Shortest Path Problems on a Polyhedral Surface[☆,☆☆]

Atlas F. Cook IV[a], Carola Wenk[a]

[a] *University of Texas at San Antonio, Department of Computer Science,*
One UTSA Circle, San Antonio, TX 78249-0667
Telephone: (210) 458-4436 — Fax: (210) 458-4437
acook@cs.utsa.edu,carola@cs.utsa.edu

Abstract

We develop algorithms to compute edge sequences, Voronoi diagrams, shortest path maps, the Fréchet distance, and the diameter for a polyhedral surface. Distances on the surface are measured either by the length of a Euclidean shortest path or by link distance.

Key words: Polyhedral Surface, Voronoi Diagram, Shortest Path Map, Fréchet distance, Diameter, Link Distance, Euclidean Shortest Path

1. Introduction

Two questions are invariably encountered when dealing with shortest path problems. The first question is how to represent the combinatorial structure of a shortest path. In the plane with polygonal obstacles, a shortest path can only turn at obstacle vertices, so a shortest path can be combinatorially described as a sequence of obstacle vertices [28]. On a polyhedral surface, a shortest path need not turn at vertices [35], so a path is often described combinatorially by an *edge sequence* that represents the sequence of edges encountered by the path [1]. A benefit of representing shortest paths by edge sequences is that a series of unfolding rotations can be used to reduce the problem of computing a shortest path on a polyhedral surface into a two-dimensional problem. This process is described more fully in section 3.

The second commonly encountered shortest path question is how to compute shortest paths in a problem space with M vertices. The following preprocessing schemes compute combinatorial representations of all possible shortest paths. In a

[☆] This work has been supported by the National Science Foundation grant NSF CAREER CCF-0643597.

[☆☆] A previous version of this work has been presented in [21].

simple polygon, Guibas et al. [28] give an optimal $\Theta(M)$ preprocessing scheme that permits a shortest path to be computed between any two points in $O(\log M)$ time. In the plane with polygonal obstacles, Chiang and Mitchell [16] support shortest path queries between any two points after $O(M^{11})$ preprocessing. On a *convex* polyhedral surface, Mount [37] shows that $\Theta(M^4)$ combinatorially distinct shortest path edge sequences exist, and Schevon and O'Rourke [40] show that only $\Theta(M^3)$ of these edge sequences are *maximal* (i.e., they cannot be extended at either end without creating a suboptimal path). Agarwal et al. [1] use these properties to compute the $\Theta(M^4)$ shortest path edge sequences in $O(M^6 2^{\alpha(M)} \log M)$ time and the *diameter* in $O(M^8 \log M)$ time, where $\alpha(M)$ is the inverse Ackermann function. The diameter is the largest shortest path distance between any two points on the surface. Despite recent efforts by Chandru et al. [13] to improve the runtimes of Agarwal et al. [1], these runtimes have not improved since 1997. Our main result improves the edge sequence and diameter algorithms of [1] by a linear factor. We achieve this improvement by combining the star unfolding technique of [1] with the kinetic Voronoi diagram structure of Albers et al. [3]. A kinetic Voronoi diagram allows its defining point sites to move.

A popular alternative to precomputing all combinatorial shortest paths is to precompute a shortest path map structure SPM(s) that describes all shortest paths from a fixed source s. In the plane with polygonal obstacles, Hershberger and Suri [30] use the continuous Dijkstra paradigm to support all queries from a fixed source after $\Theta(M \log M)$ preprocessing. On a (possibly non-convex) polyhedral surface, Mitchell, Mount, and Papadimitriou [35] use the continuous Dijkstra paradigm to construct SPM(s) by propagating a wavefront over a polyhedral surface in $O(M^2 \log M)$ time and $O(M^2)$ space. Chen and Han [15] solve the same polyhedral surface problem in $O(M^2)$ time and space by combining unfolding and Voronoi diagram techniques. Schreiber and Sharir [41] use the continuous Dijkstra paradigm to construct an *implicit* representation of a shortest path map for a *convex* polyhedral surface in $O(M \log M)$ time and space.

