in parallel. Our modification introduces and utilizes a new 3-way splitter. Second, we must avoid the use of "plates with slits" in [3], since our obstacles are all very special *convex* obstacles. Further, we must construct new gadgets using only *two* orientations of obstacle edges, rather than the three orientations (0, $\pi/4$, $\pi/2$) used previously.

Our proof introduces and utilizes several kinds of gadgets: 2-way path splitters (that double the number of shortest path classes), inverted 2-way splitters (that merge back split paths to a common path), 3-way path splitters (that triple the number of shortest path classes, without changing their ordering in the horizontal direction orthogonal to their common direction), inverted 3-way splitters, path shufflers (that perform a perfect shuffle of the input path classes), blockers (which allow a subset of the paths through without disruption, but block other paths, or rather lengthen them),

[1] We note that if there exist non-infinitesimal displacements between the planes containing the obstacles then the length of the path becomes algebraically more involved, and the analysis becomes harder. At the moment we do not know whether our proof continues to hold if we disallow infinitesimal displacements.

literal filters (that "select" paths having a particular "bit", that is, truth assignment to a literal, equal to 0 or 1), and clause filters (that ensure that each clause is true in a common satisfying truth assignment).

Overview of the Construction. Refer to Figure 10. We use n 2-way path splitters to create 2^n distinct path classes that form a "bundle" of parallel paths, all going in the northeast direction (at angle $\pi/4$ with respect to the $+x$-axis). Each path encodes a truth assignment for the n variables.

