CS-1996-12

Approximating Shortest Paths on a Convex Polytope in Three Dimensions[1]

Pankaj K. Agarwal[2] Sariel Har-Peled[3]
Micha Sharir[4] Kasturi R. Varadarajan[5]

Department of Computer Science

Duke University

Durham, North Carolina 27708–0129

August 15, 1996

[1]Work by the first and the fourth authors has been supported by National Science Foundation Grant CCR-93–01259, by an Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, by an NYI award, and by matching funds from Xerox Corporation. Work by the first three authors has been supported by a grant from the U.S.–Israeli Binational Science Foundation. Work by Micha Sharir has also been supported by National Science Foundation Grants CCR-94-24398 and CCR-93-11127, by a Max Planck Research Award, and by a grant from the G.I.F. — the German Israeli Foundation for Scientific Research and Development.

[2] Department of Computer Science, Duke University, pankaj@cs.duke.edu

[3] School of Mathematical Sciences, Tel Aviv University, sariel@math.tau.ac.il

[4] School of Mathematical Sciences, Tel Aviv University, and Courant Institute of Mathematical Sciences, New York University, sharir@math.tau.ac.il

[5] Department of Computer Science, Duke University, krv@cs.duke.edu

Approximating Shortest Paths on a Convex Polytope in Three Dimensions*

Pankaj K. Agarwal† Sariel Har-Peled‡ Micha Sharir§
Kasturi R. Varadarajan¶

August 15, 1996

Abstract

Given a convex polytope P with n faces in $\mathbb{R}^3$, points $s, t \in \partial P$, and a parameter $0 < \varepsilon \leq 1$, we present an algorithm that constructs a path on ∂P from s to t whose length is at most $(1+\varepsilon)d_P(s,t)$, where $d_P(s,t)$ is the length of the shortest path between s and t on ∂P. The algorithm runs in $O(n \log 1/\varepsilon + 1/\varepsilon^3)$ time, and is relatively simple to implement. The running time is $O(n+1/\varepsilon^3)$ if we only want the approximate shortest path distance and not the path itself. We also present an extension of the algorithm that computes approximate shortest path distances from a given source point on ∂P to all vertices of P.

*Work by the first and the fourth authors has been supported by National Science Foundation Grant CCR-93–01259, by an Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, by an NYI award, and by matching funds from Xerox Corporation. Work by the first three authors has been supported by a grant from the U.S.–Israeli Binational Science Foundation. Work by Micha Sharir has also been supported by National Science Foundation Grants CCR-94-24398 and CCR-93-11127, by a Max Planck Research Award, and by a grant from the G.I.F. — the German Israeli Foundation for Scientific Research and Development.

† Department of Computer Science, Duke University, pankaj@cs.duke.edu

‡ School of Mathematical Sciences, Tel Aviv University, sariel@math.tau.ac.il

§ School of Mathematical Sciences, Tel Aviv University, and Courant Institute of Mathematical Sciences, New York University, sharir@math.tau.ac.il

¶ Department of Computer Science, Duke University, krv@cs.duke.edu

1 Introduction

The *three-dimensional Euclidean shortest-path problem* is defined as follows: Given a set of pairwise-disjoint polyhedral objects in $\mathbb{R}^3$ and two points s and t, compute the shortest path between s and t which avoids the interiors of the given polyhedral 'obstacles'. This problem has received considerable attention in computational geometry. It was shown to be NP-hard by Canny and Reif [2], and the fastest available algorithms for this problem run in time that is exponential in the total number of obstacle vertices (which we denote by n) [16, 17]. The apparent intractability of the problem has motivated researchers to develop polynomial-time algorithms for computing approximate shortest paths and for computing shortest paths in special cases.

In the *approximate three-dimensional Euclidean shortest-path* problem, we are given an additional parameter $\varepsilon > 0$, and the goal is to compute a path between s and t that avoids the interiors of the obstacles and whose length is at most $(1+\varepsilon)$ times the length of the shortest path (we call such a path an ε-approximate path). Approximation algorithms for the three-dimensional shortest path problem were first studied by Papadimitriou [12], who gave an $O(n^4(L+\log(n/\varepsilon))^2/\varepsilon^2)$ time algorithm for computing an ε-approximate shortest path, where L is the number of bits of precision in the model of computation. A rigorous analysis of Papadimitriou's algorithm was recently given by Choi et al. [4]. A different approach was taken by Clarkson [5], whose algorithm computes an ε-approximate shortest path in roughly $O(n^2 \log^{O(1)} n/\varepsilon^4)$ time (the complexity of Clarkson's algorithm depends also on an additional parameter).

The problem of computing a shortest path between two points along the surface of a single convex polytope is an interesting special case of the three-dimensional Euclidean shortest-path problem. Sharir and Schorr [18] gave an $O(n^3 \log n)$ algorithm for this problem, exploiting the property that a shortest path on a polyhedron *unfolds* into a straight line. Mount et al. [11] improved the running time to $O(n^2 \log n)$; their algorithm works for non-convex polyhedra as well. Chen and Han [3] gave another algorithm with an improved running time of $O(n^2)$. It is a rather long-standing and intriguing open problem whether the shortest path on a convex polytope can be computed in subquadratic time. This has motivated the problem of finding near-linear algorithms that produce only an approximation of the shortest path. That is, we are given a convex polytope P with n vertices, two points s and t on its surface, and a positive real number ε. Let $\pi_P(s,t)$ denote any shortest path between s and t along the surface of P, and $d_P(s,t)$ denote its length ($\pi_P(s,t)$ is usually, but not always, unique). We want to compute a path on the surface of P between s and t whose length is at most $(1+\varepsilon)d_P(s,t)$. A recent result in this direction is by Hershberger and Suri [10]. They present a simple algorithm that runs in $O(n)$ time, and computes a path whose length is at most $2d_P(s,t)$. Their method does not seem to extend to yield better approximation factors. As mentioned above, the general approximation algorithms [5, 12] have running times worse than quadratic.

In this paper we present another, relatively simple algorithm that computes an ε-approximate shortest path (i.e., a path whose length is at most $(1+\varepsilon)d_P(s,t)$), for any prescribed $0 < \varepsilon \leq 1$. (There is no point of specifying $\varepsilon > 1$, since the algorithm of [10] already solves the problem optimally for $\varepsilon = 1$.) The running time of the algorithm is $O(n \log 1/\varepsilon + 1/\varepsilon^3)$.

The algorithm follows from a sequence of easy but technical lemmas, so we begin with an informal description of our approach. We first estimate the length $d = d_P(s,t)$, using the approximation algorithm described in [10]. We restrict our attention to a polytope Q that is the intersection of P with a cube of side $\approx 2d$ centered at s. (The polytope Q 'preserves' the shortest path that we want to approximate.) We then expand Q by distance $r \approx \varepsilon^{1.5}d$; call this expansion Q_r. We then compute a convex polytope $Q(r)$, with $O(1/\varepsilon^{1.5})$ vertices, that lies between Q and Q_r (i.e., $Q \subseteq Q(r) \subseteq Q_r$), and such that $s,t \in \partial Q(r)$. (The existence of such a polytope $Q(r)$ with only $O(1/\varepsilon^{1.5})$ vertices follows from a result of Dudley [8].) We now compute an exact shortest path σ along $\partial Q(r)$ between s and t. A main technical contribution of this paper is to prove that the length of σ is at most $(1+\varepsilon)d_P(s,t)$. If we only want to approximate $d_P(s,t)$, we can stop now. Otherwise, we apply an additional step that projects σ onto ∂P, in a manner that ensures that the length of the projected path is at most the length of σ, and is thus a good approximation of $\pi_P(s,t)$.