Another popular variation on shortest path problems is to consider different methods of measuring the length of a shortest path. For example, suppose the "length" of a path is measured not by its Euclidean distance but by the number of edges on the path. The *link distance* [7, 25, 32, 36, 42] between two points is the minimum number of edges necessary to connect the points with a polygonal path. In a simple polygon, Suri [42] shows how to construct a shortest path map that can answer queries from a fixed source in $\Theta(M)$ time and space. Both Arkin, Mitchell, and Suri [7] and Efrat et al. [25] give $\Theta(M^3)$ time and space algorithms that support queries between any two points in a simple polygon. In the plane with polygonal obstacles, Mitchell, Rote, and Woeginger [36] support queries from a fixed source after $\Theta(M^4)$ time and space preprocessing. Surprisingly, this paper

appears to be the first to consider link distance problems on a polyhedral surface.

One reason that shortest path and link distance problems are of crucial importance is that they can serve as building blocks for higher level problems such as the Fréchet distance. The Fréchet distance measures the similarity of continuous shapes [5, 8, 11, 26, 39] by calculating a distance-based value that represents the similarity of the shapes. Although the traditional Fréchet distance operates in a space that is free of obstacles, recent works have realized the potential of the Fréchet distance in domains with obstacles. Efrat et al. [26] apply the Fréchet distance to create constrained morphs. Buchin et al. [11] compute the Fréchet distance between simple polygons. Chambers et al. [12] explore a homotopic variation of the Fréchet distance. Cook and Wenk [18] compute the Fréchet distance inside a simple polygon. Maheshwari and Yi [33] explore the Fréchet distance on a convex polyhedral surface.

1.1. Notation

Throughout this paper, M is the total complexity of a polyhedral surface and any objects such as points, line segments, and polygonal curves that lie on the surface. A shortest path on a polyhedral surface between points s and t is denoted by $\pi(s,t)$, and $d(s,t)$ signifies the Euclidean length of $\pi(s,t)$. The notations $\pi_L(s,t)$ and $d_L(s,t)$ represent analogous concepts for link distance. A convex polyhedral surface is denoted by $\mathcal{P}$, and a non-convex polyhedral surface is signified by $\mathcal{P}_N$. The extremely slowly growing inverse Ackermann function is represented by $\alpha(M)$. The line segment with endpoints a and b is denoted by $\overline{ab}$.

1.2. Our Euclidean Shortest Path Results

Table 1 summarizes our Euclidean shortest path results. For a *convex* polyhedral surface, Agarwal et al. [1] give algorithms to compute the diameter and either the exact set or a superset of all $\Theta(M^4)$ shortest path edge sequences. All three of these algorithms are improved by a linear factor in sections 3.1 and 3.2. Our improvement takes advantage of small combinatorial changes between adjacent events and combines star unfolding and kinetic Voronoi diagram techniques.

Section 3.3 contains an algorithm to compute the Fréchet distance between polygonal curves on a convex polyhedral surface, and this algorithm is a linear factor faster than the algorithm of Maheshwari and Yi [33]. In addition, section 3.3 contains the first algorithm to compute the Fréchet distance between polygonal curves on a *non-convex* polyhedral surface. Our motivation for studying the Fréchet distance on a polyhedral surface is that teaming up two people for safety reasons is common practice in many real-life situations, ranging from scouts in summer camp, to fire fighters and police officers, and even to astronauts exploring the moon. In all of these applications, two team members need to coordinate their movement