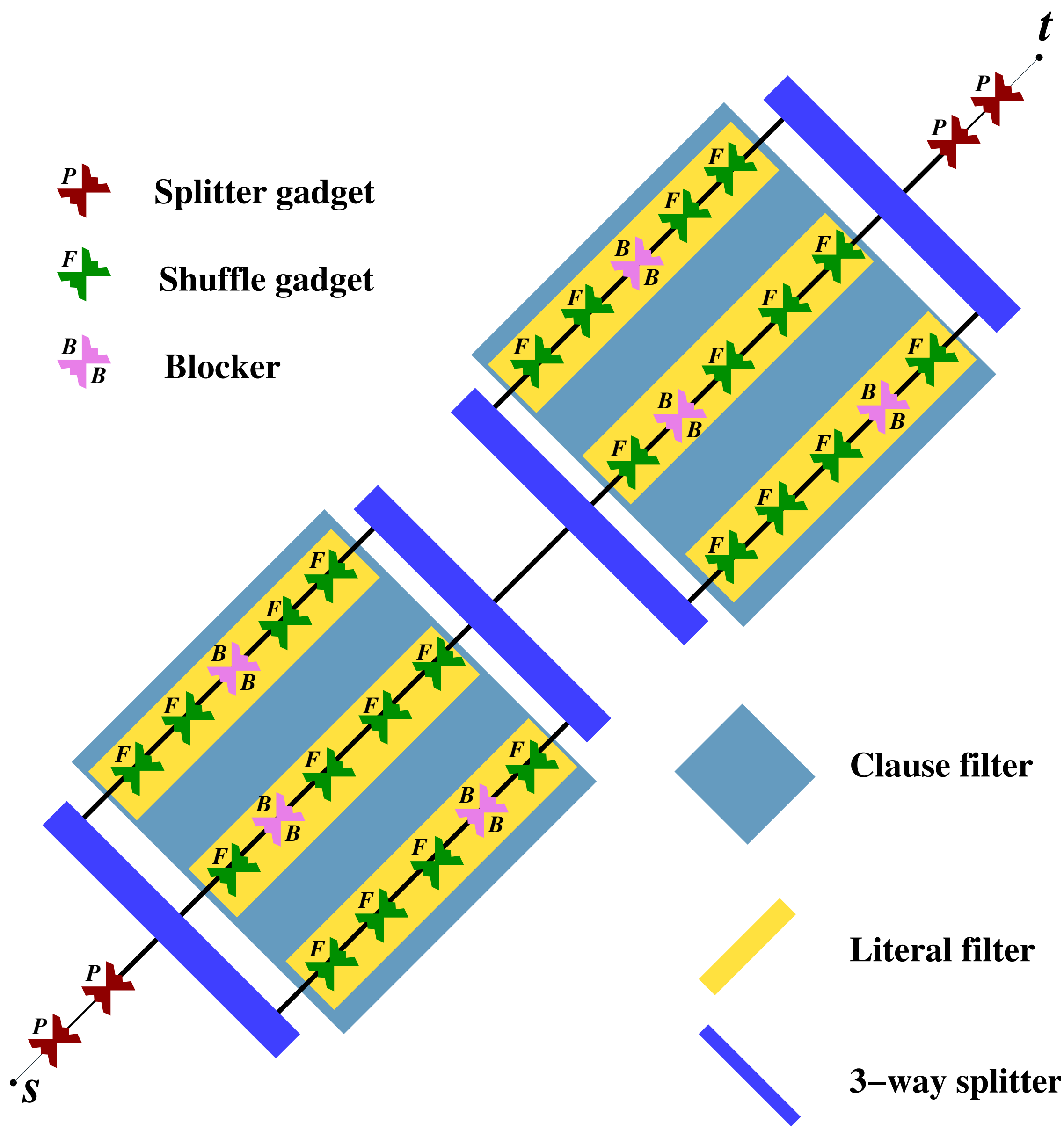

Figure 10: Overview of the construction.

A 3-way path splitter splits this bundle of paths into three parallel bundles, without changing their ordering. These three parallel bundles are then fed into a clause filter, consisting of three literal filters, to filter out those path classes whose corresponding variable assignments fail to satisfy the clause. The property of the literal filter is that only those path classes that have a particular bit (corresponding to the literal) set to 0 or 1 are able to pass through the filter without having a detour (imposed by a blocker) that forces the path to be longer than the others. The literal filter is, in turn, a sequence of shuffle gadgets, each of which rearranges the bundle of paths, so that those corresponding to a particular bit i are all together on the same half of the bundle. Then, a blocker gadget (consisting of one type-I q-rectangle and one type-III q-rectangle) allows only the appropriate half of the path bundle to continue without a detour. Another sequence of shuffle gadgets then puts the paths back into their original ordering in the bundle. The three parallel bundles of paths pass from clause gadget, to inverted 3-way splitter, which puts them back into a common bundle, to 3-way splitter, to clause gadget, to inverted 3-way splitter, etc., and then to a sequence of n inverted 2-way splitters, finally resulting in a single northeast path to the destination point t. Each path splitter and shuffle gadget introduces a certain detour length; we can compute easily the total path length, L, of a path that succeeds in getting from s to t without encountering a blocker. There is a path from s to t of length L if and only if there is a satisfying truth assignment for the input 3SAT instance.

It is important to observe that our gadgets utilize only q-rectangles of types I and III, and that there is a specific ordering of the q-rectangles within each gadget. For two obstacles $\mathcal{O}$ and $\mathcal{O}'$ from different gadgets, we say that obstacle $\mathcal{O}$ precedes (or is beneath) obstacle $\mathcal{O}'$, written $\mathcal{O} \prec \mathcal{O}'$, if either they belong to the same gadget, with $\mathcal{O}$ beneath $\mathcal{O}'$ (in the order within the gadget), or they belong to different gadgets, and there exists a path class encountering the gadget containing $\mathcal{O}$ before the gadget containing $\mathcal{O}'$. By the layout of the construction, with gadgets proceeding from s northeast to t according to a DAG, the relation $\prec$ defines a partial order. The global ordering of the stack of all obstacles is according to a total order that is consistent with the partial order $\prec$. Each q-rectangle has a single vertex (the northeast corner or the southwest corner). Within each gadget the vertices of the associated q-rectangles are localized within a ball of radius proportional to the path bundle width, w. Using a separation of 1 between parallel paths within a path bundle, we see that each of the $O(n+m)$ gadgets requires integral coordinates in their specification of size $O(2^n)$; thus, the entire construction uses only a polynomial number of bits.

