Shortest Paths on Polyhedral Surfaces and Terrains*

Siu-Wing Cheng† Jiongxin Jin‡

Abstract

We present an algorithm to compute shortest paths on polyhedral surfaces under convex distance functions such that critical refractions between any geodesic and any surface edge can be avoided. Let n be the total number of vertices, edges and faces of the surface. Our algorithm can be used to compute L_1 and L_∞ shortest paths on a polyhedral surface in $O(n^2 \log^4 n)$ time. Given an $\varepsilon \in (0,1)$, our algorithm can find $(1+\varepsilon)$-approximate shortest paths on a terrain with gradient constraints and under a cost model that combines path length and total ascent. The running time is $O\left(\frac{1}{\sqrt{\varepsilon}} n^2 \log n + n^2 \log^4 n\right)$. This is the first efficient PTAS for such a general setting of terrain navigation.

1 Introduction

There are numerous applications that require path planning on terrains and polyhedral surfaces (e.g. [15, 17, 20, 21, 22, 29]). On a polyhedral surface of n vertices, edges and faces, Mitchell, Mount and Papadimitriou [23] presented an algorithm that runs in $O(n^2 \log n)$ time, which was subsequently improved by Chen and Han [9] to $O(n^2)$. Varadarajan and Agarwal [32] proposed two approximation algorithms that run in subquadratic time: $7(1+\varepsilon)$- and $15(1+\varepsilon)$-approximate shortest paths can be found in $O(n^{5/3} \log^{5/3} n)$ and $O(n^{8/5} \log^{8/3} n)$ time, respectively. Schreiber and Sharir [28] presented an $O(n \log n)$-time algorithm for convex polyhedral surfaces.

Terrain navigation has been studied from the perspectives of minimizing energy in robotics (e.g. [13, 26, 31]) or avoiding steep paths in spatial database and GIS (e.g. [22, 33]), but either no complexity result is given or the complexities of the algorithms given depend on the terrain geometry. De Berg and van Kreveld [6] pioneered the study of some height constrained path query problems on terrains, and posed the handling of additional constraints as open problems. The special case of finding a shortest descending path (SDP) has received attention lately. Ahmed et al. [3] presented two algorithms to construct a $(1+\varepsilon)$-approximate SDP. The running times are $O\left(\frac{n^2 L}{\varepsilon \ell \cos\phi} \log \frac{nL}{\varepsilon \ell \cos\phi}\right)$ and $O\left(\frac{n^2 L}{\varepsilon \ell} \log^2 \frac{nL}{\varepsilon \ell}\right)$, where L is the largest edge length in the terrain, ℓ is the smallest distance of a vertex from a non-incident edge in the same terrain face, and ϕ is the largest acute angle between a non-horizontal edge and a vertical line. Other related results can be found in [3, 4, 5, 27]. Recently, we developed a $(1+\varepsilon)$-approximate SDP algorithm that runs in $O(n^4 \log(n/\varepsilon))$ time [10], which is the first bound that is polynomial in n and $\log(1/\varepsilon)$ and independent of the terrain geometry. It seems hard to compute the exact SDP and no such algorithm is known so far.

This paper presents an algorithm for a constrained shortest path problem on a polyhedral surface with a special property. We call this problem the PolyPath problem. Each surface triangle f is associated with a convex polygon H_f that induces a convex distance function d_f. The length of a subpath in f is measured using d_f. The constraint is an input upper bound on the number of links in the output path. The special property is that any critical refraction

*Research supported by the Research Grant Council, Hong Kong, China (project no. 611812).

†Hong Kong University of Science and Technology. Email: `scheng@cse.ust.hk`

‡Google Inc. Email: `jamesjjx@google.com`

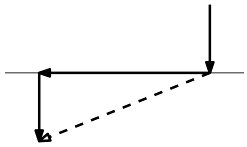

Figure 1: Assume that we are in the L_1 metric. The bold path makes critical refractions at the horizontal edge, which can be avoided using the dashed link without increasing the path cost.

between any geodesic and any surface edge can be avoided. That is, for every geodesic P, there is another geodesic Q with the same endpoints as P such that Q is not more costly than P, Q passes through the same sequence of surface edges as P, and each of these edges intersects Q transversally.[1] Figure 1 gives an example. Given two vertices s and t and a positive integer m, our algorithm returns the shortest path from s to t with m links or less in $O(hmn \log mn + mn \log^2 m \log^2 hm)$ time, where h is the maximum size of the convex polygons associated with the surface triangles. An immediate corollary is that an L_1 or L_∞ shortest path on a polyhedral surface can be computed in $O(n^2 \log^4 n)$ time.[2]

On terrains, we can optimize $c_1 \cdot$ Euclidean path length $+ c_2 \cdot$ total ascent with an ε relative error under gradient constraints for every positive constant c_1 and every non-negative constant c_2. The total ascent is the total increase in heights of all ascending subpaths, which measures the energy spent in increasing the potential energy. The weighted sum of the path length and its total ascent gives rise to a convex distance function, which can be approximately induced by a convex polygon of $O(1/\sqrt{\varepsilon})$ size. This allows us to reduce the problem to an instance of PolyPath such that $m = O(n)$ and $h = O(1/\sqrt{\varepsilon})$. Gradient constraints are specified by the maximum ascent and descent gradients allowed in $\mathcal{T}$. (The ascent and descent gradient bounds may be different, and they apply to all surface triangles.) This only changes the convex distance function slightly. Section 4 describes these reductions. In all, our algorithm can return a $(1+\varepsilon)$-approximate shortest path in $O\left(\frac{1}{\sqrt{\varepsilon}} n^2 \log n + n^2 \log^4 n\right)$ time, which makes it the first PTAS for such a general setting of terrain navigation. Our result implies that a $(1+\varepsilon)$-approximate SDP can be computed in $O\left(\frac{1}{\sqrt{\varepsilon}} n^2 \log n + n^2 \log^4 n\right)$ time.

There are several key ideas in solving the PolyPath problem. First, we discover how a geodesic bends at a surface edge. In fact, there may be multiple ways for a geodesic to bend, and we focus on a special type of geodesics in order to characterize the bending (Lemma 3.3). Second, we show that a geodesic for an edge sequence σ is completely characterized by the angles of incidence and exit at the edges in σ (Lemmas 3.5–3.7). A geodesic is thus preserved under sliding[3], which enables us to construct one geodesic for some edge sequence and generate from it other geodesics with the same edge sequence. This makes it possible to design a combinatorial algorithm that composes a shortest path by combining shorter geodesics. Lastly, we design an efficient hierarchical scheme for combining geodesics (Section 3.2).

2 Preliminaries

Let $\mathcal{T}$ denote the input polyhedral surface with n vertices, edges and faces. W.l.o.g., assume that each face of $\mathcal{T}$ is a triangle, and the source s and the destination t are vertices of $\mathcal{T}$. Each face f of $\mathcal{T}$ is associated with a convex polygon H_f, which contains the origin, lies in a plane parallel

[1] This property is not enjoyed by all convex distance functions because critical refractions are sometimes unavoidable in shortest paths in the general case [11, 24].

[2] Not much is known about L_1 and L_∞ shortest paths on polyhedral surfaces. We presented an L_1 shortest path algorithm in [10] that runs in $O(n^3 \log n)$ time.

[3] If critical refractions are allowed, it is difficult to deal with sliding a geodesic because one can slide the subpaths before and after a critical refraction by different amounts.

to f, and induces the distance function d_f. We allow the origin to be on the boundary of H_f. The cost of the directed segment $pq \subset f$ is $\text{cost}(pq) = d_f(p, q) = \inf\left\{\lambda > 0 : \frac{1}{\lambda}(q - p) \in H_f\right\}$, which can be computed in $O(\log |H_f|)$ time by binary search.

We use $\vec{u}$ to denote a vector and $\overset{\rightharpoonup}{u}$ to denote the unit vector in the same direction as $\vec{u}$. Given $\vec{u}$ and $\vec{v}$, $\theta(\vec{u}, \vec{v})$ denotes the angle measured from $\vec{u}$ to $\vec{v}$ in counter-clockwise order, which takes value in $[0, 2\pi)$. The inner product of $\vec{u}$ and $\vec{v}$ is denoted by $\langle \vec{u}, \vec{v} \rangle$.

All polygonal paths in this paper are oriented from their sources to their destinations. A *link* of a polygonal path is a maximal segment in a face or on an edge of $\mathcal{T}$, and its endpoints are called *nodes*. We assume that every node is either a vertex or a point in the interior of an edge because a node in the interior of a face can be removed by shortcutting without increasing the path cost. By the requirement of the PolyPath problem on the convex distance functions, we can further assume that every node in the interior of an edge is a *transversal* node, that is, its two incident links lie in the interiors of two distinct faces.

Let p_i, $i \in [0, k]$, be the nodes in order along a path P. Let $\vec{v}_i = p_i - p_{i-1}$ for $i \in [1, k]$. The *direction vector* of P is $(\overset{\rightharpoonup}{v}_1, \ldots, \overset{\rightharpoonup}{v}_k)$. We can specify P as $(p_0, p_1, \ldots, p_k)$ or as $(p_0, (\overset{\rightharpoonup}{v}_1, \ldots, \overset{\rightharpoonup}{v}_k))$. The subpath of P from a point x to another point y is denoted by $P[x, y]$. The cost of P is $\text{cost}(P) = \sum_{\text{face } f} \text{cost}(P \cap f)$. $\|P\|$ denotes the Euclidean length of P.

The edges that P crosses in order is the *edge sequence* of P. It includes the edge containing P's destination but not the edge containing P's source. A path may have multiple edge sequences if its interior passes though a vertex. For example, suppose that the edges $e_1, e_2, \ldots, e_k$ are incident to a vertex ν in circular order. If a path moves from the face bounded by e_1 and e_k to ν onward to the face bounded by e_i and e_{i+1}, then one edge sequence contains the substring $e_1 e_2 \ldots e_i$, and another edge sequence contains the substring $e_k e_{k-1} \ldots e_{i+1}$.

A path P is a *geodesic* if it has the minimum cost among all paths with the same source, destination, and edge sequence as P. The shortest path from s to t is the shortest geodesic from s to t over all possible edge sequences.

3 Solving PolyPath

We first characterize the geodesics by their direction vectors in Section 3.1. Then we propose an algorithm in Section 3.2 to solve the PolyPath problem.

3.1 Properties of geodesics

Let $\sigma = (e_1, e_2, \ldots, e_k)$ be the edge sequence of some geodesic that starts from a point p_0 on the boundary of some face in $\mathcal{T}$ and ends at a point p_k on the boundary of another face in $\mathcal{T}$. Thus, e_i and e_{i+1} are distinct edges of the same face, and e_i and e_{i+2} do not bound the same face. Let e_0 denote an edge adjacent to e_1 that contains the source of the geodesic. For $i \in [1, k]$, let f_i denote the face bound by e_{i-1} and e_i.

For $i \in [1, k]$, we define the positive and negative sides of a point on e_i as follows. Orient e_i to obtain a directed segment $a_i b_i$ so that f_i and f_{i+1} are on the left and right of $a_i b_i$, respectively. Let $\vec{e}_i$ denote the vector $b_i - a_i$. Given two points $p, q \in e_i$, we say that q lies on the *positive* or *negative* side of p if $\langle q - p, \vec{e}_i \rangle > 0$ or $\langle q - p, \vec{e}_i \rangle < 0$, respectively. The head and tail of the oriented e_i are the *positive and negative endpoints* of e_i, respectively.

There may be multiple geodesics that start from p_0, end at p_k, and share the same edge sequence σ. Let $P = (p_0, (\overset{\rightharpoonup}{v}_1, \ldots, \overset{\rightharpoonup}{v}_k))$ and let $Q = (p_0, (\overset{\rightharpoonup}{w}_1, \ldots, \overset{\rightharpoonup}{w}_k))$ be two such geodesics. We say that $\overset{\rightharpoonup}{v}_i$ is *smaller* than $\overset{\rightharpoonup}{w}_i$ if $\theta(\vec{e}_i, \overset{\rightharpoonup}{v}_i) < \theta(\vec{e}_i, \overset{\rightharpoonup}{w}_i)$. The *canonical geodesic* from p_0 and p_k with edge sequence σ is the geodesic that has the lexicographically smallest direction vector. Intuitively, the canonical geodesic hits every oriented e_i at a point closest to the negative endpoint of e_i. This gives rise to the next lemma (proof in Appendix A.1).