		Time	Space
Star Unfolding maintained over all edges	$\mathcal{P}_N$	$O(M^4)$	$O(M^4)$
Kinetic Voronoi Diagram maintained over all edges	$\mathcal{P}$	$O(M^5 2^{\alpha(M)} \log M)$	$O(M^5 2^{\alpha(M)})$
Edge Sequences (superset)	$\mathcal{P}$	$O(M^5)$	$O(M^5)$
Edge Sequences (exact)	$\mathcal{P}$	$O(M^5 2^{\alpha(M)} \log M)$	$O(M^4 2^{\alpha(M)})$
Diameter	$\mathcal{P}$	$O(M^7 \log M)$	$O(M^4)$
Fréchet Distance	$\mathcal{P}$	$O(M^6 \log^2 M)$	$O(M^2)$
	$\mathcal{P}_N$	$O(M^7 \log^2 M)$, $\Omega(M^4)$	$O(M^3)$
SPM($\overline{ab}$,$\mathcal{P}$)	$\mathcal{P}$	$O(M^4 2^{\alpha(M)} \log M)$	$O(M^4 2^{\alpha(M)})$
SPM($\overline{ab}$,$\mathcal{P}_N$)	$\mathcal{P}_N$	$O(M^{9+\kappa})$	$O(M^9)$

Table 1: **Our Euclidean shortest path results.** $\mathcal{P}$ (resp. $\mathcal{P}_N$) indicates a convex (resp. non-convex) polyhedral surface. $\overline{ab}$ is a line segment with endpoints a and b. The shortest path maps SPM($\overline{ab}$,$\mathcal{P}$) and SPM($\overline{ab}$,$\mathcal{P}_N$) support queries from any point $s \in \overline{ab}$ to any point on the surface. The Fréchet distance lower bound applies to the complexity of the free space diagram. κ is any constant greater than zero.

in order to stay within "walking distance" so that fast assistance can be offered in case of an emergency. The Fréchet distance is an ideal model for this scenario.

Section 3.3 also demonstrates that a subset of the star unfolding called the "core" can overlap itself for a non-convex polyhedral surface. Section 3.4 describes shortest path maps that support queries from any point on a line segment.

1.3. Our Link Distance Results

Link distance is fundamentally different from Euclidean distance and has a wealth of applications including robotic motion, wireless communications, geographic information systems, VLSI, computer vision, solid modeling, image processing, and even water pipe placement. These applications are naturally modeled by link distance because turns are costly while straight line movements are inexpensive.

Table 2 summarizes all of our link distance results on a polyhedral surface. Section 4.1 describes algorithms to compute link-based shortest path maps, and these structures are used to compute link-based diameters and Voronoi diagrams. Section 4.2 describes link-based algorithms for the Hausdorff distance and Fréchet distance on a polyhedral surface. Surprisingly, one of our Fréchet distance results

		Time	Space
SPM($\overline{ab}$, $\mathcal{C}$)	$\mathcal{C}$	$\Theta(M)$	$\Theta(M)$
SPM(s, $\mathcal{N}$), SPM($\min_{s\in\overline{ab}}$, $\mathcal{N}$)	$\mathcal{N}$*	$\Theta(M^4)$	$\Theta(M^4)$
SPM($\overline{ab}$,$\overline{cd}$)	$\mathcal{N}$	$O(M^6\lambda_6(M))$, $\Omega(M^4)$	$O(M^7)$
Link-Based Diameter	$\mathcal{C}$	$O(M^2)$	$O(M)$
	$\mathcal{N}$	$O(M^{\frac{19}{3}}\log^{3.11} M)$	$O(M^3)$
Voronoi Diagram	$\mathcal{C}$	$\Theta(M)$	$\Theta(M)$
	$\mathcal{N}$	$O(M^6)$, $\Omega(M^4)$	$O(M^6)$
Hausdorff Distance	$\mathcal{C}$	$O(M\log M)$	$O(M)$
for points	$\mathcal{N}$	$O(M^{\frac{10}{3}}\log^{3.11} M)$	$O(M^2)$
Hausdorff Distance	$\mathcal{C}$	$O(M\log M)$	$O(M)$
for line segments	$\mathcal{N}$*	$O(M^4\alpha(M)\log^2 M)$	$O(M^3)$
Fréchet Distance	$\mathcal{C}$	$O(M^2)$, $\Omega(M^2)$	$O(M^2)$
	$\mathcal{N}$*	$O(M^9\log M)$, $\Omega(M^6)$	$O(M^4)$