We next proceed to describe in detail each gadget used in the construction.

Path Splitters. Consider a path (or a bundle of paths) heading exactly northeast (at angle $\pi/4$ with respect to the x-axis) from the source point s. The 2-way path splitter gadget (Figure 11) consists of three q-rectangles: (1) R_1, of type III (in red or light gray),[2] (2) R_2, of type III (in purple or medium gray), and (3) R_3, of type I (in green or dark gray), stacked on top of a path bundle heading northeast. The dashed (resp., solid) rays and segments in the figure represent the portion of the path that lies beneath (resp., above) the new q-rectangle added to the gadget. After reaching the right boundary of R_1, the path bundle can either continue straight, or fold over the right edge, so it now lies above R_1. The rectangle R_2 lies on top of R_1. The northwest branch continues without interruption from R_2, but the northeast branch folds over the top edge of R_2, now heading southeast. Finally, R_3 is placed on top, so that the region common to the projections of the three rectangles is a *square*. The northwest branch folds over the left edge of R_3, while the southeast branch folds over its bottom edge. It is easily checked that the order of the paths in each of the two new path bundles is the same as their order in the entering bundle. Moreover, the two new path bundles are "tied" in terms of their progress to the northeast (i.e., all the points at equal

[2]The colors can be seen when viewing this paper electronically. MICHA SAYS: **Is this OK for a journal submission?!**

distance from s, in both bundles, lie along a common line of the form $x + y = C$).

The 3-way path splitter gadget is constructed similarly to the 2-way splitter. In fact, one can devise a customized 3-way splitter directly, or one can simply observe that putting two 2-way splitters in sequence results in a 4-way splitter, from which one can use only three of the four path bundles, while blocking the fourth path bundle with an appropriate blocker gadget (described below).

Path Shuffle Gadgets. The path shuffle gadgets are rather intricate, requiring care that the path lengths are preserved. An illustration is shown in Figure 12.

JOE SAYS: **More to be said, I assume....**

MICHA SAYS: **Quite so! Please supply, NOT as a long figure caption. I didnt fully understand the figure...**

Literal Filters. As in [3], a literal filter for variable x_i consists of a sequence of shuffle gadgets, which rearrange the order of the paths within a bundle so that those paths whose ith bit is set to a specific value (0 or 1, according to whether the variable is negated or not in the literal) in the numerical left-to-right ranking $(0, 1, 2, \ldots, 2^n - 1)$ in the bundle are moved to be the left half of the paths. Then, a *blocker* gadget (Figure 13) allows the left half of the paths to proceed without a detour, while the right half of the path bundle gets directed beneath a type I q-rectangle, forcing it to be suboptimal. □

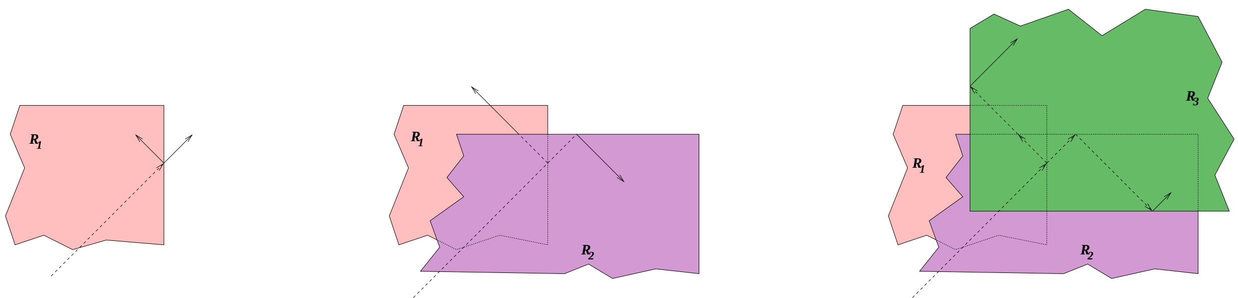

Figure 11: A 2-way path splitting gadget.