Lemma 3.1. *Let $P = (p_0, p_1, \ldots, p_k)$ and $Q = (q_0 = p_0, q_1, \ldots, q_k = p_k)$ be two geodesics from p_0 to p_k with the same edge sequence. If P is a canonical geodesic, then for $i \in [1, k-1]$, q_i does not lie on the negative side of p_i.*

We will characterize a canonical geodesic via the derivative of its cost, which may not change smoothly as its destination moves. Thus, we define the derivative using limit and it depends on how the limit is approached. Recall that $\sigma = (e_1, \ldots, e_k)$ and e_0 is an edge adjacent to e_1 containing the source p_0 of P. Let $\sigma_{ij} = (e_{i+1}, \ldots, e_j)$. For every point $p \in e_i$, define a function $C_{p,\sigma_{ij}}(x)$ to be the cost of a geodesic with edge sequence σ_{ij} from p to a point $x \in e_j$. For every point $q \in e_j$, define the function $D_{q,\sigma_{ij}}(x)$ be the cost of a geodesic with edge sequence σ_{ij} from a point $x \in e_i$ to q. We use $x' \to x^+$ and $x' \to x^-$ to denote x' approaching x from the positive and negative sides of x, respectively. Define: $\partial C^+_{p,\sigma_{ij}}(x) = \lim_{x' \to x^+} \frac{C_{p,\sigma_{ij}}(x') - C_{p,\sigma_{ij}}(x)}{\|xx'\|}$, $\partial C^-_{p,\sigma_{i,j}}(x) = \lim_{x' \to x^-} \frac{C_{\sigma_{ij}}(p,x) - C_{p,\sigma_{ij}}(x')}{\|xx'\|}$, $\partial D^+_{q,\sigma_{ij}}(x) = \lim_{x' \to x^+} \frac{D_{q,\sigma_{ij}}(x') - D_{q,\sigma_{ij}}(x)}{\|xx'\|}$, and $\partial D^-_{q,\sigma_{ij}}(x) = \lim_{x' \to x^-} \frac{D_{q,\sigma_{ij}}(x) - D_{q,\sigma_{ij}}(x')}{\|xx'\|}$.

Lemma 3.2. *$C_{p,\sigma_{ij}}(x)$ and $D_{q,\sigma_{ij}}(x)$ are convex piecewise linear function in x. If y is on the positive side of x in e_j, then $\partial C^+_{p,\sigma_{ij}}(y) \geq \partial C^-_{p,\sigma_{ij}}(y) \geq \partial C^+_{p,\sigma_{ij}}(x) \geq \partial C^-_{p,\sigma_{ij}}(x)$. If y is on the positive side of x in e_i, then $\partial D^+_{q,\sigma_{ij}}(y) \geq \partial D^-_{q,\sigma_{ij}}(y) \geq \partial D^+_{q,\sigma_{ij}}(x) \geq \partial D^-_{q,\sigma_{ij}}(x)$.*

Proof. The function $\text{cost}(x_{\ell-1}x_\ell)$ in $x_{\ell-1}$ and x_ℓ is convex, piecewise linear. Since $C_{p,\sigma_{ij}}(x) = \min\{\sum_{\ell=i+1}^{j} \text{cost}(x_{\ell-1}x_\ell) : x_i = p,\ x_j = x,\ x_\ell \in e_\ell \text{ for } \ell \in (i,j)\}$, $C_{p,\sigma_{ij}}(x)$ is the minimization of the cross-section of a convex, piecewise linear function. This implies the properties of $C_{p,\sigma_{ij}}$, $\partial C^+_{p,\sigma_{ij}}$ and $\partial C^-_{p,\sigma_{ij}}$ stated in the lemma. The same argument works for $D_{q,\sigma_{ij}}(x)$. ∎

Our algorithm will form new geodesics by concatenating shorter ones. It is clear that if we split a canonical geodesic $(p_0, \ldots, p_k)$ at p_i, we obtain two shorter canonical geodesics. Lemma 3.3 below shows that the converse is true under some conditions. Its proof is based on the fact that p_i cannot be perturbed to reduce the cost (Appendix A.2).

Lemma 3.3. *If a path $P = (p_0, p_1, \ldots, p_k)$ is a canonical geodesic with edge sequence σ, where $p_i \in e_i$, then the following conditions hold for every $i \in [1, k-1]$.*

(i) *$P[p_0, p_i]$ and $P[p_i, p_k]$ are canonical geodesics with edge sequences σ_{0i} and σ_{ik}, respectively.*

(ii) *p_i is the positive endpoint of e_i or $\partial C^+_{p_0,\sigma_{0i}}(p_i) + \partial D^+_{p_k,\sigma_{ik}}(p_i) \geq 0$.*

(iii) *p_i is the negative endpoint of e_i or $\partial C^-_{p_0,\sigma_{0i}}(p_i) + \partial D^-_{p_k,\sigma_{ik}}(p_i) < 0$.*

Conversely, if the conditions above hold for some $i \in [1, k-1]$, then P is a canonical geodesic from p_0 to p_k with edge sequence σ.

By Lemmas 3.2 and 3.3, once two canonical geodesics diverge, they will never cross.

Lemma 3.4. *Let $P = (p_0, \ldots, p_k)$ and $Q = (q_0, \ldots, q_k)$ be two canonical geodesics with edge sequence σ such that p_i and q_i lie in the interior of e_i for $i \in [1, k-1]$. If $\theta(p_i - p_{i-1}, \vec{e}_i) > \theta(q_i - q_{i-1}, \vec{e}_i)$ for some $i \in [1, k-1]$, then $\theta(p_j - p_{j-1}, \vec{e}_j) \geq \theta(q_j - q_{j-1}, \vec{e}_j)$ for all $j > i$.*

Lemma 3.5 below shows that when we slide a geodesic, the path cost changes linearly.

Lemma 3.5. *Let $P = (p_0, \ldots, p_k)$ be a geodesic with edge sequence σ, where p_i lies in the interior of e_i for $i \in [1, k]$. Let $Q = (q_0, \ldots, q_k)$ be another path such that $q_i \in e_i$ for $i \in [0, k]$ and $q_{i-1}q_i$ is parallel to $p_{i-1}p_i$ for $i \in [1, k]$. For $i, j \in [0, k]$ such that $i < j$, define δ_{ij} and γ_{ij} by the relations $\|p_iq_i\| = \delta_{ij} \cdot \|p_jq_j\|$ and $\text{cost}(Q[q_i, q_j]) = \text{cost}(P[p_i, p_j]) + \gamma_{ij} \cdot \langle q_j - p_j, \vec{e}_j \rangle$. Then, δ_{ij} and γ_{ij} depend on the direction vector of $P[p_i, p_j]$ only, $\delta_{i-1,i}$ and $\gamma_{i-1,i}$ can be computed in $O(1)$ time, and for all $\ell \in [i+1, j-1]$, $\delta_{ij} = \delta_{i\ell}\delta_{\ell j}$ and $\gamma_{ij} = \delta_{\ell j}\gamma_{i\ell} + \gamma_{\ell j}$.*

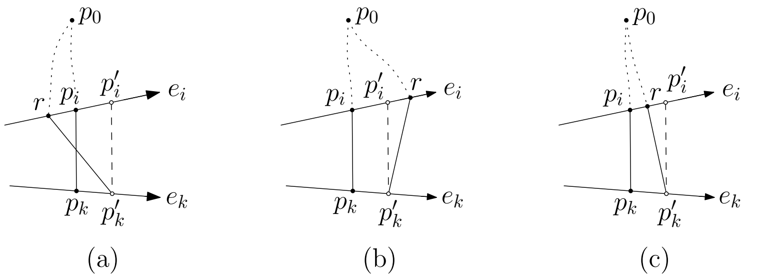

Figure 2: Three cases depending on the position of r relative to p_i and p'_i.

Proof. Let $\overrightarrow{v}_i$ be the direction of $p_{i-1}p_i$. By the sine law, $\delta_{i-1,i} = \sin(\theta(\overrightarrow{v}_i, \overrightarrow{e}_{i-1}))/\sin(\theta(\overrightarrow{v}_i, \overrightarrow{e}_i))$. The edges e_{i-1} and e_i share a negative endpoint a or a positive endpoint b. In the first case, $\|q_{i-1}q_i\| = \sin(\theta(\overrightarrow{e}_i, \overrightarrow{e}_{i-1})) \cdot \|aq_i\|/\sin(\theta(\overrightarrow{v}_i, \overrightarrow{e}_{i-1}))$. In the second case, $\|q_{i-1}q_i\| = \sin(\theta(\overrightarrow{e}_{i-1}, \overrightarrow{e}_i)) \cdot \|q_ib\|/\sin(\theta(\overrightarrow{v}_i, \overrightarrow{e}_{i-1}))$. Thus, $\gamma_{i-1,i} = c_i \cdot \sin(\theta(\overrightarrow{e}_i, \overrightarrow{e}_{i-1}))/\sin(\theta(\overrightarrow{v}_i, \overrightarrow{e}_{i-1}))$, where c_i is the cost of a unit segment with direction $\overrightarrow{v}_i$ in the face bound by e_{i-1} and e_i. So $\delta_{i-1,i}$ and $\gamma_{i-1,i}$ depend on $\overrightarrow{v}_i$ only. Assume that $i < j-1$. For all $\ell \in (i,j)$, $\|p_iq_i\| = \delta_{i\ell} \cdot \|p_\ell q_\ell\| = \delta_{i\ell}\delta_{\ell j} \cdot \|p_jq_j\|$, and $\mathrm{cost}(Q[q_i,q_j]) = \mathrm{cost}(Q[q_i,q_\ell]) + \mathrm{cost}(Q[q_\ell,q_j]) = \mathrm{cost}(P[p_i,p_\ell]) + \gamma_{i\ell} \cdot \langle q_\ell - p_\ell, \overrightarrow{e}_\ell \rangle + \mathrm{cost}(P[p_\ell,p_j]) + \gamma_{\ell j} \cdot \langle q_j - p_j, \overrightarrow{e}_j \rangle = \mathrm{cost}(P[p_i,p_j]) + (\delta_{\ell j}\gamma_{i\ell} + \gamma_{\ell j}) \cdot \langle q_j - p_j, \overrightarrow{e}_j \rangle$. So $\delta_{ij} = \delta_{i\ell}\delta_{\ell j}$ and $\gamma_{ij} = \delta_{\ell j}\gamma_{i\ell} + \gamma_{\ell j}$. Inductively, δ_{ij} and γ_{ij} depend on the direction vector of $P[i,j]$ only. ◻

We want to show that $\partial C^+_{p_0,\sigma_{0k}}$, $\partial C^-_{p_0,\sigma_{0k}}$, $\partial D^+_{p_k,\sigma_{0k}}$, and $\partial D^-_{p_k,\sigma_{0k}}$ depend on the direction vector only, i.e., not on the location of p_0 and p_k. Then, Lemmas 3.3 and 3.5 allow us to form canonical geodesics by sliding and concatenating shorter ones. The first step is to prove a conditional version of this result.

Lemma 3.6. *Let $P = (p_0, \ldots, p_k)$ be a geodesic with edge sequence σ, where p_i lies in the interior of e_i for $i \in [1,k]$. Define δ_{ij} and γ_{ij} as in Lemma 3.5. If there exists $i \in [1, k-1]$ such that $\partial C^+_{p_0,\sigma_{0i}}$ and $\partial C^-_{p_0,\sigma_{0i}}$ depend only on the director vector of $P[p_0,p_i]$ and $\partial D^+_{p_k,\sigma_{ik}}$ and $\partial D^-_{p_k,\sigma_{ik}}$ depend only on the direction vector of $P[p_i,p_k]$, then:* (i) $\partial C^+_{p_0,\sigma_{0k}}(p_k) = \min\{\partial C^+_{p_i,\sigma_{ik}}(p_k),\ \delta_{ik} \cdot \partial C^+_{p_0,\sigma_{0i}}(p_i) + \gamma_{ik}\}$ *and* $\partial C^-_{p_0,\sigma_{0k}}(p_k) = \min\{\partial C^-_{p_i,\sigma_{ik}}(p_k),\ \delta_{ik} \cdot \partial C^-_{p_0,\sigma_{0i}}(p_i) + \gamma_{ik}\}$*, and* (ii) $\partial D^+_{p_k,\sigma_{0k}}(p_0) = \min\left\{\partial D^+_{p_i,\sigma_{0i}}(p_0),\ \frac{1}{\delta_{0i}}\partial D^+_{p_k,\sigma_{ik}}(p_i) + \frac{\gamma_{0i}}{\delta_{0i}}\right\}$ *and* $\partial D^-_{p_k,\sigma_{0k}}(p_0) = \min\left\{\partial D^-_{p_i,\sigma_{0i}}(p_0),\ \frac{1}{\delta_{0i}}\partial D^-_{p_0,\sigma_{0i}}(p_i) + \frac{\gamma_{0i}}{\delta_{0i}}\right\}$.