The paper is organized as follows. Section 2 introduces the required terminology and establishes some initial properties, and Section 3 describes an algorithm for computing $Q(r)$. Section 4 describes a simple algorithm for computing a path σ between s and t that lies outside P and whose length is at most $(1+\varepsilon)d_P(s,t)$. Section 5 presents a simple technique for projecting σ onto ∂P, without increasing its length. In Section 6, we derive an extended algorithm for computing approximate shortest-path distances from a given source point on P to each of its vertices. The extended algorithm runs in time $O((n/\varepsilon^3) + (n/\varepsilon^{1.5})\log n)$. We conclude in Section 7 by mentioning a few open problems.

2 Preliminaries

We begin with some terminology and some initial observations. For a face f of the given convex polytope P, we denote by H_f the plane passing through f. Given a set $A \subseteq \mathbb{R}^3$, and a real number $r \geq 0$, let A_r denote the Minkowski sum $A \oplus B_r$, where B_r is a ball of radius r about the origin. That is,

$$A_r = \left\{ x \;\middle|\; \inf_{y \in A} |x-y| \leq r \right\}.$$

For a convex polytope P and for $x \in \partial P$, let $F_r(x) = \left\{ y \;\middle|\; y \in \partial P_r, |x-y| = r \right\}$ be the set of points of ∂P_r 'corresponding' to x. Clearly, ∂P_r consists of planar faces, each being a translated copy, by distance r, of some face of P; of cylindrical faces, each being a portion of a cylinder of radius r about some edge of P; and of spherical faces, each being a portion of a sphere of radius r about some vertex of P. Moreover, (i) if x lies in the relative interior of a face of P, then $F_r(x)$ is a singleton, consisting of the point lying at distance r from x in the direction of the outward normal of the face; (ii) if x lies in the relative interior of an edge of P, then $F_r(x)$ is a circular arc of a circle of radius r about x; and (iii) if x is a

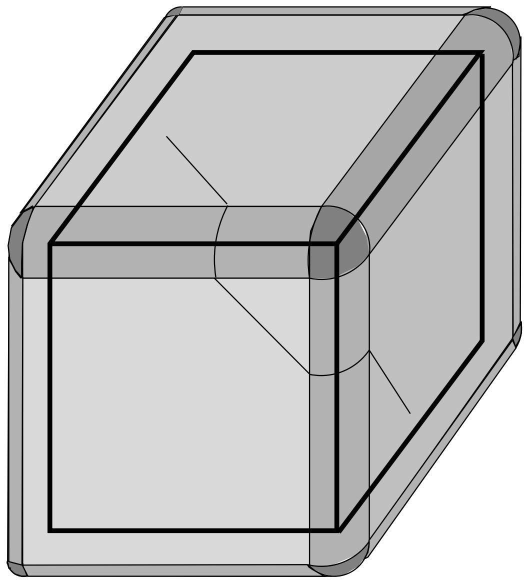

Figure 1: A cube P and the inflated set P_r

vertex of P, then $F_r(x)$ is the entire spherical face associated with x. See Figure 1 for an example of such an inflated polytope.

For any plane H that avoids the interior of P, the *positive half-space* H^+ bounded by H is the one containing P. Such a plane is a *supporting plane* of P if $\partial P \cap H \neq \emptyset$. Given two planes H_1, H_2 that avoid the interior of P, the *wedge* of H_1, H_2 is $W(H_1, H_2) = H_1^+ \cap H_2^+$.

For a curve $\gamma \subset \mathbb{R}^3$ and for $a, b \in \gamma$, we denote by $\gamma(a, b)$ the portion of γ between a and b. An *outer path* of a convex body K is a curve γ connecting two points on ∂K and disjoint from the interior of K. The length of a curve γ is denoted by $|\gamma|$.

A pair $\psi = (\sigma, \mathcal{H})$ is a *supported path* of P if $\sigma = (p_1, \ldots, p_{m+1})$ is a polygonal outer path of P and $\mathcal{H} = (H_1, \ldots, H_m)$ is a sequence of supporting planes of P, such that the line segment $p_i p_{i+1}$ is contained in H_i, for $i = 1, \ldots, m$, and $p_i p_{i+1}, p_{i+1} p_{i+2} \subset \partial W(H_i, H_{i+1})$, for $i = 1, \ldots, m-1$. The *folding angle* between $p_i p_{i+1}$ and $p_{i+1} p_{i+2}$, denoted as $\alpha(H_i, H_{i+1})$, is π minus the dihedral angle of the wedge $W(H_i, H_{i+1})$, for $i = 1, \ldots, m-1$. (Note that this angle depends on the planes H_i, H_{i+1}, and has nothing to with the actual angle between $p_i p_{i+1}$ and $p_{i+1} p_{i+2}$.) The *folding angle* of ψ is $\alpha(\psi) = \sum_{i=1}^{m-1} \alpha(H_i, H_{i+1})$. We will also write $|\sigma|$ as $|\psi|$, and call it the length of ψ. Note that a shortest path $\pi_P(s, t)$ has a natural supported path associated with it. (In this special case, each folding angle $\alpha(H_i, H_{i+1})$ is larger than or equal to π minus the actual angle between $p_i p_{i+1}$ and $p_{i+1} p_{i+2}$; we call this latter angle the *exterior angle* between these segments.)

The following well-known theorem implies that any outer path of P connecting two given points $s, t \in \partial P$ must be of length at least $d_P(s, t)$.

Theorem 2.1 (see [14]) *Let F be a convex surface bounding a body K, and γ a curve that* *does not meet the interior of K and connects points s and t on F. Then the length of γ is at least the shortest distance $d_K(s,t)$ between its endpoints along the surface; it is strictly greater than this distance if the curve does not lie entirely on the surface.*

Lemma 2.2 *Let P be a convex polytope in $\mathbb{R}^3$, let s and t be points on ∂P, and let $\psi = (\sigma, \mathcal{H})$ be a supported path connecting s and t. Then, for each $r \geq 0$, there exists an outer path γ of P_r connecting a point of $F_r(s)$ to a point of $F_r(t)$, whose length is $|\psi| + \alpha(\psi)r$. The curve γ consists of an alternating sequence of straight segments and circular arcs of radius r.*

Proof: Suppose that $\sigma = (s = p_1, \ldots, p_{m+1} = t)$, and $\mathcal{H} = (H_1, \ldots, H_m)$. We replace each segment p_ip_{i+1} of σ by its projection onto the boundary of $(H_i^+)_r$, namely, we translate p_ip_{i+1} by distance r in the direction of the outward normal of H_i. We denote the translated segment by q_iq_i' (where q_i is the image of p_i). Clearly, the total length of the new segments is $|\sigma|$. We also have $q_1 \in F_r(s)$ and $q_m' \in F_r(t)$.