With some revisions to the gadgets of Theorem 4, we are able to show also the NP-completeness of the problem of computing shortest paths among stacked rectangles that are vertical and horizontal infinite strips:

Theorem 5 *It is NP-complete to decide if there exists an obstacle-avoiding path of Euclidean length at most L among stacked horizontal and vertical strips.*

Proof: JOE SAYS: **At least a proof sketch to be added.** □

Each of the previous theorems applies also to the case of the L_1 metric, if we rotate the entire construction by $\pi/4$, e.g., so that st is horizontal. The remarkable conclusion is that, while L_1 shortest paths among axis-aligned stacked rectangles is easily solved in polynomial time (just by searching a grid graph), the problem becomes NP-complete for stacked rectangles all at angle $\pi/4$ with respect to the x-axis.

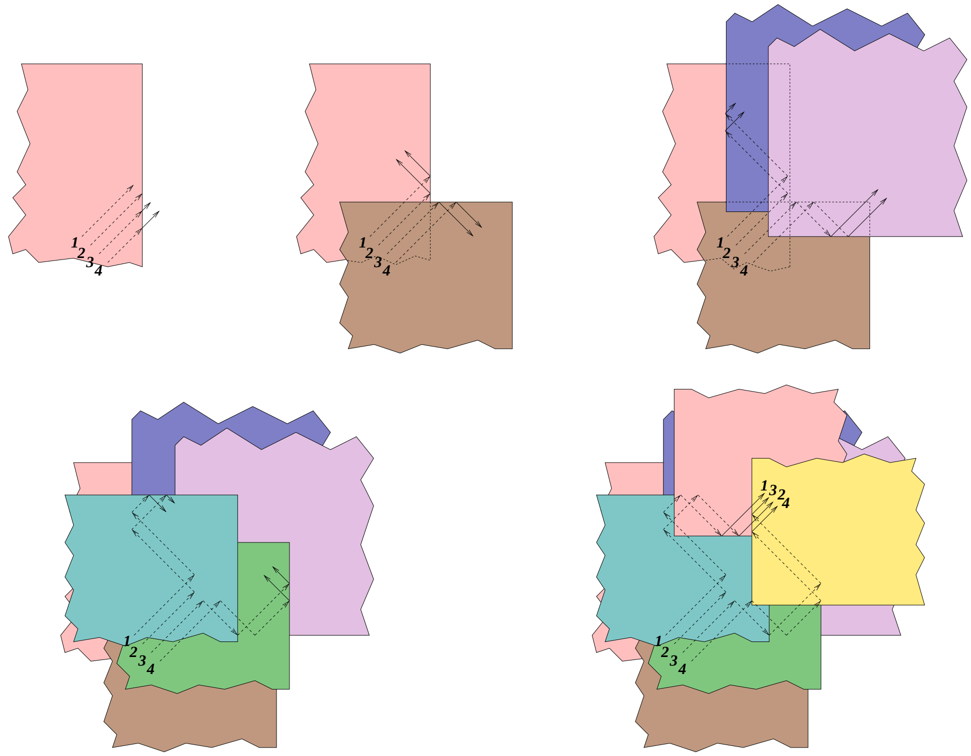

Figure 12: Path shuffle gadget. The path bundle, with paths ordered (1,2,3,4), gets shuffled to a new path bundle (still heading northeast), with paths ordered (1,3,2,4). Each q-rectangle in the stack is either of type I or type III.

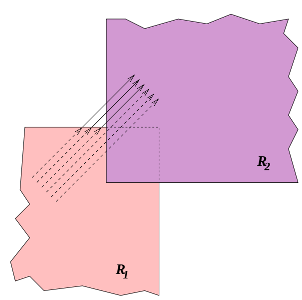

Figure 13: A blocker: the left half of the path bundle emerges from below the (red/light-gray) type III q-rectangle R_1, and passes above the (purple/medium gray) type I q-rectangle R_2, unobstructed. In contrast, the right half of the path bundle remains "trapped" below R_2, and can get on top only by making a significant detour.