Proof. (Sketch) Consider the derivation of $\partial C^+_{p_0,\sigma_{0k}}(p_k)$ in (i). The derivation of $\partial C^-_{p_0,\sigma_{0k}}(p_k)$ is symmetric. Take a point $p'_k \in e_k$ on the positive side of p_k and arbitrarily close to p_k. For $j \in [i, k-1]$, let p'_j be the point in e_j such that $p'_jp'_{j+1}$ is parallel to p_jp_{j+1}. Since p'_k is arbitrarily close to p_k, p'_i is also arbitrarily close to p_i. Therefore,

$$\begin{aligned}
C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p'_k) &= C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p_k) + \partial C^+_{p_i,\sigma_{ik}}(p_k) \cdot \|p_kp'_k\| \\
&= C_{p_0,\sigma_{0k}}(p_k) + \partial C^+_{p_i,\sigma_{ik}}(p_k) \cdot \|p_kp'_k\|, \\
C_{p_0,\sigma_{0i}}(p'_i) + C_{p'_i,\sigma_{ik}}(p'_k) &= C_{p_0,\sigma_{0i}}(p_i) + \partial C^+_{p_0,\sigma_{0i}}(p_i) \cdot \|p_ip'_i\| + C_{p_i,\sigma_{ik}}(p_k) + \gamma_{ik} \cdot \|p_kp'_k\| \\
&= C_{p_0,\sigma_{0k}}(p_k) + \left(\delta_{ik} \cdot \partial C^+_{p_0,\sigma_{0i}}(p_i) + \gamma_{ik}\right) \cdot \|p_kp'_k\|.
\end{aligned}$$

Let Q be a geodesic from p_0 to p'_k with edge sequence $\sigma_{0k} = \sigma$. Let r be the node of Q on e_i. We are done if we can show that $C_{p_0,\sigma_{0k}}(p'_k)$ is equal to $C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p'_k)$ or $C_{p_0,\sigma_{0i}}(p'_i) + C_{p'_i,\sigma_{ik}}(p'_k)$. There are three cases in Figure 2 depending on the position of r. We discuss one case here and leave the rest to Appendix A.3. Suppose that $r \in p_ip'_i$ (Figure 2(c)). Since $\|p_ir\|$ and $\|rp'_i\|$ are arbitrarily small, $C_{p_0,\sigma_{0k}}(p'_k) = C_{p_0,\sigma_{0i}}(r) + D_{p'_k,\sigma_{ik}}(r) = C_{p_0,\sigma_{0i}}(p_i) + \partial C^+_{p_0,\sigma_{0i}}(p_i) \cdot \|p_ir\| + D_{p'_k,\sigma_{ik}}(p'_i) - \partial D^-_{p'_k,\sigma_{ik}}(p'_i) \cdot \|p'_ir\|$, which is a linear function in $\|p_ir\|$ by our assumption that $\partial C^+_{p_0,\sigma_{0i}}$ and $\partial D^-_{p_k,\sigma_{ik}}$ depend only on the direction vectors of $P[p_0,p_i]$ and

$P[i,k]$ respectively (hence $\partial D^-_{p'_k,\sigma_{ik}}(p'_i) = \partial D^-_{p_k,\sigma_{ik}}(p_i)$). Thus, the last expression is minimized when $r = p_i$ or $r = p'_i$. Hence, $C_{p_0,\sigma_{0k}}(p'_k) = C_{p_0,\sigma_{0i}}(p_i) + D_{p'_k,\sigma_{ik}}(p_i) = C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p'_k)$ or $C_{p_0,\sigma_{0k}}(p'_k) = C_{p_0,\sigma_{0i}}(p'_i) + D_{p'_k,\sigma_{ik}}(p'_i) = C_{p_0,\sigma_{0i}}(p'_i) + C_{p'_i,\sigma_{ik}}(p'_k)$.

The correctness of (ii) can be proved in a similar way. ◻

Lemma 3.6 clearly lends itself to an inductive proof to establish the same result unconditionally, as stated in Lemma 3.7 below. The proof is in Appendix A.4.

Lemma 3.7. *Let $P = (p_0, \ldots, p_k)$ be a geodesic path with edge sequence σ, where p_i lies in the interior of e_i for $i \in [1,k]$. Let δ_{ij} and γ_{ij} be defined as in Lemma 3.5. Then $\partial C^+_{p_0,\sigma_{0k}}$, $\partial C^-_{p_0,\sigma_{0k}}$, $\partial D^+_{p_k,\sigma_{0k}}$, and $\partial D^-_{p_k,\sigma_{0k}}$ depend only on the direction vector of P. Moreover, the formulae in Lemma 3.6 hold for all $i \in [1,k-1]$.*

3.2 Algorithm

Chen and Han introduced the *sequence tree* to capture the edge sequences of geodesics on a polyhedral surface in the L_2 case [9]. The tree is grown until the number of tree levels meets the input upper bound on the number of links allowed in the solution path. The best path discovered from s to t is the shortest path desired. Constructing a new tree node involves finding a new shortest path with a particular edge sequence. The key is to use the structural properties in the last subsection to carry out this step and do it fast.

A sequence tree node α is a *vertex-node* or an *edge-node* which represents a vertex, denoted ν_α, or an edge of $\mathcal{T}$, denoted e_α. A *face corner* (f,ν) is the corner at a vertex ν of a face f. An edge-node α *annexes* a face corner (f,ν) if e_α is the edge of f opposite ν and the parent of α does not correspond to another vertex or edge of f. (Since e_α is opposite two face corners, the second condition ensures that α annexes the face corner just included by the growing tree.)

The root corresponds to the source s. The edge-nodes on the tree path from the root to α correspond to an edge sequence, denoted σ_α. Let α_0 be the nearest ancestor vertex-node of α. The edge-nodes on the tree path from α_0 to α correspond to a suffix of σ_α, denoted $\tilde{\sigma}_\alpha$. The edge sequences σ_α and $\tilde{\sigma}_\alpha$ are used in the analysis, but they are not stored at α. In the case that α is a vertex-node, P_α denotes the canonical geodesic from s to ν_α that passes through the edges in σ_α. We compute $\mathrm{cost}(P_\alpha)$ and stores it at α, but P_α is used in the analysis only.

The sequence tree is grown in a breadth-first manner until the number of tree levels meets the input upper bound m. When an edge-node annexing a face corner (f,ν) is expanded, it gains at most one vertex-node corresponding to ν and two edge-nodes corresponding to the edges of f incident to ν. When a vertex-node α is expanded, it gains at most one vertex-node for each vertex adjacent to ν_α and one edge-node for each edge opposite ν_α. Multiple nodes may correspond to the same edge or vertex. To control the tree size, Chan and Han introduced the *one-corner one-split property*: *at any time, at most one vertex-node corresponding to the same vertex is allowed to have any child node; at most one edge-node annexing the same face corner is allowed to have two child edge-nodes.* This property ensures that at most $O(n)$ tree nodes are ever created at each level [9, Theorem 8].

A notion of *dominance* is need to maintain the one-corner one-split property. Let α and β be two vertex-nodes corresponding to the same vertex ν or two edge-nodes annexing the same face corner (f,ν). Let α_0 and β_0 be the nearest ancestor vertex-nodes of α and β, respectively. Let P and Q be the canonical geodesics from ν_{α_0} and ν_{β_0} to ν that pass through the edges in $\tilde{\sigma}_\alpha$ and $\tilde{\sigma}_\beta$, respectively. We say that α *dominates* β if $\mathrm{cost}(P_{\alpha_0}) + \mathrm{cost}(P) < \mathrm{cost}(P_{\beta_0}) + \mathrm{cost}(Q)$, or $\mathrm{cost}(P_{\alpha_0}) + \mathrm{cost}(P) = \mathrm{cost}(P_{\beta_0}) + \mathrm{cost}(Q)$ but α is expanded before β in growing the tree. Assume that α dominates β. Suppose they are vertex-nodes. If β has been expanded, we remove all tree nodes descending from it; otherwise, we will not expand β. Suppose that α and β are edge-nodes. There is an edge e incident to ν such that every geodesic from ν_{β_0} to e through the edges in $\tilde{\sigma}_\beta$ crosses P. If β has been expanded, we prune the child node of β corresponding to e; otherwise, when we expand β, we will not generate a child node corresponding to e.

After we construct a new leaf α of the sequence tree, it takes $O(\log mn)$ amortized time to test the dominance and prune the tree, modulo the time to compute the costs of geodesics: $\text{cost}(P_\alpha)$ if α is a vertex-node, or the costs of geodesics from ν_{α_0} to e_α with edge sequence $\tilde{\sigma}_\alpha$ if α is an edge-node. Appendix B.1 describes the dominance testing and pruning. We give a proof in [10, Lemma 3.1] that the pruning preserves a shortest path from s to t in the L_∞ metric, which is also applicable here. The correctness of our algorithm thus follows. In the rest of this subsection, we describe how to compute the costs of geodesics when constructing a new leaf.

Edge-node creation. Let α be a new edge-node created at tree level ℓ. Let α_0 be the nearest ancestor vertex-node of α at tree level $k < \ell$. Let $(e_k, \ldots, e_\ell)$ be the edges corresponding to the edge-nodes on the tree path from α_0 to α, with e_ℓ denoting e_α and e_k denoting an edge incident to ν_{α_0}. We do some processing at α to aid the future growth of the subtree rooted at α.

For all $i \geq 0$ such that $k \leq \ell - 2^i$ and 2^i divides ℓ, we compute a data structure $\mathcal{L}_\alpha^i$ to represent the canonical geodesics from any point in $e_{\ell-2^i}$ to some point in e_ℓ, which can be represented by their direction vectors by Lemmas 3.3 and 3.7. The insight is that only some critical direction vectors matter, and the rest can be linearly interpolated from them.

Let $I_{\alpha,\mathbf{v}} \subseteq e_{\ell-2^i}$ be the interval of origins of canonical geodesics that reach e_ℓ with direction vector $\mathbf{v}$ and edge sequence $(e_{\ell-2^i+1}, \ldots, e_\ell)$.[4] Let $A_{\alpha,\mathbf{v}} : I_{\alpha,\mathbf{v}} \to \mathbb{R}$ and $a_{\alpha,\mathbf{v}} : I_{\alpha,\mathbf{v}} \to e_\ell$ be functions such that $A_{\alpha,\mathbf{v}}(p)$ is the cost of the canonical geodesic from p to e_ℓ with direction vector $\mathbf{v}$ and edge sequence $(e_{\ell-2^i+1}, \ldots, e_\ell)$, and $a_{\alpha,\mathbf{v}}(p)$ is the destination of this geodesic. Let $B_{\alpha,\mathbf{v}} : a_{\alpha,\mathbf{v}}[I_{\alpha,\mathbf{v}}] \to \mathbb{R}$ and $b_{\alpha,\mathbf{v}} : a_{\alpha,\mathbf{v}}[I_{\alpha,\mathbf{v}}] \to e_{\ell-2^i}$ be functions such that $B_{\alpha,\mathbf{v}}(q)$ is the cost of the canonical geodesic from $e_{\ell-2^i}$ to q with direction vector $\mathbf{v}$ and edge sequence $(e_{\ell-2^i+1}, \ldots, e_\ell)$, and $b_{\alpha,\mathbf{v}}(q)$ is the source of this geodesic. These four functions are affine and they can be stored in $O(1)$ space and evaluated in $O(1)$ time. The direction vectors in $\mathcal{L}_\alpha^i$ are stored in lexicographic order.[5] The following properties are enforced on $\mathcal{L}_\alpha^i$:

- P1: Each direction vector in $\mathcal{L}_\alpha^i$ is that of some canonical geodesic from $e_{\ell-2^i}$ to e_ℓ.
- P2: Any two adjacent direction vectors differ in exactly one entry. These two different directions point to the same edge of the convex polygon defining the distance function for the corresponding face.
- P3: Let $\mathbf{v}$ and $\mathbf{w}$ be two adjacent direction vectors. For any $p \in I_{\alpha,\mathbf{v}} \cap I_{\alpha,\mathbf{w}}$ and any $t \in [0,1]$, the cost of a geodesic from p to $t\,a_{\alpha,\mathbf{v}}(p) + (1-t)a_{\alpha,\mathbf{w}}(p)$ is $tA_{\alpha,\mathbf{v}}(p) + (1-t)A_{\alpha,\mathbf{w}}(p)$.[6]
- P4: Let $\mathbf{v}$ and $\mathbf{w}$ be two adjacent direction vectors. For any $q \in a_{\alpha,\mathbf{v}}[I_{\alpha,\mathbf{v}}] \cap a_{\alpha,\mathbf{w}}[I_{\alpha,\mathbf{w}}]$ and any $t \in [0,1]$, the cost of the geodesic from $t\,b_{\alpha,\mathbf{v}}(q) + (1-t)b_{\alpha,\mathbf{w}}(q)$ to q is $tB_{\alpha,\mathbf{v}}(q) + (1-t)B_{\alpha,\mathbf{w}}(q)$.