We still need to connect the endpoints q_i' and q_{i+1}, for $i = 1, \ldots, m-1$. Let $W_i = (W(H_i, H_{i+1}))_r$. Clearly, q_iq_i', $q_{i+1}q_{i+1}' \subset \partial W_i$. Moreover, both q_i' and q_{i+1} are images of p_{i+1} under the two projections onto $(H_i^+)_r$ and onto $(H_{i+1}^+)_r$, respectively, and thus can be connected by a circular arc of radius r about p_{i+1}. The length of this arc is clearly $r \cdot \alpha(H_i, H_{i+1})$. The alternating concatenation of the segments q_iq_i' and of these circular arcs yields the desired outer path γ. See Figure 1 for an illustration. The total length of the new path is clearly $|\psi| + \alpha(\psi)r$. Finally, since $P_r \subset W_i$, for $i = 1, \ldots, m-1$, and each point of γ lies on the boundary of one of these inflated wedges, γ is an outer path of P_r. ■

Let $\psi = (\sigma, \mathcal{H})$ be a supported path of P, for $\sigma = (p_1, \ldots, p_{m+1})$ and $\mathcal{H} = (H_1, \ldots, H_m)$. Let $U = (u_1, \ldots, u_m)$ be the sequence of outward unit normals of the planes $H_1, \ldots, H_m$. We create a curve corresponding to ψ on the sphere $\mathcal{S}$ of directions in the following manner. Let γ_i be the shorter great circular arc connecting u_i to u_{i+1} on $\mathcal{S}$, for $i = 1, \ldots, m-1$, and let $\gamma = \bigcup_{i=1}^{m-1} \gamma_i$. Since $|\gamma_i| = \alpha(H_i, H_{i+1})$, it follows that

$$|\gamma| = \sum_{i=1}^{m-1} |\gamma_i| = \sum_{i=1}^{m-1} \alpha(H_i, H_{i+1}) = \alpha(\psi)\,.$$

We call γ the *curve of directions* of ψ.

Lemma 2.3 *Let P be a convex polytope, let s and t be points on ∂P, and let $0 < \varepsilon \leq 1$. Then there exists a supported path $\psi = (\sigma, \mathcal{H})$ of P, such that $|\psi| \leq (1 + \varepsilon/2)d_P(s,t)$, and the folding angle $\alpha(\psi)$ is at most $100/\sqrt{\varepsilon}$.*

Proof: Let $\sigma_0 = (s = p_1, \ldots, p_{m+1} = t)$ be a shortest path on ∂P from s to t, and let $\psi_0 = (\sigma_0, \mathcal{H}_0)$ be the corresponding supported path, where $\mathcal{H}_0 = (H_1, \ldots, H_m)$. Let γ_0 denote the curve of directions of ψ_0 on the sphere $\mathcal{S}$ of directions. Cover $\mathcal{S}$ by at most $100/\varepsilon$ pairwise openly disjoint spherically-convex regions, each contained in some spherical cap with *angular opening* $\sqrt{\varepsilon}$; that is, each cap is the intersection of $\mathcal{S}$ with a cone with apex at the origin and with opening angle $\sqrt{\varepsilon}/2$. This can be easily done, by covering $\mathcal{S}$ with a grid of latitudes and longitudes, with angular spacing of $\sqrt{\varepsilon}/2$ between them, and by placing the caps so that each is centered at a grid point. The number of grid points is

$$\left\lceil \frac{2\pi}{\sqrt{\varepsilon}/2} \right\rceil \cdot \left\lceil \frac{\pi}{\sqrt{\varepsilon}/2} \right\rceil < \frac{100}{\varepsilon},$$

for $0 < \varepsilon \le 1$. Trimming the caps into pairwise openly disjoint spherically-convex regions is also easy to do. For simplicity of exposition, we refer to these regions also as 'caps'. Let $\mathcal{C} = (c_1, \ldots, c_m)$ be the sequence of caps traversed by γ_0 in this order. With no loss of generality, we assume that no vertex of γ_0 lies on a boundary of a cap. This can be enforced by an appropriate perturbation of the caps, if necessary. For each cap c, the intersection $\gamma_0 \cap c$ is a collection of great circular arcs. Our strategy is to modify ψ_0 into a supported path ψ, such that the intersection of the curve of directions of ψ with each cap is either empty or a single connected great circular arc.

For a cap c that appears in the sequence $\mathcal{C}$, let $i(c), j(c)$ be the indices of the first and last appearances of c in $\mathcal{C}$. From the caps $c \in \mathcal{C}$ such that $c \cap \gamma_0$ consists of more than one connected great circular arc, we pick the cap c such that $i = i(c)$ is minimal, and put $j = j(c)$. Let u and u' be the first and last normals in $c \cap \gamma_0$. By assumption, this intersection consists of more than one great circular arc, where the first such arc starts at u and the last arc ends at u'. Let v and v' be, respectively, the first and last points on σ_0 such that the planes supporting P at v and v' have outward normals u and u', respectively. We 'shortcut' σ_0 from v to v' as follows (in fact, this 'shortcutting' may increase the length of σ a little). Draw the planes H, H' that support P at v and v' and have outward normals u and u', respectively. Note that the smaller angle between H and H' is at most $\sqrt{\varepsilon}$. Form the wedge $W = W(H, H')$ between them that contains P. Both v and v' lie on ∂W. (This holds initially, since both v and v' lie on ∂P. Later we will repeat this shortcutting, but in each application the corresponding points v and v' will still lie on ∂P.) Let $\hat{\sigma}_{v,v'}$ be the shortest path along ∂W from v to v'. As noted above, the larger angle θ between its two segments is at least the dihedral angle of W, namely $\ge \pi - \sqrt{\varepsilon}$. This implies, using the cosine law, that $|\hat{\sigma}_{v,v'}| \le (1 + \varepsilon/2)|vv'|$, where $|vv'|$ is the straight Euclidean distance between v and v'.[1] Hence

$$|\hat{\sigma}_{v,v'}| \le (1 + \varepsilon/2)|\sigma_0(v, v')|.$$

We replace $\sigma_0(v, v')$ by $\hat{\sigma}_{v,v'}$, and delete from $\mathcal{C}$ all elements with indices $i + 1, \ldots, j$. We also add the plane H' to an output sequence $\mathcal{H}$ (initialized with the plane H arising in the first cap). It is easily seen that the curve of directions of the modified path intersects c at a single connected great circular arc. By repeating this step as necessary, we end up with a supported path $\psi = (\sigma, \mathcal{H})$, so that the intersection of the curve of directions of

[1] To see this, let w be the middle vertex of $\hat{\sigma}_{v,v'}$, and put $a = |vw|$, $b = |v'w|$ and $c = |vv'|$. Then $c^2 = a^2 + b^2 - 2ab\cos\theta$, or $c^2 = (a+b)^2 - 2ab(1 + \cos\theta)$. Since $1 + \cos\theta \le 1 - \cos\sqrt{\varepsilon} \le \varepsilon/2$, we obtain $c^2 \ge (a+b)^2 \left(1 - \frac{ab\varepsilon}{(a+b)^2}\right)$. Since $(a+b)^2 \ge 4ab$, we have $c \ge (a+b)\sqrt{1 - \varepsilon/4}$, and this is easily seen to imply that $a + b \le (1 + \varepsilon/2)c$, for $0 < \varepsilon \le 1$.