Table 2: **Our link distance results.** $\mathcal{C}$ is a *convex* subdivision of a polyhedral surface such that no two adjacent faces are coplanar, and $\mathcal{N}$ is an arbitrary polyhedral surface (see section 4). The shortest path map SPM($\overline{ab}$, $\mathcal{C}$) supports $d_L(s,t)$, $\pi_L(s,t)$ queries from any point $s \in \overline{ab} \in \mathcal{C}$ to any point $t \in \mathcal{C}$, SPM(s, $\mathcal{N}$) supports $d_L(s,t)$, $\pi_L(s,t)$ queries from a fixed source point s to any point $t \in \mathcal{N}$, SPM($\min_{s\in\overline{ab}}$, $\mathcal{N}$) supports $\min_{s\in\overline{ab}} d_L(s,t)$ and $\min_{s\in\overline{ab}} \pi_L(s,t)$ queries, and SPM($\overline{ab}$,$\overline{cd}$) supports $d_L(s,t)$, $\pi_L(s,t)$ queries between any points $s \in \overline{ab} \in \mathcal{N}$ and $t \in \overline{cd} \in \mathcal{N}$. The Fréchet distance lower bounds apply to the complexity of the free space diagram. An asterisk * indicates that, in addition to the exact runtimes that are shown, we also give approximation algorithms.

can be computed a logarithmic factor faster than the traditional Fréchet distance in the plane [5].

2. Preliminaries

A *polyhedral surface* is a "connected union of a finite number of polygonal faces, with any two polygons intersecting in a common edge, a common vertex, or not at all," and each edge belonging to at most two polygons [34]. A *Voronoi diagram* [10, 38] of M sites is a subdivision of a space such as the plane into maximal regions such that all points within a given region have the same nearest neighbor site with respect to some distance measure. A *shortest path map* is also a subdivision of a space into maximal regions. A traditional shortest path map

SPM(s) [34] is defined from a fixed source s such that all points within a region have the same combinatorial shortest path to s.

The *Hausdorff distance* [5, 6] is a similarity metric commonly used to compare sets of points or sets of higher-dimensional objects such as line segments or triangles. Since the Hausdorff distance relies heavily on nearest neighbor distance calculations, it is often computed with a Voronoi diagram. The *directed* Hausdorff distance is $\tilde{\delta}_H(A, B) = \sup_{a \in A} \inf_{b \in B} d(a, b)$, where A and B are compact sets and d is a distance metric for points. The (*undirected*) Hausdorff distance is the larger of the two directed distances: $\delta_H(A, B) = \max(\tilde{\delta}_H(A, B), \tilde{\delta}_H(B, A))$.

The *Fréchet distance* [5] is a similarity metric for *continuous* shapes that is defined for two polygonal curves $A, B : [0, 1] \to \mathbb{R}^{\mathfrak{d}}$ as

$$\delta_F(A, B) = \inf_{\alpha,\beta:[0,1]\to[0,1]} \sup_{t \in [0,1]} d(\ A(\alpha(t)), B(\beta(t))\ )$$

where α and β range over continuous non-decreasing reparameterizations and d is a distance metric for points. For a given constant $\varepsilon \geq 0$, *free space* is $\{(s, t) \mid s \in A,\ t \in B,\ d(s, t) \leq \varepsilon\}$. A *cell* is the parameter space defined by two line segments $\overline{ab} \in A$ and $\overline{cd} \in B$, and the free space inside the cell consists of all points $\{(s, t) \mid s \in \overline{ab}, t \in \overline{cd}, d(s, t) \leq \varepsilon\}$.