The first direction vector in $\mathcal{L}_\alpha^i$ is stored in its full form. For any other direction vector, we only store its first link, last link, and the difference from its predecessor in $\mathcal{L}_\alpha^i$. By P2, the storage required by $\mathcal{L}_\alpha^i$ is $O(2^i)$ plus the number of direction vectors in the list.

The construction of $\mathcal{L}_\alpha^i$ proceeds in increasing i. The base case is $\mathcal{L}_\alpha^0$. Let H_ℓ denote the convex polygon that induces the distance function for the face bounded by $e_{\ell-1}$ and e_ℓ. $\mathcal{L}_\alpha^0$

[4]Let P be the canonical geodesic from $I_{\alpha,\mathbf{v}}$ to e_ℓ with direction vector $\mathbf{v}$ and edge sequence $(e_{\ell-2^i+1}, \ldots, e_\ell)$. By Lemma 3.7, we can slide P in both directions until it is stuck, and the path remains a canonical geodesic during the sliding. Thus, $I_{\alpha,\mathbf{v}}$ is an interval, and so is the image of $I_{\alpha,\mathbf{v}}$ under $a_{\alpha,\mathbf{v}}$.

[5]Two directions $\overrightarrow{v}_i$ and $\overrightarrow{w}_i$ for the links hitting e_i are ordered by comparing $\theta(\overrightarrow{e}_i, \overrightarrow{v}_i)$ and $\theta(\overrightarrow{e}_i, \overrightarrow{w}_i)$.

[6]By Lemma 3.3, given two canonical geodesics from p to e_ℓ with adjacent direction vectors $\mathbf{v}$ and $\mathbf{w}$, then any linearly interpolation of the two different directions yield another direction vector for which there is a canonical geodesic from p to e_ℓ. The same holds for two canonical geodesics with adjacent direction vectors $\mathbf{v}$ and $\mathbf{w}$ from $e_{\ell-2^i}$ to the same point in e_ℓ. Therefore, $I_{\alpha,\mathbf{v}} \cap I_{\alpha,\mathbf{w}} \neq \emptyset \iff a_{\alpha,\mathbf{v}}[I_{\alpha,\mathbf{v}}] \cap a_{\alpha,\mathbf{w}}[I_{\alpha,\mathbf{w}}] \neq \emptyset$.

consists of the direction vector $(-\vec{e}_{\ell-1}, \vec{e}_\ell)$ or $(\vec{e}_{\ell-1}, -\vec{e}_\ell)$ depending on whether $e_{\ell-1}$ and e_ℓ share a negative or positive endpoint, respectively, and every vector consisting of a single direction that points to a vertex of H_ℓ and can be used to go from $e_{\ell-1}$ to e_ℓ. For $i > 0$, let β be the ancestor edge-node of α at level $\ell - 2^{i-1}$ ("midway" between α and α_0), and let $(\mathbf{u}_1, \ldots, \mathbf{u}_r)$ and $(\mathbf{v}_1, \ldots, \mathbf{v}_{r'})$ be the sequences of direction vectors in $\mathcal{L}^{i-1}_\beta$ and $\mathcal{L}^{i-1}_\alpha$, respectively. $\mathcal{L}^{i-1}_\beta$ has been computed as β is an ancestor of α. Choose an arbitrary point $p \in I_{\beta,\mathbf{u}_k} \cap I_{\beta,\mathbf{u}_{k+1}}$. For all $k \in [1, r-1]$, compute $\lambda_{\beta,k} = \frac{A_{\beta,\mathbf{u}_{k+1}}(p) - A_{\beta,\mathbf{u}_k}(p)}{\left\| a_{\beta,\mathbf{u}_k}(p)\ a_{\beta,\mathbf{u}_{k+1}}(p) \right\|}$. By P3, $\lambda_{\beta,k}$ is a constant, so $\lambda_{\beta,k}$ equals $\partial C^+_{p,\sigma}$ at $a_{\beta,\mathbf{u}_k}(p)$, where $\sigma = (e_{\ell-2^i+1}, \ldots, e_{\ell-2^{i-1}})$, which is consistent with Lemma 3.7: $\partial C^+_{p,\sigma}$ is independent of the source p and the destination. Similarly, take an arbitrary $q \in a_{\alpha,\mathbf{v}_k}[I_{\alpha,\mathbf{v}_k}] \cap a_{\alpha,\mathbf{v}_{k+1}}[I_{\alpha,\mathbf{v}_{k+1}}]$, and for $k \in [1, r'-1]$, compute $\pi_{\alpha,k} = \frac{B_{\alpha,\mathbf{v}_{k+1}}(q) - B_{\alpha,\mathbf{v}_k}(q)}{\left\| b_{\alpha,\mathbf{v}_k}(q)\ b_{\alpha,\mathbf{v}_{k+1}}(q) \right\|}$. By Lemma 3.3, $\mathcal{L}^i_\alpha$ consists of every concatenation $\mathbf{u}_j\mathbf{v}_k$ such that $\lambda_{\beta,j} + \pi_{\alpha,k} \geq 0$ and $\lambda_{\beta,j-1} + \pi_{\alpha,k-1} < 0$. By Lemma 3.2, $\lambda_{\beta,j} \leq \lambda_{\beta,j+1}$ and $\pi_{\alpha,k} \leq \pi_{\alpha,k+1}$, so we can scan $\lambda_{\beta,j}$ in increasing j and $\pi_{\alpha,k}$ in decreasing k to identify the good concatenations. We first find $k_0 \in [1, r']$ such that $\lambda_{\beta,1} + \pi_{\alpha,k_0} \geq 0$ and $\lambda_{\beta,1} + \pi_{\alpha,k_0-1} < 0$, and so $\mathbf{u}_1\mathbf{v}_{k_0}$ is a good concatenation. Note that $\lambda_{\beta,1} + \pi_{\alpha,k} < 0$ for all $k < k_0$. Next, we find $k_1 \leq k_0$ such that $\lambda_{\beta,2} + \pi_{\alpha,k_1} \geq 0$ and $\lambda_{\beta,2} + \pi_{\alpha,k_1-1} < 0$. Thus, $\lambda_{\beta,2} + \pi_{\alpha,k} \geq 0$ and $\lambda_{\beta,1} + \pi_{\alpha,k-1} < 0$ for all $k \in [k_1, k_0]$, which makes $\mathbf{u}_2\mathbf{v}_k$ a good concatenation for all $k \in [k_1, k_0]$. Repeating the above gives $\mathcal{L}^i_\alpha$. When adding a concatenation $\mathbf{uv}$, we compute the auxiliary information in $O(1)$ time: $I_{\alpha,\mathbf{uv}} = b_{\beta,\mathbf{u}}[\, I_{\alpha,\mathbf{v}} \cap a_{\beta,\mathbf{u}}[I_{\beta,\mathbf{u}}]\,]$, $A_{\alpha,\mathbf{uv}} = A_{\beta,\mathbf{u}} + A_{\alpha,\mathbf{v}} \circ a_{\beta,\mathbf{u}}$, $a_{\alpha,\mathbf{uv}} = a_{\alpha,\mathbf{v}} \circ a_{\beta,\mathbf{u}}$, $B_{\alpha,\mathbf{uv}} = B_{\alpha,\mathbf{v}} + B_{\beta,\mathbf{u}} \circ b_{\alpha,\mathbf{v}}$, and $b_{\alpha,\mathbf{uv}} = b_{\beta,\mathbf{u}} \circ b_{\alpha,\mathbf{v}}$, where the operator $\circ$ composes two functions. Appendix B.2 shows that the construction preserves P1–P4.

Lemma 3.8. *Creating an edge-node at level ℓ takes $O(2^i h)$ time, where 2^i is the largest power of 2 that divides ℓ, and h is the maximum size of the convex polygons associated with the faces.*

Proof. Let α be a new edge-node at level ℓ. $\mathcal{L}^0_\alpha$ stores $O(h)$ direction vectors. For $i > 0$, let β be the edge-node at level $\ell - 2^{i-1}$, the size of $\mathcal{L}^i_\alpha$ is at most the total size of $\mathcal{L}^{i-1}_\alpha$ and $\mathcal{L}^{i-1}_\beta$. Inductively, we obtain a time bound of $\sum_{j=0}^{i-1} O(2^j h) = O(2^i h)$. ◻

Compute a geodesic to a vertex. Suppose that we expand an edge-node α, which annexes a face corner (f, ν). Let α_0 be the nearest ancestor vertex-node of α at tree level $\ell_0 < \ell$. Let $(e_{\ell_0+1}, \ldots, e_\ell)$ be the edge sequence corresponding to the edge-nodes on the tree path from α_0 to α. We are to create a vertex-node β for ν and compute the cost of the canonical geodesic P_β from s to ν through the edges $(e_{\ell_0+1}, \ldots, e_\ell)$.

For $i = 1, 2, \cdots$, find the largest ℓ_i such that $\ell_i \leq \ell$ and $\ell_i - \ell_{i-1}$ is a power of 2 that divides both ℓ_i and ℓ_{i-1}. This gives a sequence $\ell_0 < \ell_1 < \ldots < \ell_r = \ell$, where $r = O(\log \ell)$. For $i \geq 1$, let $k_i = \ell_i - \ell_{i-1}$ and let α_i be the ancestor edge-node of α at level ℓ_i. We also use α_r to denote α. Let σ_i denote the edge sequence $(e_{\ell_0+1}, \ldots, e_{\ell_i})$ for $i \in [1, r]$. P_β is the concatenation of P_{α_0} and the canonical geodesic P from ν_{α_0} to ν through the edges $(e_{\ell_0+1}, \ldots, e_\ell)$. We already know $\mathrm{cost}(P_{\alpha_0})$. We compute $\mathrm{cost}(P)$ by combining the $\mathcal{L}^{k_i}_{\alpha_i}$'s in at most $r+1$ stages. At the end of the i-th stage, $i \in [1, r]$, we fix the prefix Q_i of P up to e_{ℓ_i}.[7]

Assume that Q_{i-1} is fixed. Let x_{i-1} denote its destination. Assume that we have computed $\partial C^+_{\nu_{\alpha_0},\sigma_{i-1}}(x_{i-1})$ and $\partial C^-_{\nu_{\alpha_0},\sigma_{i-1}}(x_{i-1})$. Consider the canonical geodesics from ν_{α_0} through $(e_{\ell_{i-1}+1}, \ldots, e_\ell)$ with Q_{i-1} as a common prefix. By Lemmas 3.2 and 3.4, these geodesics spread out from x_{i-1} to e_ℓ and form a fan that contains ν_α. We want to construct a path R from x_{i-1} to some point $y \in e_{\ell_i}$ such that $Q_i = Q_{i-1}R$ and the canonical geodesics that spread out from y to e_ℓ form a fan that contains ν_α. Let $\mathbf{u}_{i-1}$ be the direction vector of Q_{i-1}. The idea is to find the direction vector $\mathbf{v}$ in $\mathcal{L}^{k_i}_{\alpha_i}$ by binary search such that $\mathbf{uv}$ is the direction vector of Q_i.

[7] We define the prefix Q_r of P only up to $e_{\ell_r} = e_\ell$ instead of an edge of f incident to ν. It is because we will apply Lemmas 3.5–3.7, which require the nodes the path other than its source to be in the interior of edges.