ψ with any cap is either empty or consists of a single connected great circular arc. Note that our rule for picking the cap to 'shortcut' ensures that no portion of the path resulting from a 'shortcut' will participate in a later 'shortcut'. (In particular, the modified portion of the curve of directions after a shortcut remains within the current cap, and thus does not penetrate into any cap that has already been processed.) This implies that each pair of points v, v' between which the shortcuts are made lie on ∂P, so the length of the new path is at most $(1+\varepsilon/2)|\sigma_0| = (1+\varepsilon/2)d_P(s,t)$. We apply a final trimming step to σ, by noting that in general σ may contain two consecutive segments on each plane $H \in \mathcal{H}$. We therefore replace each such pair of segments by a single segment (the sequence $\mathcal{H}$ does not change by this trimming).

The final path $\psi = (\sigma, \mathcal{H})$ satisfies the following properties:

(i) σ does not meet the interior of P.

(ii) σ has at most $100/\varepsilon$ edges.

(iii) $d_P(s,t) = |\sigma_0| \leq |\sigma| \leq (1+\varepsilon/2)d_P(s,t)$.

(iv) The curve of directions of ψ intersects each cap in a single (possibly empty) great circular arc.

It now follows that the total folding angle of ψ is bounded by $100/\sqrt{\varepsilon}$. ■

Remark 2.4 The previous lemma raises the problem whether the folding angle of any shortest path on a convex polytope is bounded by some absolute constant. An extension of this problem to shortest paths on arbitrary convex surfaces is mentioned in [14]. Unfortunately, as shown recently by Pach [13], this problem has a negative answer. For the sake of completeness, we provide in an Appendix an improved variant of Pach's analysis, constructing a family of polytopes with arbitrarily large folding angles of shortest paths along their boundaries. Another, somewhat weaker and still open problem is whether the sum of the exterior angles (as defined above) between consecutive segments of a shortest path on a convex polytope is bounded by an absolute constant. This is a weaker problem because, as already observed, the exterior angle between two consecutive segments of a shortest path is at most the folding angle between the faces containing them.

Lemma 2.5 *Let P, s, t, and ε be as above, and let $r > 0$. For any $s_r \in F_r(s)$, $t_r \in F_r(t)$, there exists an outer path of P_r connecting s_r with t_r, whose length is at most $(1+\varepsilon/2)d_P(s,t) + 2\pi r + 100r/\sqrt{\varepsilon}$.*

Proof: By Lemma 2.3, there exists a supported path ψ of P connecting s to t of length at most $(1+\varepsilon/2)d_P(s,t)$, such that $\alpha(\psi) \leq 100/\sqrt{\varepsilon}$. By applying Lemma 2.2 to ψ, we obtain an outer path γ of P_r whose length is at most $(1+\varepsilon/2)d_P(s,t) + 100r/\sqrt{\varepsilon}$, which connects some point $x \in F_r(s)$ with some point $y \in F_r(t)$. Extending γ to connect s_r to t_r lengthens γ by at most $2\pi r$, so the resulting curve satisfies the properties asserted in the lemma. ■

3 Approximating a Polytope

Let Q be a convex polytope in $\mathbb{R}^3$, with n vertices, that is contained in a unit ball, and let $0 < \mu \leq 1$ be a real parameter. In this section, we present an algorithm to compute a convex polytope $Q(\mu)$, with $O(1/\mu)$ vertices, such that $Q \subseteq Q(\mu) \subseteq Q_\mu$. The algorithm is a straightforward implementation of the constructive proof of Dudley [8], which asserts the existence of a $Q(\mu)$ with $O(1/\mu)$ vertices. We first outline Dudley's proof, and then describe an efficient implementation of his scheme.

Let $C \subseteq \mathbb{R}^3$ be a convex set. A *point-normal pair* on C is an ordered pair (p, η_p) such that (1) $p \in \partial C$, (2) the plane $H(p, \eta_p)$ that passes through p and whose normal is η_p is a supporting plane of C, and (3) η_p is an outward normal to C at p. We denote by $H^+(p, \eta_p)$ the halfspace containing C that is bounded by $H(p, \eta_p)$. For any $\delta > 0$, we call a collection S of point-normal pairs on C a *δ-dense* set if for any point-normal pair (q, η_q) on C, there exists a $(p, \eta_p) \in S$ such that $d(p, q) \leq \delta$, and the angle between η_p and η_q is at most δ. We let $P(S)$ denote the intersection of all the halfspaces corresponding to elements of S, i.e., $P(S) = \bigcap_{(p,\eta_p)\in S} H^+(p, \eta_p)$.

Lemma 3.1 (Dudley [8]) *Let C be a convex body in $\mathbb{R}^3$, and let A be a δ-dense set on C. Then $P(A) \subseteq C_{2\delta^2}$.*

Hence, C can be approximated by computing a 'small' dense set on C. The existence of such a set is guaranteed by the following lemma:

Lemma 3.2 (Dudley [8]) *Let C be a convex set in $\mathbb{R}^3$ that is contained in a unit ball. Then there exists a δ-dense set A on C of size $O(1/\delta^2)$.*

Proof: We only give a sketch of the proof, and refer to [8] for full details. Let B be a ball of radius at most 2, so that for all $x \in C$ and for any $|y| \leq 1$, one has $x + y \in B$. Clearly, $C \subset B$.

Let $A' \subseteq \partial B$ be a collection of points such that for any $q \in \partial B$, there is a $p \in A'$ such that $d(p, q) \leq \delta$. Such a set can be easily constructed by 'covering' ∂B by a grid of latitudes and longitudes (as in the proof of Lemma 2.3). Moreover, $|A'| = O(1/\delta^2)$.

Given a point $p \in \partial B$, let $n(p)$ denote the point-normal pair on C that is formed by taking the unique point $p_C \in \partial C$ that is closest to p, and the unit normal in the direction $\vec{pp_C}$. Let $A = \{n(p) \mid p \in A'\}$.

Dudley shows (see [8]) that A is a 2δ-dense set on C, and this implies the theorem. ■

Theorem 3.3 *Let Q be a convex polytope in $\mathbb{R}^3$ contained in a unit ball, let n denote the number of its vertices, and let $\mu > 0$ be a real parameter. Then one can compute, in $O(n+(1/\mu)\log(1/\mu))$ time, a polytope $Q(\mu)$ with $O(1/\mu)$ vertices such that $Q \subseteq Q(\mu) \subseteq Q_\mu$.*

Proof: Let B be a ball of radius 2 concentric with a unit ball that contains Q. We first compute a set $A' \subseteq \partial B$ such that for any $q \in \partial B$, there is a $p \in A'$ so that $d(p, q) \leq \sqrt{\mu}/4$.

Such a set can easily be calculated in $O(1/\mu)$ time, by 'covering' ∂B with a regular grid of resolution $O(\sqrt{\mu})$, as in the proof of Lemma 3.2.

Next, we construct, in $O(n)$ time, the Dobkin-Kirkpatrick hierarchical decomposition [6, 7] of Q, so that the closest point in Q to any query point can be computed in $O(\log n)$ time. Using this hierarchy, we compute, for each point $p \in A'$, the point-normal pair $n(p)$ on Q corresponding to p, thereby obtaining the set $A = \{n(p) \mid p \in A'\}$ in a total of $O((1/\mu)\log n)$ time.