Alt and Godau's [5] Fréchet *decision problem* decides whether the Fréchet distance $\delta_F(A, B) \leq \varepsilon$ for some constant $\varepsilon \geq 0$. To make this decision, they build a *free space diagram* which measures the distance between all pairs of points $s \in A$ and $t \in B$. Dynamic programming is then used to check for the existence of a monotone path through the free space. Such a monotone path only permits non-decreasing reparameterizations. The Fréchet *optimization problem* [5] returns the smallest value of ε such that the decision problem returns true, and this optimal value of ε is the Fréchet distance.

3. Shortest Path Problems on a Polyhedral Surface

This section contains all of our Euclidean shortest path results on a polyhedral surface. Sections 3.1 and 3.2 describe algorithms that compute shortest path edge sequences and the diameter of a convex polyhedral surface. These results improve algorithms of Agarwal et al. [1] by a linear factor. Sections 3.3 and 3.4 contain algorithms to compute the Fréchet distance and shortest path maps.

3.1. Shortest Path Edge Sequences

This section contains superset and exact algorithms to compute the $\Theta(M^4)$ shortest path edge sequences on a convex polyhedral surface $\mathcal{P}$. Both of these algorithms improve results of Agarwal et al. [1] by a linear factor.

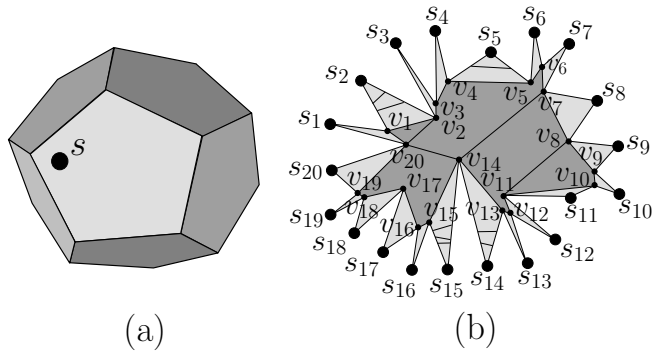

Figure 1: (a) A convex polyhedral surface $\mathcal{P}$. (b) The star unfolding of $\mathcal{P}$ is created by cutting along the shortest paths from s to every vertex $v_1, ..., v_{20}$ of $\mathcal{P}$. The source point s has an *image* s_i in the star unfolding for each vertex v_i. The heavily-shaded *core* of $\mathcal{P}$ is a simple polygon defined by the closed polygonal *equator* through the points $v_1, ..., v_{20}, v_1$ [1, 15].

Let $v_1, ..., v_M$ be the vertices of $\mathcal{P}$, and let $\Pi = \{\pi(s, v_1), ..., \pi(s, v_M)\}$ be an angularly ordered set of non-crossing shortest paths from a source point $s \in \mathcal{P}$ to each vertex $v_j \in \mathcal{P}$. The *star unfolding* $\mathcal{S}$ is a simple polygon [9] defined by cutting $\mathcal{P}$ along each of the shortest paths in Π and unfolding the resulting shape into the plane. Since the source point s touches all of the M cuts, $s \in \mathcal{P}$ maps to M image points $s_1, ..., s_M$ on the (two-dimensional) boundary of the unfolded simple polygon $\mathcal{S}$ (see Figures 1 and 3).

The *equator* [15] in the star unfolding is the closed polygonal curve through the points $v_1, ..., v_M, v_1$. The region inside the equator contains no source image and is called the *core* [23].[1] The regions outside the core each contain a source image and are collectively referred to as the *anti-core* [23]. A *core edge* is an image of an edge of $\mathcal{P}$ that was not cut during the unfolding process. Each of the $O(M)$ core edges has both of its endpoints at vertices and is entirely contained in the core. Each of the $\Theta(M^2)$ *anti-core edges* is an image of an edge of $\mathcal{P}$ that was cut during the unfolding process (see Figure 1b).