The binary search works as follows. Let $\mathbf{v}$ be the "median" direction vector in the sublist of $\mathcal{L}^{k_i}_{\alpha_i}$ that we are working on. If $x_{i-1} \notin I_{\alpha_i,\mathbf{v}}$, we remove half of the sublist of $\mathcal{L}^{k_i}_{\alpha_i}$ and recurse. Suppose that $x_{i-1} \in I_{\alpha_i,\mathbf{v}}$. We find the smallest direction vector $\mathbf{w}$ and the largest direction vector $\mathbf{w}'$ such that $\mathbf{u}_{i-1}\mathbf{v}\mathbf{w}$ and $\mathbf{u}_{i-1}\mathbf{v}\mathbf{w}'$ extend Q_{i-1} to two canonical geodesics through $(e_{\ell_{i-1}+1},\ldots,e_\ell)$ and the face f. (We will describe how to find $\mathbf{w}$ and $\mathbf{w}'$ shortly.) If these two geodesics lie on the same side of ν_α, by Lemma 3.4, we can remove half of the sublist of $\mathcal{L}^{k_i}_{\alpha_i}$ and recurse. If these two geodesics sandwich ν_α, then $\mathbf{u}_{i-1}\mathbf{v}$ extends Q_{i-1} to Q_i. The destination of Q_i is $y = a_{\alpha_i,\mathbf{v}}(x_{i-1})$ and $\mathrm{cost}(Q_i) = \mathrm{cost}(Q_{i-1}) + A_{\alpha_i,\mathbf{v}}(x_{i-1})$. By Lemmas 3.5–3.7, $\partial C^+_{\nu_{\alpha_0},\sigma_i}(y)$ and $\partial C^-_{\nu_{\alpha_0},\sigma_i}(y)$ can be computed in $O(1)$ time from $\partial C^+_{\nu_{\alpha_0},\sigma_{i-1}}(x)$ and $\partial C^-_{\nu_{\alpha_0},\sigma_{i-1}}(x)$. Then we fix the next prefix Q_{i+1}. The binary search may also finish with two adjacent direction vectors in $\mathcal{L}^{k_i}_{\alpha_i}$ without fixing Q_i, a terminating case that we discuss after the next paragraph.

How do we find the smallest and largest direction vectors $\mathbf{w}$ and $\mathbf{w}'$? By Lemma 3.3, we find the smallest direction vector $\mathbf{w}_{i+1}$ in $\mathcal{L}^{k_{i+1}}_{\alpha_{i+1}}$ by binary search such that $\mathbf{v}\mathbf{w}_{i+1}$ is the direction vector of some canonical geodesic from x_{i-1} through $(e_{\ell_{i-1}+1},\ldots,e_{\ell_{i+1}})$. Let $y = a_{\alpha_i,\mathbf{v}}(x_{i-1})$ and let $z = a_{\alpha_{i+1},\mathbf{w}_{i+1}}(y)$. We apply Lemmas 3.5–3.7 to compute $\partial C^+_{\nu_{\alpha_0},\sigma_{i+1}}(z)$ and $\partial C^-_{\nu_{\alpha_0},\sigma_{i+1}}(z)$ in $O(1)$ time from $\partial C^+_{\nu_{\alpha_0},\sigma_i}(y)$ and $\partial C^-_{\nu_{\alpha_0},\sigma_i}(y)$. Then, we find the smallest direction vector $\mathbf{w}_{i+2}$ in $\mathcal{L}^{k_{i+2}}_{\alpha_{i+2}}$ by binary search and extend to $\mathbf{v}\mathbf{w}_{i+1}\mathbf{w}_{i+2}$. Repeating the above gives $\mathbf{w}_{i+1}\mathbf{w}_{i+2}\ldots\mathbf{w}_r$. Finally, we apply pick the smallest direction $\overleftrightarrow{w}_{r+1}$ according to Lemma 3.3 that extends $\mathbf{w}_{i+1}\mathbf{w}_{i+2}\ldots\mathbf{w}_r$ through f, and $\mathbf{w}_{i+1}\mathbf{w}_{i+2}\ldots\mathbf{w}_r\,(\overleftrightarrow{w}_{r+1})$ is the desired $\mathbf{w}$. The largest direction vector $\mathbf{w}'$ is obtained symmetrically.

Recall the terminating case that Q_i cannot be fixed and the binary search finishes with two adjacent direction vectors $\mathbf{v}$ and $\mathbf{v}'$ in $\mathcal{L}^{k_i}_{\alpha_i}$. We find the smallest direction vector $\mathbf{w}$ as before to extend $\mathbf{v}$ and $\mathbf{v}'$ through $(e_{\ell_i+1},\ldots,e_\ell)$ and f. Note that $\mathbf{v}\mathbf{w}$ and $\mathbf{v}'\mathbf{w}$ extend Q_{i-1} to two canonical geodesics that sandwich ν. The last direction $\overleftrightarrow{w}_{r+1}$ in $\mathbf{w}$ brings us from ν to a point $z \in e_\ell$, and the cost is $\mathrm{cost}(z\nu)$. We continue to $b_{\alpha_r,\mathbf{w}_r}(z) \in e_{\ell_{r-1}}$ and the cost accumulates to $\mathrm{cost}(z\nu) + B_{\alpha_i,\mathbf{w}_r}(z)$, and so on to a point $y \in e_{\ell_i}$ between $a_{\alpha_i,\mathbf{v}}(x_{i-1})$ and $a_{\alpha_i,\mathbf{v}'}(x_{i-1})$, where x_{i-1} is the destination of Q_{i-1}. Let C be the cost of the path that we have retraced from ν to e_{ℓ_i}. Suppose that $y = (1-t)a_{\alpha_i,\mathbf{v}}(x_{i-1}) + t\,a_{\alpha_i,\mathbf{v}'}(x_{i-1})$. By P3 and Lemma 3.3, $\mathrm{cost}(P) = \mathrm{cost}(Q_{i-1}) + (1-t)A_{\alpha_i,\mathbf{v}}(x_{i-1}) + tA_{\alpha_i,\mathbf{v}'}(x_{i-1}) + C$.

Another terminating case is that we proceed all the way to stage r and fix Q_r. Then, $\mathrm{cost}(P) = \mathrm{cost}(Q_r) + \mathrm{cost}(x_r\nu)$, where x_r is the destination of Q_r.

Theorem 3.1. *Let $\mathcal{T}$ be a polyhedral surface with n vertices in an instance of* PolyPath. *Given a source s, a destination t and an integer m, the shortest path from s to t on $\mathcal{T}$ with no more than m links can be found in $O(hmn\log mn + mn\log^2 m\log^2 hm)$ time, where h is the maximize size of the convex polygons that define the distance functions in the faces of $\mathcal{T}$.*

Proof. (Sketch) The correctness follows from Lemma B.2 that the pruning of the sequence tree preserves shortest paths. The details are given in Appendix B.3. We spend $O(\log mn)$ amortized time in each invocation of dominance testing and pruning (Appendix B.1). At most $O(mn)$ tree nodes are created and thus $O(mn\log mn)$ total time is spent.

Divide the edge-nodes into $O(\log m)$ groups such that an edge-node is in group i if its level is a multiple of 2^i but not 2^{i+1}. Group i contains $O(mn/2^i)$ edge-nodes. By Lemma 3.8, creating a node in group i takes $O(2^i h)$ time. So it takes $O(hnm\log m)$ time to create all the edge-nodes.

To compute the cost of a geodesic for a vertex-node, we fix $O(\log m)$ prefixes Q_i's. To extend Q_{i-1} to Q_i, we binary search in $\mathcal{L}^{k_i}_{\alpha_i}$ in $O(\log 2^i h) = O(\log \ell h) = O(\log hm)$ probes by Lemma 3.8. Each probe requires $O(\log \ell) = O(\log m)$ binary searches among the lists $\mathcal{L}^{k_{i+1}}_{\alpha_{i+1}}$, $\mathcal{L}^{k_{i+2}}_{\alpha_{i+2}},\ldots$. So it takes $O(\log^2 m\log^2 hm)$ time to compute the cost of a geodesic. The total time spent on all vertex-nodes is thus $O(mn\log^2 m\log^2 hm)$. We can reconstruct the direction vector of the shortest path in similar way as in dealing with the terminating case of not fixing some Q_i. So constructing the path takes only linear time, i.e. $O(m)$. ◻

4 Applications

Under the L_1 and L_∞ metrics, $h = O(1)$. Under the L_p metric for some $p \geq 2$, Dudley's result [16] allows us to approximate the "unit disk" by a polygon of $O(1/\sqrt{\varepsilon})$ vertices such that the polygon diameter is approximated with an ε relative error, i.e., $h = O(1/\sqrt{\varepsilon})$. In the above cases, $m = O(n)$ because there exists a shortest path that visits a face no more than once.

Theorem 4.1. *Given a polyhedral surface of size n, the L_1 and L_∞ shortest paths between two vertices can be computed in $O(n^2 \log^4 n)$ time, and for every constant $p \geq 2$ and every $\varepsilon \in (0,1)$, a $(1+\varepsilon)$-approximate L_p shortest path can be computed in $O\left(\frac{1}{\sqrt{\varepsilon}} n^2 \log n + n^2 \log^4 n\right)$ time.*

Figure 3: Left: The face f makes an angle ϕ_f with the horizontal, and the ascent is $\mathsf{len} \cdot \sin\varphi \sin\phi_f$. Right: The bold segment represents the clipping.

Consider path planning on a terrain with the cost function $c_1 \cdot$ Euclidean length $+\ c_2 \cdot$ total ascent for some constants $c_1 > 0$ and $c_2 \geq 0$. Refer to the left image in Figure 3. The ascent within a face f is $\mathsf{len} \cdot \sin\varphi \sin\phi_f$, where len is the distance travelled in f, ϕ_f is the gradient of a face f, and φ is the angle between the travel direction and the horizontal. Let S_f denote the "unit disk" induced. On the uphill side, the boundary of S_f satisfies the equation $1 = (c_1 + c_2 \sin\varphi \sin\phi_f)\,\mathsf{len}$; on the downhill side, the boundary of S_f is the half-circle with radius $1/c_1$. S_f is convex with bounded aspect ratio, so we can approximate it by Dudley's result [16] to obtain a PolyPath problem instance with $h = O(1/\sqrt{\varepsilon})$ and $m = O(n)$.

We can incorporate uphill gradient constraints. Let ψ be the input limit on the uphill path gradient. Let pq be an oriented segment in the interior of f that makes an angle larger than ψ with the horizontal. We can traverse a zigzag path from p to q in which each link makes an angle ψ with the horizontal. The path length is equal to the height difference between p and q divided by $\sin\psi$, irrespective of the exact zigzag pattern. Each link can be as short as we wish, and the zigzag path can stay inside f. Under this constraint, the top part of S_f that makes an angle at least ψ with the horizontal should be clipped. Refer to the right image in Figure 3. We can similarly handle downhill gradient constraints.

The above suggests that we can transform a shortest path P such that it visits a face no more than once. There are some technical issues though. Suppose that P visits f twice at points p and q. If p is in the interior of an edge, we may need to transform P to move from p to a nearby point $p' \in f$ and then move from p' to q by a zigzag path in f. Luckily, we can make $\mathrm{cost}(pp')$ negligible. If q is a vertex of f, it may be impossible to move from the interior of f straight to q. In this case, P enters q from another face f''. We transform P so that it moves uphill from a point q' near q on an edge of f to a point q'' near q on an edge of f'' and then to q, such that every link in the detour has uphill gradient ψ. The detour cost is negligible, and such detours around vertices introduce at most $O(n)$ extra links. Thus, after transforming, the path can be partitioned into two types of subpaths such that a subpath of the first type lies within a face of $\mathcal{T}$ and a face does not contain two subpaths of the first type, and the subpaths of the second type have $O(n)$ links and negligible cost altogether. This produces an instance of PolyPath with $h = O(1/\sqrt{\varepsilon})$ and $m = O(n)$. Appendix C gives the details.

Theorem 4.2. *Given a source s and a destination t on a polyhedral terrain of size n, we can find a $(1+\varepsilon)$-approximate shortest path under the cost function of $c_1\,$*length*$\,+\,c_2\,$*ascent* for some constants $c_1 > 0$ and $c_2 \geq 0$, where* length *is the Euclidean path length and* ascent *is the total ascent. Gradient constraints can be imposed. The running time is $O\left(\frac{1}{\sqrt{\varepsilon}} n^2 \log n + n^2 \log^4 n\right)$.*

References

[1] P.K. Agarwal, S. Har-Peled, M. Sharir, and K.R. Varadarajan. Approximating shortest paths on a convex polytope in three dimensions. *Journal of ACM*, 44 (1997), 567–584.

[2] M. Ahmed. Constrained Shortest Paths in Terrains and Graphs. PhD thesis, University of Waterloo, Canada, 2009.

[3] M. Ahmed, S. Das, S. Lodha, A. Lubiw, A. Maheshwari and S. Roy. Approximation algorithms for shortest descending paths in terrains. *Journal of Discrete Algorithms*, 8 (2010), 214–230.