We next compute the polytope $Q(\mu) = P(A)$, which is the intersection of $O(1/\mu)$ halfspaces, $O((1/\mu)\log(1/\mu))$ time [15]. By Lemma 3.1, the polytope $Q(\mu)$ satisfies the properties asserted in the theorem. The overall time needed to compute $Q(\mu)$ is

$$O\left(n + \frac{1}{\mu}\log n + \frac{1}{\mu}\log\frac{1}{\mu}\right) = O\left(n + \frac{1}{\mu}\log\frac{1}{\mu}\right) .$$

■

4 Approximating $d_P(s,t)$

Let P be a convex polytope in $\mathbb{R}^3$, s and t two points on ∂P, and $0 < \varepsilon < 1$ a real parameter. We present a simple algorithm for computing an outer path of P from s to t whose length is at most $(1+\varepsilon)d_P(s,t)$. For $l > 0$, let $B(s,l)$ be the cube of side $2l$ centered at s, i.e.,

$$B(s,l) = \left\{p \;\middle|\; |p_x - s_x| \le l, |p_y - s_y| \le l, |p_z - s_z| \le l\right\} .$$

Algorithm Approximate-Path

1. Compute a value Δ such that $d_P(s,t) \le \Delta \le 2d_P(s,t)$, using the algorithm of Hershberger and Suri [10]. Compute $Q = P \cap B(s, 2\Delta)$.
2. Set $r = \varepsilon^{3/2}\Delta/440$. Using the scheme described in Section 3, compute a polytope $Q(r)$ such that $Q \subseteq Q(r) \subseteq Q_r$.
3. Let H_s^+ (resp. H_t^+) be any halfspace that has s (resp. t) on its boundary and that contains Q. Compute $Q(r) \cap H_s^+ \cap H_t^+$. Abusing notation, let $Q(r)$ now denote this new polytope. Note that $s, t \in \partial Q(r)$.
4. Compute a shortest path σ between s and t on $\partial Q(r)$, using the algorithm of Chen and Han [3].

If we are interested only in computing an approximate value of $d_P(s,t)$, we return $|\sigma|$. However, if we also want to compute an ε-approximate shortest path between s and t on ∂P, we complete the algorithm by projecting σ on ∂P, using an additional procedure described in Section 5.

We prove that σ is an outer path of P whose length is at most $(1+\varepsilon)d_P(s,t)$, and then analyze the running time of the algorithm.

Lemma 4.1 *The algorithm* Approximate-Path *returns an outer path of P between s and t whose length is at most $(1+\varepsilon)d_P(s,t)$, and which consists of $O(1/\varepsilon^{1.5})$ segments.*

Proof: First observe that the cube $B(s, 2\Delta)$ is chosen large enough so that if π is a shortest path between s and t on P (and so $|\pi| < 2\Delta$), π lies in the interior of $B(s, 2\Delta)$, and is therefore also a shortest path on Q. By Lemma 2.5, there exists an outer path ξ of Q_r connecting any two points $s_r \in F_r(s)$ and $t_r \in F_r(t)$, whose length is at most $(1+\varepsilon/2)d_P(s,t) + 2\pi r + 100r/\sqrt{\varepsilon}$. As a consequence of step 3 of the algorithm, we may choose s_r and t_r such that the segments ss_r and tt_r do not intersect the interior of $Q(r)$. The path obtained by concatenating the segments ss_r and $t_r t$ to the beginning and the end of ξ, respectively, is an outer path of $Q(r)$ connecting s and t. It follows from Theorem 2.1 that the length of the shortest path σ on $Q(r)$ between s and t, computed by the algorithm, is at most $|\xi| + |ss_r| + |tt_r|$. Since $|ss_r|, |tt_r| \leq r$, we obtain, by the choice of r,

$$|\sigma| \leq (1+\varepsilon/2)d_P(s,t) + (2\pi+2)r + 100r/\sqrt{\varepsilon} \leq (1+\varepsilon)d_P(s,t).$$

Observe that any outer path of Q lying inside $B(s, 2\Delta)$ is also an outer path of P. Since $0 < \varepsilon < 1$, the above inequality implies that σ lies in the interior of the cube $B(s, 2\Delta)$. The path σ is clearly an outer path of Q, therefore σ is an outer path of P too.

The polytope $Q(r)$ has $O(1/\varepsilon^{1.5})$ faces, and any shortest path intersects a face of the polytope along (at most) a single segment, therefore σ consists of at most $O(1/\varepsilon^{1.5})$ segments. ■

Lemma 4.2 *The running time of the algorithm* Approximate-Path *is $O(n+1/\varepsilon^3)$, where n is the number of faces of P.*

Proof: Hershberger and Suri's algorithm [10] runs in $O(n)$ time. The polytope Q can be calculated in linear time. By Theorem 3.3, $Q(r)$ can be computed in $O(n + (1/\varepsilon^{1.5})\log 1/\varepsilon)$ time. The number of faces in $Q(r)$ is $O(1/\varepsilon^{1.5})$, and the algorithm by Chen and Han used in step 4 runs in quadratic time. Hence, computing σ takes $O(1/\varepsilon^3)$ time. Summing up all these bounds, the overall running time of the algorithm is $O(n + 1/\varepsilon^3)$, as asserted. ■

Putting these lemmas together, we obtain the following result.

Theorem 4.3 *Given a convex polytope P with n faces, two points s and t on ∂P, and a parameter $\varepsilon > 0$, a polygonal outer path of P between s and t whose length is at most $(1+\varepsilon)d_P(s,t)$, and which consists of $O(1/\varepsilon^{1.5})$ segments, can be computed in time $O(n+1/\varepsilon^3)$.*

5 Projecting an Outer Path to a Polytope

In this section we present an algorithm for projecting a polygonal outer path σ of a polytope P onto the surface of P. The output of the algorithm is a polygonal path σ_{out} such that:

(i) $|\sigma_{out}| \leq |\sigma|$.

(ii) $\sigma_{out} \subset \partial P$.

(iii) σ and σ_{out} have the same pair of endpoints.

(iv) The number of segments of σ_{out} is at most n, the number of faces of P.

We assume that ∂P is triangulated, so each face is a triangle.

Let σ be the given polygonal outer path of P, connecting $s \in \partial P$ to $t \in \partial P$ and consisting of m segments. We direct σ from s to t. For a face f of P that contains s, let w be the last intersection point of σ with H_f (the plane supporting f). Clearly, the path $\sigma' = sw \| \sigma(w, t)$ is not longer than σ, and $\sigma' \subseteq H_f^+$. We denote by Project-on-Face(f,σ) the procedure that returns w, the last intersection point of σ with H_f.

The projection algorithm is presented in Figure 2. An illustration of a single step of the main loop of the algorithm is shown in Figure 3.