The star unfolding of s can be used to compute a shortest path $\pi(s, t)$ for points $s, t \in \mathcal{P}$ as follows. The shortest path $\pi(s, t)$ can always be represented by a two-dimensional shortest path in the star unfolding that originates from one of the source images $s_1, ..., s_M$ and terminates at the image of t. If the image of t lies in the anti-core region containing s_i, then the two-dimensional shortest path in the star unfolding from s_i to the image of t is optimal. By contrast, if the image of

[1] The core has also been referred to as the *kernel* or the *antarctic* in [1, 15]. Note that neither the star unfolding nor its core are necessarily star-shaped [1].

t lies in the core, then the nearest source image can be determined with Voronoi diagram techniques, and the two-dimensional shortest path in the star unfolding from this nearest source image to the image of t is optimal. This means that it is easier to determine a shortest path when t maps to an anti-core region than when t maps to the core.

Agarwal et al. [1] partition the M edges of the convex polyhedral surface $\mathcal{P}$ into $O(M^3)$ line segment *edgelets* such that all source points on an edgelet can be associated with the same combinatorial star unfolding. These edgelets are constructed in $O(M^3 \log M)$ time by computing a shortest path between each pair of vertices on $\mathcal{P}$ and intersecting these $O(M^2)$ shortest paths with each of the M edges of $\mathcal{P}$ [1]. Agarwal et al. [1] compute a star unfolding for each edgelet and use these structures to construct an $O(M^6)$ superset of the $\Theta(M^4)$ shortest path edge sequences for $\mathcal{P}$ in $O(M^6)$ time and space [1]. In addition, Agarwal et al. [1] show how to compute the exact set of $\Theta(M^4)$ shortest path edge sequences in $O(M^6 2^{\alpha(M)} \log M)$ time.

Although we have previously mentioned the star unfolding only for a *convex* polyhedral surface $\mathcal{P}$, the concept generalizes to a *non-convex* polyhedral surface $\mathcal{P}_N$ because the star unfolding can still be defined by an angularly ordered set of non-crossing shortest path cuts from the source to every vertex [15, 35]. In addition, there are still $O(M^3)$ edgelets on $\mathcal{P}_N$ because a shortest path between each pair of vertices can intersect each edge at most once.

We show how to *maintain* a combinatorial star unfolding in $O(M^4)$ total time and space as a source point varies continuously over all $O(M^3)$ edgelets on a possibly non-convex polyhedral surface $\mathcal{P}_N$. Our approach takes advantage of small combinatorial changes between adjacent edgelets and achieves a linear factor improvement over the approach [1] of computing a separate star unfolding for each edgelet.

Theorem 1. *A star unfolding can be maintained as a source point s varies continuously over all M edges of a (possibly non-convex) polyhedral surface $\mathcal{P}_N$ in $O(M^4)$ time and space.*

Proof. Let $\Pi = \{\pi(s, v_1), ..., \pi(s, v_M)\}$ be an angularly ordered set of non-crossing shortest path edge sequences from a source point $s \in \mathcal{P}$ to each vertex $v_j \in \mathcal{P}$. Since a star unfolding is defined combinatorially by Π, we first maintain Π as s varies continuously over all M edges of $\mathcal{P}_N$. To do this, we compute a star unfolding for each vertex $v_j \in \mathcal{P}_N$ in $O(M^3)$ total time and space [15], and we compute $O(M^3)$ edgelets in $O(M^3 \log M)$ time and space [1]. As s varies continuously over all edges, a shortest path $\pi(s, v_j) \in \Pi$ can only change combinatorially when s touches one of the $O(M^3)$ edgelet endpoints. Each edgelet endpoint is defined by an edge of a shortest path map SPM(v_j) and consequently identifies which shortest

path $\pi(s, v_j)$ changes at that endpoint. The new path $\pi(s, v_j)$ can be looked up using the precomputed star unfolding for v_j and substituted for the old path in Π in $O(M)$ time. Thus, Π can be maintained over all edgelets in $O(M \cdot M^3)$ total time.