Theorem 6 *The L_1 3D shortest path problem is NP-complete for obstacles that are stacked copies of either (1) type-I and type-III q-rectangles, rotated by $\pi/4$; or (2) diagonal strips (at orientation $\pm\pi/4$).*

Finally, we prove hardness of two more instances of the L_1-shortest path problem: that in which the obstacles are stacked two-dimensional disks ("pancakes"), and that in which the obstacles are disjoint balls in $\mathbb{R}^3$.

Theorem 7 *The L_1-shortest path problem is NP-hard for stacked obstacles that are two-dimensional disks in $\mathbb{R}^3$.*

Proof: JOE SAYS: **This is one of the main things I need to add (or we must remove the claim). I need also to draw figures for it.** □

Corollary 2 *The L_1-shortest path problem is NP-hard for obstacles that are disjoint balls in $\mathbb{R}^3$.*

Proof: The corollary is immediate from projection. JOE SAYS: **More to explain...** □

5 Polynomial-Time Cases of Stacked Obstacles

Complementing our hardness/completeness results from the previous section, we consider now cases of n stacked flat obstacles, $\mathcal{S} = (P_1, P_2, \dots, P_n)$, for which we can compute shortest paths in polynomial time.

First, we note that the results of Dror et al. [11] imply that if each of the polygons P_i is the complement of a convex region (i.e., one can cross between successive layers through convex regions), then the problem is solvable in polynomial time, using their algorithm for visiting a sequence of convex regions in the plane. This case includes that in which each P_i is a halfplane.

We focus now on the case in which each P_i is an axis-aligned rectangle, possibly infinite in any of the coordinate directions $(\pm x, \pm y)$. We say that the stack $\mathcal{S}$ of axis-aligned rectangles is *terrain-like* if there is at least one of the four directions (north $(+y)$, south $(-y)$, east $(+x)$, or west $(-x)$) for which each of the polygons P_i is unbounded.

Theorem 8 *Let $\mathcal{S}$ be a stack of n terrain-like axis-parallel rectangles. One can compute a Euclidean shortest path among $\mathcal{S}$ in polynomial time.*

Proof: Without loss of generality, assume that each P_i is unbounded in the $(-y)$-direction, and let $s = (0, 0)$, and $t = (t_x, t_y)$, with $t_y > 0$. Each P_i has a *top edge*, $e_i = (a_i, b_i)$, a left edge l_i (which is a downwards ray from a_i), and a right edge r_i (which is a downwards ray from b_i). It is possible that $a_i = -\infty$ and/or $b_i = +\infty$. The following claim is a special case of the analysis in Section 3.

Claim 1 *An optimal path π^* folds over the top edge of at most one obstacle, and, if it does so, that edge has a y-coordinate greater than t_y. Thus, an optimal path decomposes into at most two y-monotone paths: π^+ that ascends in y to a point $t' \in e^*$ on a (horizontal) top edge e^*, and π^- that descends in y from t' to t.*

Proof: JOE SAYS: **At least a proof sketch to be added.** □

We therefore focus on the case of computing a y-ascending path from s to t'. An optimal path from s to t is obtained by concatenating two such optimal paths, searching over all possible points t' on horizontal (top) edges, e_i, at y-height y_i.

Each pair of distinct obstacles, P_i and P_j with $i < j$, defines up to two points where the boundary of P_i crosses the boundary of P_j.[3] We call such a point u a *crossing point* if there is no obstacle P_k, with $i < k < j$, separating P_i and P_j in the stack at the point u. (We make a nondegeneracy assumption, for simplicity, that no two edges of obstacles are collinear.)

An *elementary path* is a shortest polygonal path, joining two crossing points $u \in \partial P_i \cap \partial P_j$ and $u' \in \partial P_{i'} \cap \partial P_{j'}$, with $i < j \leq i' < j'$, that bends only on left/right edges of obstacles P_k (if any), at points that are not crossing points, in the layers between P_j and $P_{i'}$ ($j < k < i'$) that have $y_k > \min\{y_u, y_{u'}\}$.