Algorithm Project-Path(P, σ)
  `Input:` A convex polytope P and an outer path σ
  `Output:` A path σ_{out} on the boundary of P
begin
  $s \leftarrow$ starting-point(σ), $t \leftarrow$ end-point(σ)
  $\sigma_{curr} \leftarrow \sigma$, $\sigma_{out} \leftarrow \emptyset$
  while $s \neq t$ do
(A) $f \leftarrow$ any face of P containing s and not yet visited
    $w \leftarrow$ Project-on-Face(f, σ_{curr})
    $sv \leftarrow sw \cap f$
    $\sigma_{out} \leftarrow \sigma_{out} \| sv$, $\sigma_{curr} \leftarrow vw \| \sigma_{curr}(w, t)$
    $s \leftarrow v$
  end while
  return σ_{out}
end Project-Path

Figure 2: Algorithm for projecting an outer path onto the boundary of a polytope

Lemma 5.1 *The operation (A) in the algorithm* Project-Path *never fails, i.e., there always exists a face adjacent to the current s that was not yet visited.*

Proof: Assume, for the sake of contradiction, that (A) does fail at some point, call it p. Let F denote the collection of faces of P that contain p, and let $K = \bigcap_{f \in F} H_f^+$. Let q be the last intersection point of σ_{curr} with ∂K. Note that $q \neq p$, for otherwise $\sigma_{curr}(p, t)$ has to move from p away from at least one halfspace H_f^+, for some $f \in F$, which is impossible, since, after f has been processed, σ_{curr} is always contained in H_f^+. Note also that $q \neq t$, for

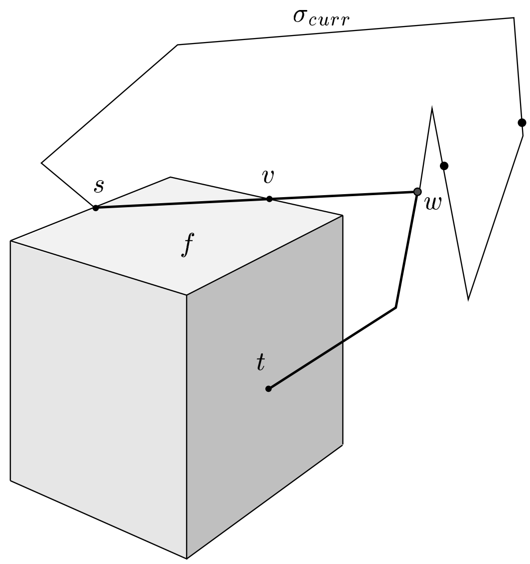

Figure 3: A single step of the main loop of PROJECT-PATH; σ_{curr} crosses H_f at three marked points, the last of which is w.

otherwise the algorithm would have terminated when processing any face $f \in F$ such that $t \in H_f$. It is also easily checked that q lies on the original path σ. Let f be a face such that $q \in H_f$, and let ξ be the last intersection point of the segment $\vec{pq}$ with f. Since $t \in \partial P \subseteq K$, and $t \neq q$, the subpath $\sigma(q,t) \setminus \{q\}$ lies in the interior of $K \subseteq H_f^+$, and therefore q is also the last intersection point of σ with H_f. Since σ_{curr} is an outer path of P, $t \in \partial P$, and K is the same as P in a sufficiently small neighborhood of p, it is easy to see that pq lies on f in a sufficiently small neighborhood of p, which implies that $\xi \neq p$.

Suppose f has already been visited. Then let σ' be σ_{curr} immediately before f was visited, and let $s'q'$ be the segment that was added to σ_{out} while processing f. Notice that q is also the last intersection point of σ' with H_f, because $q \in \sigma$, and thus $\sigma_{curr}(q,t) = \sigma'(q,t)$. Therefore $\sigma'(s',q)$, the initial portion of σ' up to q, would have been replaced, at the step where f is processed, by the segment $q'q$. Moreover, the algorithm runs in a manner that guarantees that the portion of σ_{curr} on H_f can only be shortened (or eliminated) in subsequent steps of the algorithm. It follows that $\sigma_{curr}(p,q) = pq \subseteq q'q$ and thus $\xi \neq q' \in q'q$. By construction, q' is the last intersection point of the segment $s'q$ with f. Therefore the segment $q'q$ does not intersect f except at q', which is false, since $\xi \in f$. This contradiction implies that f has not yet been visited, and thus completes the proof of the lemma. ■

Lemma 5.2 *The output path σ_{out} produced by algorithm* PROJECT-PATH*(P,σ) satisfies the properties (i)–(iv).*

Proof: The algorithm PROJECT-PATH performs at most n iterations (Lemma 5.1 implies that the algorithm terminates properly). Each iteration of the while loop adds one segment to the polygonal output path σ_{out}, thus σ_{out} consists of at most n segments. Each such segment is contained in ∂P, thus $\sigma_{out} \subset \partial P$. Since σ_{out} is generated from σ by performing a sequence of calls to PROJECT-ON-FACE, it follows that $|\sigma_{out}| \le \sigma$. Finally, σ_{out} and σ clearly have the same pair of endpoints. ■

We next present a reasonably efficient technique for implementing the procedure PROJECT-ON-FACE. Note that σ_{curr}, with the possible exception of the first edge, is a subpath of σ, ending at t, and that each iteration in PROJECT-PATH replaces an initial subpath of σ_{curr} by a line segment, possibly also truncating the first surviving segment of the preceding path. Set $b = \lceil \log m \rceil + 1$. We divide σ_{curr} into two subpaths σ_1 and σ_2; σ_1 is an initial subpath of σ_{curr}, consisting of at most b edges, and σ_2 is the rest of σ_{curr}. We compute the Gaussian diagram of $\mathrm{Conv}(\sigma_2)$, and preprocess it into a data structure for answering point-location queries in the Gaussian diagram. The preprocessing takes $O(m \log m)$ time, and each point-location query takes $O(\log m)$ time [15]. The subpaths σ_1, σ_2 change as the algorithm progresses, as will be described below. Initially, σ_1 consists of the first b edges of σ (and σ_2 consists of all the remaining segments).

Let $f \in P$ be a query face, and let $\mathbf{n}$ denote the outward normal of f. We locate $\mathbf{n}$ in the Gaussian diagram of $\mathrm{Conv}(\sigma_2)$, and thus obtain the vertex $\xi \in \mathrm{Conv}(\sigma_2)$ that the supporting plane of $\mathrm{Conv}(\sigma_v)$ with outward normal $\mathbf{n}$ touches. If $\xi \in \mathrm{int}\,(H_f^+)$, that is, H_f does not intersect σ_2, we traverse σ_1 and compute the last intersection point of σ_1 with H_f in $O(b) = O(\log m)$ time. σ_{curr} is updated as described in the algorithm PROJECT-PATH. σ_2 and its starting point ξ remain the same, and we set $\sigma_1 = \sigma_{curr}(s, \xi)$.

If H_f intersects σ_2, we traverse σ_2, compute the last intersection point of σ_2 with H_f, in $O(m)$ time, and update σ_{curr} as described in the algorithm PROJECT-PATH. If σ_{curr} has at most b edges, then we set $\sigma_1 = \sigma_{curr}$ and $\sigma_2 = \emptyset$; otherwise, σ_1 consists of the first b edges of σ_{curr}, and σ_2 consists of the remaining edges. We compute the Gaussian diagram of $\mathrm{Conv}(\sigma_2)$, in $O(m \log m)$ time, and preprocess it in additional $O(m \log m)$ time for point-location queries, as described above.