[4] M. Ahmed and A. Lubiw. Shortest descending paths through given faces. *Computational Geometry: Theory and Applications*, 42 (2009), 464–470.

[5] M. Ahmed and A. Lubiw. Shortest descending paths: towards an exact algorithm. *International Journal of Computational Geometry and Applications*, 21 (2011), 431–466.

[6] M. de Berg and M. van Kreveld. Trekking in the Alps without freezing or getting tired. *Algorithmica*, 18 (1997), 306–323.

[7] J. Canny and J.H. Reif. New lower bound techniques for robot motion planning problems. *Proceedings of the 28th Annual IEEE Symposium on Foundation of Computer Science*, 1987, 49–60.

[8] T. Chan. Optimal output-sensitive convex hull algorithm in two and three dimensions. *Discrete and Computational Geometry*, 16 (1996), 361–368.

[9] J. Chen and Y. Han. Shortest paths on a polyhedron, part I: computing shortest paths. *International Journal of Computational Geometry and Applications*, 6 (1996), 127–144.

[10] S.-W. Cheng and J. Jin. Approximate shortest descending paths. *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2013.

[11] S.-W. Cheng, H.-S. Na, A. Vigneron, and Y. Wang. Approximate shortest paths in anisotropic regions. *SIAM Journal on Computing*, 38 (2008), 802–824.

[12] J. Choi, J. Sellen, and C.-K. Yap. Approximate Euclidean shortest path in 3-space. *Proceedings of the 10th Annual Symposium on Computational Geometry*, 1994, 41–48.

[13] S. Choi, J. Park, E. Lim, and W. Yu. Global path planning on uneven elevation maps. *Proceedings of the 9th International Conference on Ubiquitous Robots and Ambient Intelligence*, 2012, 49–54.

[14] K.L. Clarkson. Approximation algorithms for shortest path motion planning. *Proceedings of the 19th Annual ACM Symposium on Theory of Computation*, 1987, 56–65.

[15] K. Deng, X. Zhou, H.T. Shen, Q. Liu, K. Xu, and X. Lin. A multi-resolution surface distance model for k-nn query processing. *VLDB Journal*, 17 (2008), 1101–1119.

[16] R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *Approximation Theory*, 10(3):227–236, 1974.

[17] E. Galin, A. Peytavie, N. Maréchal, and E. Guérin. Procedural generation of roads. *Eurographics*, 2010.

[18] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28 (1999), 2215–2256.

[19] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4 (1984), 373–395.

[20] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *SIGGRAPH*, 2003, 954–961.

[21] M. van Kreveld. On quality paths on polyhedral terrains. *IGIS'94*, LNCS 884, 1994, 113–122.

[22] L. Liu and R. C.-W. Wong. Finding shortest path on land surface. *SIGMOD'11*, 433–444.

[23] J.S.B. Mitchell, D.M. Mount and C.H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16 (19897), 647–668.

[24] J.S.B. Mitchell and C.H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *Journal of ACM*, 38 (1991), 18–73.

[25] C.H. Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Information Processing Letters*, 20 (1985), 259–263.

[26] N.C. Rowe and R.S. Ross. Optimal grid-free path planning across arbitrarily contoured terrain with anisotropic friction and gravity effects. *IEEE Transactions on Robotics and Automation*, 6 (1990), 540–553.

[27] S. Roy, S. Das, and S.C. Nandy. Shortest monotone descent path problem in polyhedral terrain. *Computational Geometry: Theory and Applications*, 27 (2007), 115–133.

[28] Y. Schreiber and M. Sharir. An optimal-time algorithm for shortest paths on a convex polytope in three dimensions. *Proceedings of the 22nd annual symposium on Computational geometry*, 2006, 30–39.

[29] C. Shahabi, L.-A. Tang, and S. Xing. Indexing land surface for efficient knn query. *PVLDB*, 1 (2008), 1020–1031.

[30] M. Sharir and A. Schorr. On shortest paths in polyhedral spaces. *SIAM Journal on Computing*, 15 (1996), 193–215.

[31] Z. Sun and J.H. Reif. On finding energy-minimizing paths on terrains. *IEEE Transactions on Robotics*, 21 (2005), 102–114.

[32] K.R. Varadarajan and P.K. Agarwal. Approximating shortest paths on a non-convex polyhedron. *SIAM Journal on Computing*, 30 (2001), 1321–1340.

[33] C. Yu, J. Lee, and M.J. Munro-Stasiuk. Extensions to least-cost path algorithms for roadway planning. *International Journal of Geographical Information Science*, 17 (2003), 361–376.

A Missing proofs

A.1 Proof of Lemma 3.1

Let j be the smallest integer such that $p_j \neq q_j$. Since P is the canonical geodesic and $P[p_0, p_{j-1}] = Q[q_0, q_{j-1}]$, p_j must be on the negative side of q_j. If for all $i > j$, $p_i = q_i$ or

p_i is on the negative side of q_i, then we are done. Otherwise, let i be the smallest integer such that p_i is on the positive side of q_i. Then $p_{i-1}p_i$ must cross $q_{i-1}q_i$, say at x. Since both P and Q are geodesic paths, $P[x, p_k]$ and $Q[x, p_k]$ are both geodesics and have the same cost.

$$\text{cost}(P[p_0, p_{i-1}]) + d_{f_i}(p_{i-1}q_i) + \text{cost}(Q[q_i, p_k]) \leq \text{cost}(P[p_0, x]) + \text{cost}(Q[x, p_k]) = \text{cost}(P)$$

We obtain a new geodesic $R = (p_0, p_1, \ldots, p_{i-1}, q_i, q_{i+1}, \ldots, q_k)$. Note that $\theta(\vec{e}_i, q_i - p_{i-1}) < \theta(\vec{e}_i, p_i - p_{i-1})$. Therefore, the direction vector of R is lexicographically smaller than that of P, contradicting the assumption that P is a canonical geodesic.

A.2 Proof of Lemma 3.3

Suppose that P is a canonical geodesic. Then $P[p_0, p_i]$ and $P[p_i, p_k]$ are canonical geodesics as well. If p_i is not the positive endpoint of e_i, pick a point p'_i on the positive side of p_i and arbitrarily close to p_i. By the definition of the functions $\partial C^+_{p_0,\sigma_{0i}}$ and $\partial D^+_{p_k,\sigma_{ik}}$, we obtain

$$\begin{aligned} & C_{p_0,\sigma_{0i}}(p'_i) + D_{p_k,\sigma_{ik}}(p'_i) - C_{p_0,\sigma_{0i}}(p_i) - D_{p_k,\sigma_{ik}}(p_i) \\ = & \left(\partial C^+_{p_0,\sigma_{0i}}(p_i) + \partial D^+_{p_k,\sigma_{ik}}(p_i)\right) \cdot \|p'_i p_i\|. \end{aligned}$$

Since P is a geodesic, $C_{p_0,\sigma_{0i}}(p'_i) + D_{p_k,\sigma_{ik}}(p'_i) \geq C_{p_0,\sigma_{0i}}(p_i) + D_{p_k,\sigma_{ik}}(p_i)$, which implies that $\partial C^+_{p_0,\sigma_{0i}}(p_i) + \partial D^+_{p_k,\sigma_{ik}}(p_i) \geq 0$. If p_i is not the negative endpoint of e_i, we pick $p'_i \in e_i$ on the negative side of p_i and sufficiently close to p_i. Then,

$$\begin{aligned} & C_{p_0,\sigma_{0i}}(p_i) + D_{p_k,\sigma_{ik}}(p_i) - C_{p_0,\sigma_{0i}}(p'_i) + D_{p_k,\sigma_{ik}}(p'_i) \\ = & \left(\partial C^-_{p_0,\sigma_{0i}}(p_i) + \partial D^-_{p_k,\sigma_{ik}}(p_i)\right) \cdot \|p'_i p_i\|. \end{aligned}$$

By Lemma 3.1, $C_{p_0,\sigma_{0i}}(p'_i) + D_{p_k,\sigma_{ik}}(p'_i) > C_{p_0,\sigma_{0i}}(p_i) + D_{p_k,\sigma_{ik}}(p_i)$, and therefore, $\partial C^-_{p_0,\sigma_{0i}}(p_i) + \partial D^-_{p_k,\sigma_{ik}}(p_i) < 0$.

Conversely, suppose that the three conditions are satisfied for some $i \in [0, k]$. Let p'_i be the intersection point between e_i and the canonical geodesic from p_0 to p_k with edge sequence σ. If $p'_i = p_i$, we are done. Suppose that $p'_i \neq p_i$.

Consider the case of p'_i lying on the positive side of p_i. By Lemma 3.2, $C_{p_0,\sigma_{0i}}$ and $D_{p_k,\sigma_{ik}}$ are convex functions. Therefore,

$$\begin{aligned} C_{p_0,\sigma_{0i}}(p'_i) &\geq C_{p_0,\sigma_{0i}}(p_i) + \partial C^+_{p_0,\sigma_{0,i}}(p_i) \cdot \|p_i p'_i\| \\ D_{p_k,\sigma_{ik}}(p'_i) &\geq D_{p_k,\sigma_{ik}}(p_i) + \partial D^+_{p_k,\sigma_{ik}}(p_i) \cdot \|p_i p'_i\| \end{aligned}$$

Combining these two inequalities and condition (ii) in the lemma gives

$$C_{p_0,\sigma_{0i}}(p'_i) + D_{p_0,\sigma_{0i}}(p'_i) \geq C_{p_0,\sigma_{0i}}(p_i) + D_{p_0,\sigma_{0i}}(p_i),$$

which shows that P is also a geodesic. However, p_i is on the negative side of p'_i, which is a contradiction to Lemma 3.1.

Consider the case of p'_i lying on the negative side of p_i. By the convexity argument again, we obtain

$$\begin{aligned} C_{p_0,\sigma_{0i}}(p'_i) &\geq C_{p_0,\sigma_{0i}}(p_i) + \partial C^-_{p_0,\sigma_{0,i}}(p_i) \cdot \|p_i p'_i\| \\ D_{p_k,\sigma_{ik}}(p'_i) &\geq D_{p_k,\sigma_{ik}}(p_i) + \partial D^-_{p_k,\sigma_{ik}}(p_i) \cdot \|p_i p'_i\| \end{aligned}$$

But then these two inequalities and condition (iii) in the lemma imply that

$$C_{p_0,\sigma_{0i}}(p'_i) + D_{p_0,\sigma_{0i}}(p'_i) > C_{p_0,\sigma_{0i}}(p_i) + D_{p_0,\sigma_{0i}}(p_i).$$

This is impossible because P cannot be shorter than a geodesic.

A.3 Proof of Lemma 3.6

Consider the derivation of $\partial C^+_{p_0,\sigma_{0k}}(p_k)$ in (i). The derivation of $\partial C^-_{p_0,\sigma_{0k}}(p_k)$ is symmetric. Take a point $p'_k \in e_k$ on the positive side of p_k and arbitrarily close to p_k. For $j \in [i,k]$, let p'_j be the point in e_j such that $p'_j p'_{j+1}$ is parallel to $p_j p_{j+1}$. Since p'_k is arbitrarily close to p_k, p'_i is also arbitrarily close to p_i. Therefore,

$$\begin{aligned}
C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p'_k) &= C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p_k) + \partial C^+_{p_i,\sigma_{ik}}(p_k) \cdot \|p_k p'_k\| \\
&= C_{p_0,\sigma_{0k}}(p_k) + \partial C^+_{p_i,\sigma_{ik}}(p_k) \cdot \|p_k p'_k\|, \\
C_{p_0,\sigma_{0i}}(p'_i) + C_{p'_i,\sigma_{ik}}(p'_k) &= C_{p_0,\sigma_{0i}}(p_i) + \partial C^+_{p_0,\sigma_{0i}}(p_i) \cdot \|p_i p'_i\| + C_{p_i,\sigma_{ik}}(p_k) + \gamma_{ik} \cdot \|p_k p'_k\| \\
&= C_{p_0,\sigma_{0k}}(p_k) + \left(\delta_{ik} \cdot \partial C^+_{p_i,\sigma_{ik}}(p_k) + \gamma_{ik}\right) \cdot \|p_k p'_k\|.
\end{aligned}$$

Let Q be a geodesic path from p_0 to p'_k with edge sequence $\sigma_{0k} = \sigma$. Let r be the node of Q on e_i. We prove (ii) by showing that $C_{p_0,\sigma_{0k}}(p'_k)$ is equal to either $C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p'_k)$ or $C_{p_0,\sigma_{0i}}(p'_i) + C_{p'_i,\sigma_{ik}}(p'_k)$. There are three cases as shown in Figure 2 depending on the position of r.