Claim 2 *An optimal y-ascending path, π^+, from s to $t' \in e_i$ is a polygonal path that is a sequence of $O(n)$ elementary paths.*

Proof: JOE SAYS: **At least a proof sketch to be added.** □

Claim 3 *Each elementary subpath $\pi(u,u')$ that joins a crossing point $u \in \partial P_i \cap \partial P_j$ to a crossing point $u' \in \partial P_{i'} \cap \partial P_{j'}$ ($i' \geq j$) is a polygonal path with vertices that alternate between left edges and right edges of rectangles P_k, with $j < k < i'$ and $y_k > \min\{y_u, y_{u'}\}$. Each segment of $\pi(u,u')$ makes the same angle, $\theta_{u,u'}$, with respect to the x-axis, where $\theta_{u,u'} = \sin^{-1}((y_{u'} - y_u)/d(u,u'))$, and $d(u,u')$ is the length of $\pi(u,u')$.*

Proof: JOE SAYS: **At least a proof sketch to be added.** □

We construct a graph, $G = (V, E)$, on the set V of crossing points (augmented with s and t'). Each edge $(u,u') \in E$ corresponds to a shortest path between crossing point u and crossing point u', treating each of the obstacles P_k with $y_k > \min\{y_u, y_{u'}\}$ in the layers in between as infinite vertical strips (ignoring their top edges). The lengths of each of the $O(n^4)$ edges (u,u') is computed, in total time $O(n^4)$, by finding first (in time $O(n^3 \log n)$), for each y_i, the all-pairs shortest path distances between the endpoints of the (stacked) line segments that are the projections onto the xz-plane of rectangles P_j that extend to y-height at least y_i (i.e., with $y_j \geq y_i$); the length of (u,u') is given by $\sqrt{(y_{u'} - y_u)^2 + d^2}$, where d is the shortest path length in the xz-projection between the corresponding segment endpoints. We then search the graph G for a shortest path, which is, by the structural results above, an optimal path from s to t'.

The remaining issue is that we do not know the point t', where an optimal s-t path should bend on the top edge (at the y-highest point of π^*). We briefly outline our solution here. Using the graph G, we compute (in time $O(n^4)$) a tree of shortest paths from s to all $O(n^2)$ of the crossing points, and we record with each crossing point its shortest path distance from s. Then, for each choice of crossing point u and top edge e_i, we construct the shortest path map decomposition, $SPM_{e_i}(u)$, of e_i with respect to a source point at u, treating each obstacle P_j with $y_j > y_u$ as if it were an infinite vertical strip. This is readily solved as a shortest path problem in the xz-plane among a stack of infinitesimally separated line segments (the projections of the P_j's with $y_j > y_u$). Each $SPM_{e_i}(u)$ is in fact a partition of e_i into at most two subsegments, due to the special nature of the shortest path problem for stacked segments. For any choice of u, all n of the decompositions $SPM_{e_i}(u)$ are computed in total time $O(n \log n)$, by constructing the shortest path map with respect to u among the relevant segments in the xz-projection. Thus, the shortest path map, $SPM_{e_i}(s)$, with

[3]Technically, these boundaries are infinitesimally apart, so the crossing refers to the crossing of their projections onto the xy-plane.

respect to the source point s can be obtained by computing the lower envelope of the $O(n)$ distance functions, $f_{i,u}(x)$, each of complexity $O(1)$, which give the total distance from s to points, $p(x) \in e_i$, at coordinate x along e_i, using a shortest path from s to u, then a shortest path (treating the P_j's with $y_j > y_u$ as infinite vertical strips) from u to $p(x)$. We similarly compute the decomposition, $SPM_{e_i}(t)$, for each e_i that lies above t.

Once we have the two decompositions, $SPM_{e_i}(s)$ and $SPM_{e_i}(t)$, we can readily compute a shortest path from s to t by examining each edge e_i and computing the point $t' \in e_i$ that minimizes the sum of the distances to s and to t.

The total time required by this algorithm is bounded by $O(n^4)$. □

Figure 14 summarizes the breakdown of the different cases of stacked rectangles into those that are polynomial-time solvable and those that are NP-complete.