This PROJECT-ON-FACE procedure is clearly correct. We call a query *short* if the last intersection point of σ with H_f lies in σ_1, and *long* otherwise. A short query takes $O(\log m)$ time, and a long query takes $O(m \log m)$ time. Suppose the last intersection point of σ_{curr} with H_f lies on the $(k+1)$-st edge of σ_{curr}. Then the first k edges of σ_{curr} are deleted, the $(k+1)$-st edge of σ_{curr} is truncated, and possibly a new edge is added in front of the trimmed $(k+1)$-st edge; see Figure 3. Therefore, the number of edges in σ_{curr} is reduced by $k-1$. Since $k \ge b$ for a long query, each long query reduces the number of edges in σ_{curr} by at least $b-1$, thereby implying that the number of long queries is at most $m/(b-1) = m/\lceil \log m \rceil$. The while loop in PROJECT-PATH is executed at most n times, therefore the overall running time of the algorithm is $O(n \log m + m^2)$. We have thus shown:

Theorem 5.3 *Let P be a convex polytope with n faces, and let σ be a polygonal outer path of P consisting of m segments. One can construct, in $O(n \log m + m^2)$ time, a polygonal* *path $\sigma' \subset \partial P$ with the same endpoints as σ, such that $|\sigma'| \le |\sigma|$, and the number of edges of σ' is at most n.*

The algorithm described in the previous section returns an outer path with at most $O(1/\varepsilon^{1.5})$ edges. We can thus project it onto ∂P in additional $O(n \log 1/\varepsilon + 1/\varepsilon^3)$ time. Combining this with the bound in Theorem 4.3, we obtain the main result of the paper:

Corollary 5.4 *Let P be a convex polytope with n faces, s and t two points on ∂P, and $\varepsilon > 0$ a real parameter. Then a polygonal path between s and t on ∂P, consisting of at most n segments, whose length is at most $(1+\varepsilon)d_P(s,t)$, can be computed in time $O(n \log 1/\varepsilon + 1/\varepsilon^3)$.*

6 Approximate Shortest-Path Distances to All Vertices

We can generalize the above algorithm to compute approximate shortest-path distances from a given source point s on the polytope P to each of its vertices, in time $O((n/\varepsilon^3) + (n/\varepsilon^{1.5}) \log n)$ time. That is, for each vertex v of P, the algorithm computes a real value Δ_v such that $d_P(s,v) \le \Delta_v \le (1+\varepsilon)d_P(s,v)$. Hershberger and Suri [10] present another algorithm that runs in $O(n \log n)$ time and computes crude approximations of the shortest path distances from s to all the vertices of P; for each vertex v, the distance computed is between $d_P(s,v)$ and about $2.4d_P(s,v)$. The first step of our algorithm applies this procedure and obtains these crude approximations. Next we compute, in $O(n)$ time, the Dobkin-Kirkpatrick hierarchical decomposition of the polytope P [6, 7], using which we can compute the closest point in P to a query point in $O(\log n)$ time.

To compute an approximate shortest path from s to a vertex v of P, we use the following slight variation of the algorithm Approximate-Path. Let $\Delta_v \le 2.4d_P(s,v)$ be the crude approximation of the shortest path distance $d_P(s,v)$, obtained by the Hershberger-Suri algorithm. Let $r_v = c\varepsilon^{3/2}\Delta_v$, where c is a sufficiently small constant, and let $Q = P \cap B(s, 2\Delta_v)$. As in the Approximate-Path algorithm, we use Dudley's scheme to compute a polytope $Q(r_v)$, with $O(1/\varepsilon^{1.5})$ vertices, such that $Q \subseteq Q(r_v) \subseteq Q_{r_v}$. However, we cannot afford to explicitly compute each of the Q's, so we have to carry out the steps in the algorithm of Theorem 3.3 without working with Q explicitly. To this end, we observe that, by slightly modifying the query procedure, the Dobkin-Kirkpatrick hierarchical decomposition of P itself can actually be used to compute the closest point in $Q = P \cap B(s, 2\Delta)$ to a query point ξ, in $O(\log n)$ time, as follows. We first compute the point p on ∂P closest to ξ. If $p \in B(s, 2\Delta)$, we are done. Otherwise, the closest point to ξ in Q lies on the boundary of $B(s, 2\Delta)$. More precisely, as is easily checked, this point lies on a face of the box $B(s, 2\Delta)$. Dobkin and Kirkpatrick [7] have shown that, for a query plane h and a query point ξ, the point in $h \cap P$ closest to ξ can be computed in $O(\log n)$ time, using the hierarchical decomposition of P. A slight variant of their procedure can also compute the point in $f \cap P$ closest to ξ, where f is a face of $B(s, 2\Delta)$. Repeating this procedure for all six faces of $B(s, 2\Delta)$, the closest point in Q to a query point ξ can be computed in $O(\log n)$ time. The remaining steps in the algorithm of Theorem 3.3 are readily modified, yielding an $O(\frac{1}{\varepsilon^{1.5}}(\log n + \log \frac{1}{\varepsilon}))$-time procedure for computing $Q(r_v)$.

Using $Q(r_v)$, we proceed, as in the APPROXIMATE-PATH algorithm, to compute an outer path between s and v whose length is at most $(1+\varepsilon)d_P(s,v)$. Summing up, this procedure takes $O((1/\varepsilon^3) + (1/\varepsilon^{1.5})\log n)$ time for a single vertex of P. Iterating over all vertices, we get an algorithm that computes approximate distances from s to all vertices of P in $O((n/\varepsilon^3) + (n/\varepsilon^{1.5})\log n)$ time.

7 Conclusions

In this paper we have presented a simple and efficient algorithm for computing approximate shortest paths on the surface of a convex polytope in 3-space. We conclude by mentioning a few open problems.

- Can the approximate shortest path between two points on a polyhedral terrain, or on the surface of a nonconvex polyhedron, be computed in time that is near-linear in the number of faces?
- Can we preprocess a convex polytope in near-linear time into a data structure so that the approximate distance between any two query points on the polytope can be computed in logarithmic time? Can we achieve this if we want to compute the approximate distance from a fixed point to any query point?
- Finally, can the exact shortest path between two points on a convex polyhedron be computed in near-linear time? in subquadratic time?

Acknowledgments

The authors wish to thank Boris Aronov, Imre Bárány, Ken Clarkson, Alon Efrat, Joe Malkevitch, János Pach, Subhash Suri, and Boaz Tagansky for helpful discussions concerning the problems studied in this paper and related problems. In particular, the construction in the Appendix of shortest paths with arbitrarily large folding angles is a variant of a construction initially provided by János Pach.

References

[1] A. Baltsan and M. Sharir, On shortest paths between two convex polyhedra, *J. Assoc. Comput. Mach.* 35 (1988), 267–287.

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

[3] J. Chen and Y. Han, Shortest paths on a polyhedron, *Proc. 6th Annu. ACM Symp. Comput. Geom.*, 1990, 360–369.

[4] J. Choi, J. Sellen, and C. K. Yap, Approximate Euclidean shortest paths in 3-space, *Proc. 10th Annu. ACM Symp. Comput. Geom.*, 1994, 41–48.

[5] K.L. Clarkson, Approximation algorithms for shortest path motion planning, *Proc. 19th ACM Symp. Theory Comput.*, 1987, 56–65.

[6] D. Dobkin and D. Kirkpatrick, A linear algorithm for determining the separation of convex polyhedra, *J. Algorithms* 6 (1985), 381–392.

[7] D. Dobkin and D. Kirkpatrick, Determining the separation of preprocessed polyhedra — A unified approach, *Proc. 17th Internat. Colloq. Automata Lang. Program.*, 1990, 400–413.

[8] R. M. Dudley, Metric entropy of some classes of sets with differentiable boundaries, *J. Approx. Theory* 10 (1974), 227–236.

[9] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.

[10] J. Hershberger and S. Suri, Practical methods for approximating shortest paths on a convex polytope in $\mathbb{R}^3$, *Proc. 6th ACM-SIAM Symp. Discr. Algo.*, 1995, 447–456.

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

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

[13] J. Pach, personal communication.