Suppose that r is on the negative side of p_i. See Figure 2(a). $Q[r, p'_k]$ and $P[p_i, p_k]$ must cross in this case, say at point x. Since both P and Q are geodesics, their subpaths are also geodesics. Therefore, $\text{cost}(P[p_0, x]) = \text{cost}(Q[p_0, x])$, which yields $C_{p_0,\sigma_{0k}}(p'_k) = \text{cost}(Q) = \text{cost}(P[p_0, x]) + \text{cost}(Q[x, p'_k]) \geq C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p'_k)$. A geodesic to p'_k cannot be longer than any path to p'_k via p_i. Therefore, $C_{p_0,\sigma_{0k}}(p'_k) = C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p'_k)$.

Suppose that r is on the positive side of p'_i. See Figure 2(b). Since $C_{p_0,\sigma_{0i}}$ is a convex function by Lemma 3.2, $C_{p_0,\sigma_{0i}}(r) \geq C_{p_0,\sigma_{0i}}(p_i) + \partial C^+_{p_0,\sigma_{0i}}(p_i) \cdot \|p_i r\| = C_{p_0,\sigma_{0i}}(p'_i) + \partial C^+_{p_0,\sigma_{0i}}(p_i) \cdot \|p'_i r\|$, where the last equality follows from the fact that p'_i is arbitrarily close to p_i. Because $p'_j p'_{j+1}$ is parallel to $p_j p_{j+1}$ for all $j \in [i, k-1]$, we obtain $\partial D^+_{p'_k,\sigma_{ik}}(p'_i) = \partial D^+_{p_k,\sigma_{ik}}(p_i)$ by the assumption that $\partial D^+_{p_k,\sigma_{ik}}$ depends on the direction vector of $P[i,j]$ only. So $D_{p'_k,\sigma_{ik}}(r) \geq D_{p'_k,\sigma_{ik}}(p'_i) + \partial D^+_{p'_k,\sigma_{ik}}(p'_i) \cdot \|p'_i r\| = C_{p'_i,\sigma_{ik}}(p'_k) + \partial D^+_{p_k,\sigma_{ik}}(p_i) \cdot \|p'_i r\|$. Combining the two inequalities above gives

$$\begin{aligned}
C_{p_0,\sigma_{0k}}(p'_k) &= C_{p_0,\sigma_{0i}}(r) + D_{p'_k,\sigma_{ik}}(r) \\
&\geq C_{p_0,\sigma_{0i}}(p'_i) + \partial C^+_{p_0,\sigma_{0i}}(p_i) \cdot \|p'_i r\| + C_{p'_i,\sigma_{ik}}(p'_k) + \partial D^+_{p'_k,\sigma_{ik}} \cdot \|p'_i r\| \\
&= C_{p_0,\sigma_{0i}}(p'_i) + C_{p'_i \sigma_{ik}}(p'_k) + \left(\partial C^+_{p_0,\sigma_{ik}}(p_i) + \partial D^+_{p_k,\sigma_{ik}}(p_i)\right) \cdot \|p'_i r\| \\
&\geq C_{p_0,\sigma_{0i}}(p'_i) + C_{p'_i \sigma_{ik}}(p'_k). \qquad (\because \text{Lemma 3.3})
\end{aligned}$$

Suppose that $r \in p_i p'_i$. See Figure 2(c). Since $\|p_i r\|$ and $\|r p'_i\|$ are arbitrarily small, $C_{p_0,\sigma_{0k}}(p'_k) = C_{p_0,\sigma_{0i}}(r) + D_{p'_k,\sigma_{ik}}(r) = C_{p_0,\sigma_{0i}}(p_i) + \partial C^+_{p_0,\sigma_{0i}}(p_i) \cdot \|p_i r\| + D_{p'_k,\sigma_{ik}}(p'_i) - \partial D^+_{p'_k,\sigma_{ik}}(p'_i) \cdot \|p'_i r\|$. The last expression is a function linear in $\|p_i r\|$ by our assumption that $\partial C^+_{p_0,\sigma_{0i}}$ and $\partial D^+_{p_k,\sigma_{ik}}$ depend only on the direction vectors of $P[p_0, p_i]$ and $P[i,k]$ respectively (hence $\partial D^+_{p'_k,\sigma_{ik}}(p'_i) = \partial D^+_{p_k,\sigma_{ik}}(p_i)$). Therefore, the last expression is minimized when $r = p_i$ or $r = p'_i$. It follows that either $C_{p_0,\sigma_{0k}}(p'_k) = C_{p_0,\sigma_{0i}}(p_i) + D_{p'_k,\sigma_{ik}}(p_i) = C_{p_0,\sigma_{0i}}(p_i) + C_{p_i,\sigma_{ik}}(p'_k)$ or $C_{p_0,\sigma_{0k}}(p'_k) = C_{p_0,\sigma_{0i}}(p'_i) + D_{p'_k,\sigma_{ik}}(p'_i) = C_{p_0,\sigma_{0i}}(p'_i) + C_{p'_i,\sigma_{ik}}(p'_k)$.

The correctness of (ii) can be proved in a similar way.

A.4 Proof of Lemma 3.7

We first show that $\partial C^+_{p_{i-1},(e_i)}(p_i)$ depends only on the direction of $p_i - p_{i-1}$. Divide all directions into *cones*, each being the set of directions from the origin to all points in one edge of the polygon H_{f_i} defining the distance function for the face f bound by e_{i-1} and e_i.

If $p_i - p_{i-1}$ points to a vertex of H_{f_i}, there are two cones that contain $p_i - p_{i-1}$. We use ℓ_- to denote the support line of the edge of H_{f_i} defining the cone that comes first in anticlockwise

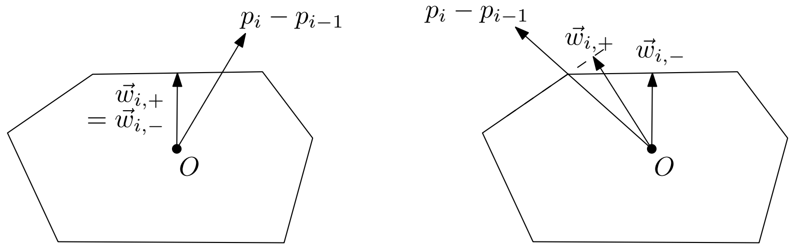

Figure 4: Left: The ray in the direction of $p_i - p_{i-1}$ crosses the boundary of H_{f_i} at a point that is not a vertex. $\vec{w}_{i,-} = \vec{w}_{i,+}$. Right: The ray in the direction of $p_i - p_{i-1}$ crosses the boundary of H_{f_i} at a vertex, so $\vec{w}_{i,-}$ and $\vec{w}_{i,+}$ are defined by the edges of H_{f_i} incident to that vertex.

order among these two cones, and ℓ_+ denotes the support line of the edge of H_{f_i} that defines the other cone. If $p_i - p_{i-1}$ points to the interior of an edge of H_{f_i}, then both ℓ_+ and ℓ_- denote the support line of this edge. Let $\vec{w}_{i,+}$ and $\vec{w}_{i,-}$ be the vectors that are orthogonal to ℓ_+ and ℓ_-, respectively. See Figure 4.

Then $\mathrm{cost}(p_{i-1}p'_i) = \langle p'_i - p_{i-1}, \vec{w}_{i,+} \rangle / \|\vec{w}_{i,+}\|$ and $\mathrm{cost}(p_{i-1}p_i) = \langle p_i - p_{i-1}, \vec{w}_{i,+} \rangle / \|\vec{w}_{i,+}\|$. So $\partial C^+_{p_{i-1},(e_i)}(p_i) = \langle \overleftrightarrow{e}_i, \vec{w}_{i,+} \rangle / \|\vec{w}_{i,+}\|$, which only depends on the direction of $p_i - p_{i-1}$. Similarly, one can verify that

$$\partial C^-_{p_{i-1},(e_i)}(p_i) = \frac{\langle \overleftrightarrow{e}_i, \vec{w}_{i,-} \rangle}{\|\vec{w}_{i,-}\|}, \ \partial D^+_{p_i,(e_i)}(p_{i-1}) = -\frac{\langle \overleftrightarrow{e}_{i-1}, \vec{w}_{i,+} \rangle}{\|\vec{w}_{i,+}\|}, \ \partial D^-_{p_i,(e_i)}(p_{i-1}) = \frac{\langle \overleftrightarrow{e}_{i-1}, \vec{w}_{i,-} \rangle}{\|\vec{w}_{i,-}\|}.$$

They all depend on the direction of $p_i - p_{i-1}$ only.

$\partial C^+_{p_0,\sigma_{01}}(p_1)$ and $\partial C^-_{p_0,\sigma_{01}}(p_1)$ depend only on the direction of $p_1 - p_0$ by the discussion above. Applying Lemma 3.6(ii) with $i = 1$ and $k = 2$ shows that $\partial C^+_{p_0,\sigma_{02}}(p_2)$ and $\partial C^-_{p_0,\sigma_{02}}(p_2)$ depend only on the directions of $p_1 - p_0$ and $p_2 - p_1$. By repeatedly applying Lemma 3.6(ii) with $k = i + 1$, one can show that $\partial C^+_{p_0,\sigma_{0j}}(p_j)$ and $\partial C^-_{p_0,\sigma_{0j}}(p_j)$ depend only on the direction vector of $P[p_0, p_j]$.

Analogously, using Lemma 3.6(iii), we can show that $\partial D^+_{p_k,\sigma_{jk}}(p_j)$ and $\partial D^-_{p_k,\sigma_{jk}}(p_j)$ depend only on the direction vector of $P[p_j, p_k]$. Thus, the conditions on Lemma 3.6(i) and (ii) can be removed.

A.5 Proof of Lemma 3.4

Without loss of generality, assume that $\theta(p_\ell - p_{\ell-1}, \vec{e}_\ell) = \theta(q_\ell - q_{\ell-1}, \vec{e}_\ell)$ for $\ell < i$. By Lemma 3.7, we can assume that $p_0 = q_0$. So $q_\ell = p_\ell$ for $\ell < i$ and q_i is on the positive side of p_i. By Lemma 3.2, $\partial C^+_{p_0,\sigma_{0i}}(q_i) \geq \partial C^-_{p_0,\sigma_{0i}}(q_i) \geq \partial C^+_{p_0,\sigma_{0i}}(p_i) \geq \partial C^-_{p_0,\sigma_{0i}}(p_i)$, which enforces that $\theta(p_{i+1} - p_i, \vec{e}_{i+1}) \geq \theta(q_{i+1} - q_i, \vec{e}_{i+1})$ by applying Lemma 3.3 with $k = i + 1$. Then q_{i+1} is also on the positive side of the p_{i+1}, and therefore $\partial C^+_{p_0,\sigma_{0,i+1}}(q_{i+1}) \geq \partial C^-_{p_0,\sigma_{0,i+1}}(q_{i+1}) \geq \partial C^+_{p_0,\sigma_{0,i+1}}(p_{i+1}) \geq \partial C^-_{p_0,\sigma_{0,i+1}}(p_{i+1})$. Repeating the argument, one can show $\theta(p_j - p_{j-1}, \vec{e}_j) \geq \theta(q_j - q_{j-1}, \vec{e}_j)$ for all $j > i$.

B Missing details of the algorithm

B.1 Dominance checking and tree pruning

The vertex-node case is easy. For each vertex of $\mathcal{T}$, we record the current corresponding vertex-node β that dominates all other vertex-nodes that correspond to v_β. When a new vertex-node α corresponding to v_β is created, we compare α and β to see which of the two dominates the other. Thus, it takes only $O(1)$ time modulo the time for computing the geodesic cost from s to v_α with edge sequence σ_α.

It takes more time to handle edge-nodes. For every face corner (f, v), we record the edge-node β that annexes (f, v) and dominates all other edge-nodes annexing (f, v). We say that β *occupies* (f, v). Suppose that a new edge-node α annexing (f, v) is generated. Let α' and β' be the nearest proper ancestor vertex-nodes of α and β, respectively. Let P and Q be the geodesics from $v_{\alpha'}$ and $v_{\beta'}$ to v through the edges in $\tilde{\sigma}_\sigma$ and $\tilde{\sigma}_\beta$, respectively. We must have computed and recorded $\text{cost}(P_{\beta'}) + \text{cost}(Q)$ beforehand as β occupies (f, v). Therefore, modulo the time to compute $\text{cost}(P)$, we can compare $\text{cost}(P_{\alpha'}) + \text{cost}(P)$ with $\text{cost}(P_{\beta'}) + \text{cost}(Q)$ to decide the dominance in $O(1)$ time. Without loss of generality, assume that α dominates β. Then, α replaces β as the edge-node that occupies (f, v).