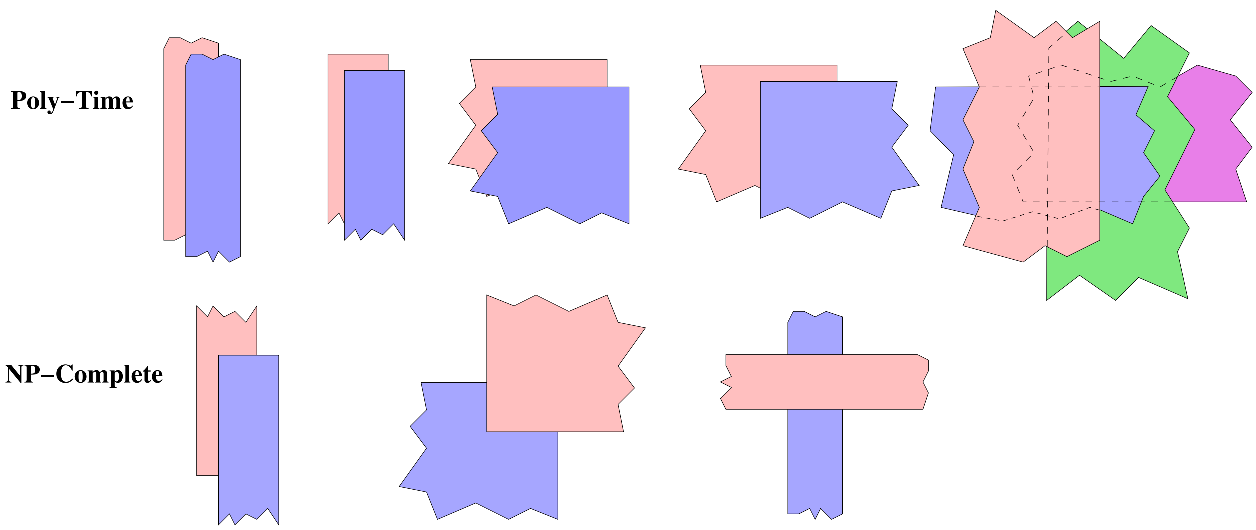

Figure 14: Instances of stacked axis-aligned rectangular obstacles. The top row shows instances that are solvable in polynomial time; the bottom row shows instances that are NP-complete.

6 Conclusion

The goal of this paper has been to add to our understanding of the three-dimensional shortest path problem: Which instances are hard? Which instances can be solved in polynomial time?

We show that a critical property of a set of obstacles in order to assure polynomiality in computing shortest paths is the *terrain property*, that the obstacles all be unbounded in some common direction. We give positive results for several cases, including L_1-shortest paths over arbitrary polyhedral terrains, L_2-shortest paths over parallel walls, and L_2-shortest paths over terrain-like stacked axis-aligned rectangles.

On the flip side, we show that computing Euclidean shortest paths among a set of stacked axis-aligned rectangles is NP-complete, even in the special case that they are quadrants unbounded to the northeast or to the southwest. Indeed, any class of instances of stacked rectangles that does not have the terrain property is shown to be NP-complete, with the exception of the class of halfplanes, which admits a polynomial-time solution.

In the case of L_1 metric, of course, shortest paths among axis-aligned rectangles (or boxes, orthohedral polyhedra, etc.) is readily solved in polynomial time. We give a contrasting result: It is NP-complete to decide if there is a path of L_1 length at most l among a set of stacked horizontal squares, all at angle $\pi/4$ with respect to the coordinate axes. We also show that it is NP-hard to compute L_1-shortest paths among disjoint balls in 3-space, showing that "fatness" is not enough to make the shortest path problem easy.

Many questions remain, among them:

(1) What is the complexity of the Euclidean shortest path problem for flying over a polyhedral terrain? What is the combinatorial complexity of the shortest path map over terrains? (This question is open both for the L_2 and L_1 metrics, except for the case of parallel walls, studied in Section 3.)