[14] A. V. Pogorelov, *Extrinsic Geometry of Convex Surfaces*, Volume 35 of *Translations of Mathematical Monographs*, American Mathematical Society, Providence, RI, 1973.

[15] F.P. Preparata and M.I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, Berlin 1985.

[16] J. Reif and J. Storer, Shortest paths in Euclidean space with polyhedral obstacles, *J. Assoc. Comput. Mach.* 41 (1994), 1013–1019.

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

[18] M. Sharir and A. Schorr, On shortest paths in polyhedral spaces, *SIAM J. Comput.* 15 (1986), 193–215.

Appendix: Folding Angles of Shortest Paths Can Be Large

In this appendix we show that the folding angle of a shortest path on the boundary of a convex polytope in 3-space can be arbitrarily large. In fact, we establish a lower bound on the maximum possible folding angle, which is linear in the number of facets of the polytope. As already mentioned, the construction given here is a variant of a construction of János Pach (which has a somewhat weaker lower bound), and we are grateful to him for allowing us to include this construction in this paper.

For each integer n, define a polytope P_n as follows. P_n is the intersection of n tetrahedra, $T_1, \ldots, T_n$, each of which has one horizontal facet lying on the xy-plane, and its opposite vertex lies on the positive z-axis. The horizontal facet f_i of T_i is an equilateral triangle, centered at the origin, whose sides have orientations 0, $2\pi/3$ and $4\pi/3$, for even i, or $\pi/3$, π and $5\pi/3$, for odd i.

Let $\alpha(T_i)$ denote the dihedral angle between f_i and any of its other facets, and let $\beta(T_i)$ denote the angle between f_i and any of the non-horizontal edges of T_i. Note that $\tan\alpha(T_i) = 2\tan\beta(T_i)$.

We construct the tetrahedra T_i one after the other. Suppose we have already constructed $T_1, \ldots, T_n$, for $n \geq 1$ (the construction of T_1 will be detailed below). We assume inductively that the highest vertex of P_n is the top vertex of T_n (this clearly holds for $n = 1$).

Let h be a horizontal plane, such that the halfspace h^+ lying above h cuts P_n in a tetrahedron (which is a top portion of T_n). Let $g = T_n \cap h$, and let g' be the equilateral triangle that lies in h and has the vertices of g as the midpoints of its edges (g' is twice as large as g and is rotated by $\pi/3$). Construct T_{n+1} so that $\alpha(T_{n+1}) < \beta(T_n)$ and its non-horizontal facets pass through the edges of g'. The construction implies that the top vertex of T_{n+1} lies below the top vertex of T_n, so the inductive invariant concerning the top vertices of the P_j's continues to hold for $n+1$ too.

It is easy to verify, using induction on n, that the following properties are satisfied:

(i) Each non-horizontal facet of each T_i contributes a facet to P_n.

(ii) For each $1 < i < n$, the three non-horizontal facets of T_i form a 'band' around ∂P_n, which disconnects the non-horizontal facets of the T_j's with $j < i$ from the non-horizontal facets of the T_j's with $j > i$.

An illustration of P_3 is shown in Figure 4.

Now let P'_n be the reflected copy of P_n through the xy-plane, and let $Q_n = P_n \cup P'_n$. Let s be the top vertex of P_n and t be the bottom vertex of P'_n. Properties (i) and (ii) imply that *any* path from s to t along ∂Q_n must visit at least one facet of each T_i. Moreover, the path must make a transition from a facet of T_i to a facet of T_{i-1}, for each $i = 2, \ldots, n$.

We next define the angles of the tetrahedra T_i as follows. Let $\varepsilon > 0$ be an arbitrarily small number. Set $\beta_n = \frac{\pi}{2} - \varepsilon$, and let α_n be the angle that satisfies $\tan\alpha_n = 2\tan\beta_n$. Suppose α_j has already been defined. Then set $\beta_{j-1} = \left(\frac{\pi}{2} + \alpha_j\right)/2$, and define α_{j-1} by $\tan\alpha_{j-1} = 2\tan\beta_{j-1}$. Having defined this sequence of angles, we construct the T_i's so that $\alpha(T_i) = \alpha_i$.

By construction, the unit normals of the facets of T_i are

$$(\sin\alpha_i \cos 2j\pi/3,\ \sin\alpha_i \sin 2j\pi/3,\ \cos\alpha_i)\,, \quad \text{for } j = 0, 1, 2, \text{ if } i \text{ is even}$$

or

$$(\sin\alpha_i \cos(2j+1)\pi/3,\ \sin\alpha_i \sin(2j+1)\pi/3,\ \cos\alpha_i)\,, \quad \text{for } j = 0, 1, 2, \text{ if } i \text{ is odd.}$$

Hence, the smallest possible folding angle θ_i between a facet of T_{i-1} and a facet of T_i (which is equal to the angle between the normals to these facets) satisfies

$$\cos\theta_i = \frac{1}{2}\sin\alpha_{i-1}\sin\alpha_i + \cos\alpha_{i-1}\cos\alpha_i \leq \frac{1}{2} + \left(\frac{\pi}{2} - \alpha_{i-1}\right)\left(\frac{\pi}{2} - \alpha_i\right) \leq \frac{1}{2} + \varepsilon^2\,,$$

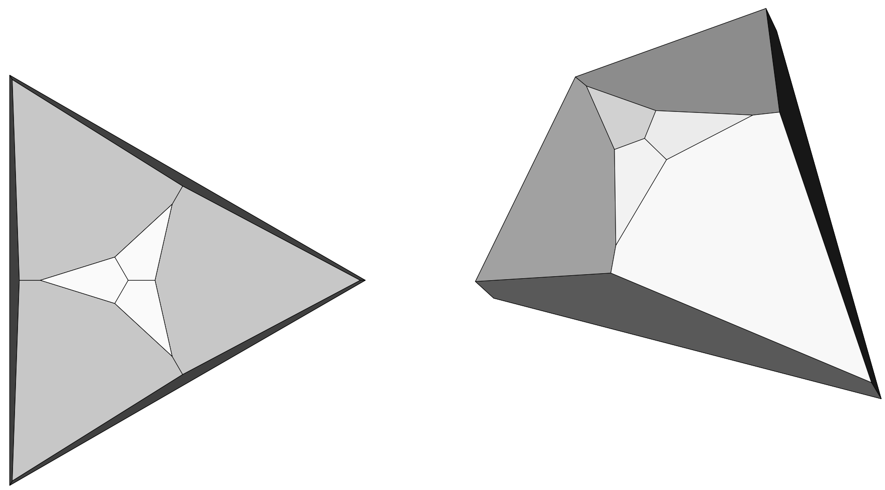

Figure 4: The polytope P_3: a top view and a side view

since $\frac{\pi}{2} - \varepsilon \leq \alpha_i \leq \pi/2$. That is, $\theta_i \geq \arccos\left(\frac{1}{2} + \varepsilon^2\right)$, and this lower bound can be made arbitrarily close to $\pi/3$, provided ε is chosen sufficiently small.

To sum up, we have shown that the folding angle of any path on ∂Q_n from s to t is lower-bounded by an angle close to $2(n-1)\pi/3$, which is linear in the number of facets of Q_n.

Remark. Pach's construction is originally presented in terms of the dual polytopes of (variants of) the P_n's. We felt, however, that it is more instructive to describe the actual polytopes explicitly.