To decide which child edge-node of β to prune, we need to refine the notion of dominance. Consider the two edge sequences σ_α and σ_β. Let e denote the first edge in the longest common suffix of σ_α and σ_β.

- If σ_α is not a suffix of σ_β, let e_α be the edge in σ_α before e. Then α dominates β *on the positive side* (resp. *negative side*) if e_α and e share the positive (resp. negative) endpoint.

- If σ_α is a suffix of σ_β, let e_β be the edge in σ_β before e, and α dominates β on the positive side (resp. negative side) if e_β and e share the negative (resp. positive) endpoint.

Correspondingly, we use e^+ and e^- to denote the two edges of f incident to v such that v is the positive and negative endpoints of e^+ and e^-, respectively. Suppose that α dominates β on the positive side. If β has been already expanded, we delete the child edge-node of β corresponding to e^+ as well as its descendants. If β has not yet been expanded, we will not let β gain a child edge-node corresponding to e^+. The pruning is symmetric for the case of α dominating β on the negative side.

Recall that we do not explicitly store σ_β for a tree node β. So we cannot just trace σ_α and σ_β to decide whether α dominates β on the positive or negative side. This tracing could be time-consuming anyway. Instead, we introduce some auxiliary data structures for making this decision. For every face corner (f, v), we maintain a ordered list of edge-nodes annexing it. Note that these edge-nodes correspond to the same edge e of f, and e is oriented consistently with respect to these edge-nodes. Let u^+ and u^- be the positive and negative endpoints of e, respectively. Let α' and β' be the parent nodes of α and β, respectively. Let $g = wu^+u^-$ be the face of $\mathcal{T}$ that shares e with f. The ordering of two edge-nodes α and β in the ordered list for (f, v) is determined as follows.

- Suppose that α' and β' annex different face corners of g. If $e_{\alpha'}$ and e share the common positive endpoint u^+, then α precedes β in the ordered list for (f, v). Otherwise, $e_{\alpha'}$ and e share the common negative endpoint u^-, and β precedes α in the ordered list for (f, v).

- If α' and β' are edge-nodes annexing the same corner of g, and α' precedes β' in the ordered list for that face corner, then α precedes β in the ordered list for (f, v).

- If α' is an edge-node annexing (g, u^+) and β' is a vertex-node corresponding to w, then α precedes β in the ordered list for (f, v).

- If α' is a vertex-node corresponding to w and β' is an edge-node annexing (g, u^-), then α precedes β in the ordered list for (f, v).

Assume that α dominates β. If α precedes β in the ordered list for (f, v), then α dominates β on the positive side; otherwise, α dominates β on the negative side. The rules above are based on the information at the parents of α and β in such a way that the decision process is equivalent to tracing σ_α and σ_β. This explains the correctness. Since an edge-node annexing (f, v) can change, we need to represent the sorted list for (f, v) with a balanced binary search tree. The sorted list is no more than the tree size which is $O(mn)$. Therefore, the dominance testing can be done in $O(\log mn)$ time.

B.2 Preservation of properties P1–P4

Lemma B.1. *For any $i \geq 0$, $\mathcal{L}_\alpha^i$ satisfies* P1–P4.

Proof. Consider the base case of $i = 0$. P1 clearly holds because any direction vector added has only one link. P2 also holds by the choices of directions picked by the algorithm. Suppose that $\mathbf{v}_j = (\overrightarrow{v}_j)$ and $\mathbf{v}_{j+1} = (\overrightarrow{v}_{j+1})$ are two adjacent direction vectors. The direction of the oriented segment from p to any point between $a_{\alpha,\mathbf{v}_j}(p)$ and $a_{\alpha,\mathbf{v}_{j+1}}(p)$ lies between $\overrightarrow{v}_j$ and $\overrightarrow{v}_{j+1}$. Since $\overrightarrow{v}_j$ and $\overrightarrow{v}_{j+1}$ point to the same edge of the convex polygon that defines the distance function, the cost of the segment from p to a point between $a_{\alpha,\mathbf{v}_j}(p)$ and $a_{\alpha,\mathbf{v}_{j+1}}(p)$ is a linear interpolation of $A_{\alpha,\mathbf{v}_j}(p)$ and $A_{\alpha,\mathbf{v}_{j+1}}(p)$. Therefore, P3 holds. P4 can be proved similarly.

Consider the case of $i > 0$. Let ℓ be the level of α. Let β be the ancestor edge-node of α at level $\ell - 2^{i-1}$. Assume that P1–P4 hold for both $\mathcal{L}_\alpha^{i-1}$ and $\mathcal{L}_\beta^{i-1}$. P1 holds for $\mathcal{L}_\alpha^i$ by the order in which good concatenations are added.

Consider P2. Any two successive concatenations added to $\mathcal{L}_\alpha^i$ share either a prefix, i.e. $\mathbf{uv}$ and $\mathbf{uv}'$, or a suffix, i.e. $\mathbf{uv}$ and $\mathbf{u}'\mathbf{v}$. By P2, $\mathbf{u}$ and $\mathbf{u}'$ differ in exactly one entry, and so do $\mathbf{v}$ and $\mathbf{v}'$. It follows that P2 is also satisfied by $\mathbf{uv}$ and $\mathbf{uv}'$ and $\mathbf{uv}$ and $\mathbf{u}'\mathbf{v}$.

Consider P3. Take any two adjacent direction vectors in $\mathcal{L}_\alpha^{i-1}$ and $\mathcal{L}_\beta^{i-1}$. They differ at one entry by P2 and we can write them as $\mathbf{u} = \mathbf{w}\,(\overrightarrow{w}_0)\,\mathbf{w}'$ and $\mathbf{v} = \mathbf{w}\,(\overrightarrow{w}_1)\,\mathbf{w}'$. Let P and Q be the canonical geodesics from p to $a_{\alpha,\mathbf{u}}(p)$ and $a_{\alpha,\mathbf{v}}(p)$ respectively. Consider the canonical geodesic R from p to a point $q = (1-t)a_{\alpha,\mathbf{u}}(p) + t\,a_{\alpha,\mathbf{v}}(p)$ for some $t \in [0,1]$. By Lemma 3.3, the direction vector of R is $\mathbf{w}\,(\overrightarrow{w})\,\mathbf{w}'$, where $\overrightarrow{w}$ lies between $\overrightarrow{w}_0$ and $\overrightarrow{w}_1$. Let rx, ry and rz be the segments of P, R and Q, respectively, that have directions $\overrightarrow{x}$, $\overrightarrow{y}$ and $\overrightarrow{z}$, respectively. Because $R[y,q]$, $P[x, a_{\alpha,\mathbf{u}}(p)]$ and $Q[z, a_{\alpha,\mathbf{v}}(p)]$ have the same direction vector, we get $y = (1-t)x + tz$, and $\mathrm{cost}(R[y,q]) = (1-t)\,\mathrm{cost}(P[x, a_{\alpha,\mathbf{u}}(p)]) + t\,\mathrm{cost}(Q[z, a_{\alpha,\mathbf{v}}(p)])$. By P2, $\mathrm{cost}(ry) = (1-t)\,\mathrm{cost}(rx) + t\,\mathrm{cost}(rz)$. Therefore, $\mathrm{cost}(R) = (1-t)\,\mathrm{cost}(P) + t\,\mathrm{cost}(Q)$.

P4 can be proved similarly. ◻

B.3 Correctness of the algorithm

The following lemma was originally proved for L_∞ metric. But the proof only uses the triangle inequality, so the result can be naturally generalized to the distance functions defined in this paper.

Lemma B.2 ([10, Lemma 3.1])**.** *Let α and β be two edge-nodes annexing the same face corner (f, ν) such that α dominates β on the positive side (resp. negative side). Let e be the edge in f whose negative (resp. positive) endpoint is ν.*

(i) *α is not a descendant of β.*

(ii) *Let α' and β' be the nearest proper ancestor vertex-nodes of α and β, respectively. For every point $x \in e$ and every geodesic Q with edge sequence $\tilde{\sigma}_\beta \cdot (e)$ from $\nu_{\beta'}$ to ν, the geodesic P with edge sequence $\tilde{\sigma}_\alpha \cdot (e)$ from $\nu_{\alpha'}$ to ν satisfies* $\mathrm{cost}(P_{\alpha'}) + \mathrm{cost}(P) \leq \mathrm{cost}(P_{\beta'}) + \mathrm{cost}(Q)$*, and if they are equal, then α is expanded before β.*

Consider the correctness of the algorithm. Let P_0 be the shortest path from s to t with no more than m links. By the requirement of the PolyPath problem, we can assume that every node of P_0 is either a transversal node or a vertex of $\mathcal{T}$. If there are multiple choices for P_0, we pick P_0 to be one that has the fewest nodes.

The sequence tree is grown to contain the prefix of P_0 until the vertex-node corresponding to t is reached or an edge-node α_0 is dominated by some other edge-node β such that the child node of α_0 that would contain a longer prefix of P_0 is pruned. In the former case, the sequence tree captures the edge sequence of P_0, and the algorithm computes the cost of the geodesic with respect to that edge sequence, so we are done. Consider the latter case. Let x be the intersection point between P_0 and the edge corresponding to α_0 and β. By Lemma B.2, β contains a path Q to x that is at least as good as $P[s,x]$. Let $P_1 = Q \cdot P[x,t]$. By our choice of P_0, $\text{cost}(P_1) = \text{cost}(P_0)$, P_1 has the same number of nodes as P_0, and β is at the same depth as α_0 but expanded earlier. Note that β cannot be dominated by any other edge-node. The subtree of β grows to contain P_1, or a descendant α_1 of β is dominated by some other edge-node and the child of α_1 that would contain a longer prefix of P_1 is pruned. We can then repeat the analysis above, which can happen at most m times. The correctness thus follows.

C Reduction to a PolyPath instance

Let P be a shortest path s to t that satisfies the gradient constraints. Consider the intersections between P and a face f. Let p be the first point of entry. Let q be the last point of exit. For now assume that neither p nor q is a vertex of f. Let p_0 and q_0 be in the interior of f arbitrarily close to p and q, respectively, such that we can move from p to p_0 and from q_0 to q with uphill gradient ψ. We assume that the gradient of pq exceeds ψ; otherwise, we can connect p and q and skip P.

Without loss of generality, assume that q is higher than p. Let ψ be the ascent gradient bound. Let p' be the last point on P after p such that p and p' have the same height. That is, the subpath after p' never drops below the height of p. Let P' be the longest ascending subpath of P that starts at p' and does not go above the height of q. Let x be the other endpoint of P'.

Let y be the point on the segment $pq \subset f$ that has the same height as x. We can follow a zigzag path from p to p_0 and then to y, and the length of this path is $H/(\sin\psi)$, where H is the height difference between p and y. The subpath of P' from p' to x cannot be shorter because the same height difference H is covered with a ascent gradient no greater than ψ. It follows that the subpath of P from p to x is not shorter than the zigzag path length from p to y. At the same time, the ascent of the zigzag path is the smallest possible because it does not goes monotonically upward.

We can repeat the above argument to the next ascending subpath of P that rises above the point x. Eventually, we reach the conclusion that the a zigzag path from p to q is as long as P, and this zigzag path has the smallest ascent possible.

If p or q is a vertex of f, then we may have to first detour as described in the paragraph in Section 4 preceding Theorem 4.2 before applying the detour in the above. Again, the detour cost is negligible.

We have already described in Section 4 how our approximate terrain navigation problem gives rise to a convex distance function induced by a convex polygon of $O(1/\sqrt{\varepsilon})$ vertices. We have argued in the above that there exists a path with cost arbitrarily close to the optimum, and it can be partitioned into subpaths of two types such that a subpath of the first type lies within a face of $\mathcal{T}$ and a face does not contain two subpaths of the first type, and the subpaths of the second type have $O(n)$ links and negligible cost altogether. Although a subpath of the first type has many links due to the zigzag, zigzagging is only needed for the physical path on the terrain. Under the convex distance function metric induced by the problem, the zigzag path from p to q is replaced by an oriented segment from p to q. Similarly, a subpath of the second type may also zigzag within a face, and the zigzagging is replaced by a single oriented segment under the convex distance function metric. This produces an instance of PolyPath with $h = O(1/\sqrt{\varepsilon})$ and $m = O(n)$.