(2) What is the complexity of the Euclidean shortest path problem for obstacles that are disjoint balls? We show that the problem is NP-hard for the L_1 metric; however, that proof required a family of balls whose sizes vary drastically (the ratio between largest and smallest radius is exponential in n). What can be said about the L_1- and L_2-shortest path problems for disjoint unit balls?

(3) What is the complexity of the Euclidean or L_1 shortest path problem in the case that *free space* is a union of n balls? Perhaps it is not fatness of the obstacles that is critical, but rather the coverage of free space by a small number of fat subsets of free space.

(4) Can Euclidean shortest paths be computed efficiently for the case of arbitrary stacked terrain polygons? What can be said if the terrain polygons lie in parallel planes, but there may be non-infinitesimal spacing (in z) between the planes? Our analysis of the case of parallel walls (halfplanes) is a step in this direction. The challenge is to understand the combinatorics better, despite the fact that paths become algebraically more involved in the non-stacked instances.

MICHA SAYS: **It is amazing how many references are still only conference versions, after many years of existence! Is there really no journal version?!**

References

[1] C. Bajaj. The algebraic complexity of shortest paths in polyhedral spaces. In *Proc. 23rd Allerton Conf. Commun. Control Comput.*, pages 510–517, 1985.

[2] C. Bajaj. The algebraic degree of geometric optimization problems. *Discrete Comput. Geom.*, 3:177–191, 1988.

[3] J. Canny and J. H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.

[4] J. Chen and Y. Han. Shortest paths on a polyhedron. *Internat. J. Comput. Geom. Appl.*, 6:127–144, 1996.

[5] J. Choi, J. Sellen, and C.-K. Yap. Precision-sensitive Euclidean shortest path in 3-space. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 350–359, 1995.

[6] J. Choi, J. Sellen, and C. K. Yap. Approximate Euclidean shortest paths in 3-space. *Internat. J. Comput. Geom. Appl.*, 7(4):271–295, 1997.

[7] J. Choi and C.-K. Yap. Monotonicity of rectilinear geodesics in d-space. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 339–348, 1996.

[8] K. L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proc. 19th Annu. ACM Sympos. Theory Comput.*, pages 56–65, 1987.

[9] K. L. Clarkson, S. Kapoor, and P. M. Vaidya. Rectilinear shortest paths through polygonal obstacles in $O(n(\log n)^2)$ time. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 251–257, 1987.

[10] M. de Berg, M. van Kreveld, and B. J. Nilsson. Shortest path queries in rectilinear worlds of higher dimension. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 51–60, 1991.

[11] M. Dror, A. Efrat, A. Lubiw, and J. S. B. Mitchell. Touring a sequence of polygons. In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, pages 473–482, 2003.

[12] L. Gewali, S. Ntafos, and I. G. Tollis. Path planning in the presence of vertical obstacles. Technical report, Computer Science, University of Texas at Dallas, 1989.

[13] S. Har-Peled. Constructing approximate shortest path maps in three dimensions. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 383–391, 1998.

[14] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28(6):2215–2256, 1999.

[15] S. Kapoor. Efficient computation of geodesic shortest paths. In *Proc. 31st Annu. ACM Sympos. Theory Comput.*, pages 770–779, 1999.

[16] J. S. B. Mitchell. An optimal algorithm for shortest rectilinear paths among obstacles. In *Abstracts 1st Canad. Conf. Comput. Geom.*, page 22, 1989.

[17] J. S. B. Mitchell. L_1-shortest paths among polygonal obstacles in the plane. *Algorithmica*, 8:55–88, 1992.

[18] J. S. B. Mitchell. Shortest paths and networks. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, 2nd Edition, chapter 27, pages 607–641. CRC Press, 1997.

[19] J. S. B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, eds, *Handbook of Computational Geometry*, pages 633–701. Elsevier Science Publishers, 2000.

[20] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16:647–668, 1987.

[21] C. H. Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Inform. Process. Lett.*, 20:259–263, 1985.

[22] M. Sharir. On shortest paths amidst convex polyhedra. *SIAM J. Comput.*, 16:561–572, 1987.

MICHA SAYS: **Add the Sharir-Schorr REF?!**