Each change to Π requires removing and adding a constant number of $O(M)$ complexity anti-core regions from the star unfolding $\mathcal{S}$ and possibly updating all $O(M)$ core edges in $\mathcal{S}$.As s varies continuously in the interior of an edgelet, each source image is parameterized along a line segment in the star unfolding, and the remaining vertices in the star unfolding are fixed [1]. See Figures 3a and 3b. Thus, $O(M \cdot M^3)$ time and space is sufficient to maintain $\mathcal{S}$ combinatorially over all edgelets. □

The below lemma computes a superset of the shortest path edge sequences on $\mathcal{P}$ in $O(M^5)$ time and space. Note that we do not attempt to compute shortest path edge sequences on a non-convex polyhedral surface $\mathcal{P}_N$ because Mount [37] has shown that there can be exponentially many shortest path edge sequences on $\mathcal{P}_N$.

Theorem 2. *A superset of the $\Theta(M^4)$ shortest path edge sequences for a convex polyhedral surface $\mathcal{P}$ with M vertices can be constructed in $O(M^5)$ time and space.*

Proof. Each edgelet defines a star unfolding with source images $s_1, ..., s_M$. For each s_i, construct an edge sequence from s_i to each of the $O(M)$ anti-core edges in the anti-core region containing s_i and to each of the $O(M)$ core edges. This yields $O(M^2)$ edge sequences per edgelet, and $O(M^5)$ edge sequences over all edgelets. The result is the desired superset because only core edges have shortest path edge sequences to multiple sites, and this approach considers all possibilities. These edge sequences can be stored with $O(M^5)$ space in the traditional prefix tree structure of [1]. □

The *exact* set of shortest path edge sequences for each combinatorial star unfolding can be determined with a kinetic Voronoi diagram that allows its defining point sites to move. In our case, the moving sites are the source images $s_1, ..., s_M$, and a result of Agarwal et al. [1] ensures that each source image is parameterized along a line segment as a source point varies continuously over an edgelet. Albers et al. [3] show that for point sites moving along line segments at constant speeds, each *pair* of sites defines $\lambda_4(M) = O(M2^{\alpha(M)})$ Voronoi events. They also show how to maintain a kinetic Voronoi diagram in $O(\log M)$ time per event with a sorted event queue.

Theorem 3. *A kinetic Voronoi diagram of source images $s_1, ..., s_M$ can be maintained in $O(M^4 2^{\alpha(M)} \log M)$ time and $O(M^4 2^{\alpha(M)})$ space as a source point varies continuously over* one *edge e of a convex polyhedral surface $\mathcal{P}$.*

Proof. A kinetic Voronoi diagram for the *first* edgelet on e defines $O(M^2 \cdot M2^{\alpha(M)})$ events [3] due to the linear motion of $O(M^2)$ *pairs* of source images in the star unfolding. Each of the $O(M^2)$ subsequent edgelets on e can be handled by removing one source image site and adding one new source image site. All other sites continue to be parameterized along the same line segments as in the previous edgelet. Thus, each of these $O(M^2)$ edgelets contributes $M - 1$ new *pairs* of sites and $O(M \cdot M2^{\alpha(M)})$ new events to the event queue. Handling each event in $O(\log M)$ time and $O(1)$ space as in [3] yields the stated runtime. □

We construct the exact set of shortest path edge sequences as follows. For the moment, fix an edgelet α and a core vertex $v_i \in \mathcal{S}$ such that v_i touches the anti-core region that contains the source image s_i. Maintaining a kinetic Voronoi diagram over all points $s \in \alpha$ yields a two-dimensional parameterized Voronoi cell φ_i for the source image s_i. Due to the properties of Voronoi diagrams, the unique edge sequence in the star unfolding from s_i to a core edge e represents a shortest path if and only if e intersects φ_i for some $s \in \alpha$. This follows because s_i must be a nearest source image to some point on e in order to define a shortest path to e.

One way to decide whether a core edge e ever intersects a parameterized kinetic Voronoi cell φ_i is to represent each vertex of φ_i as an algebraic curve and to exhaustively test whether each algebraic curve intersects e. However, this approach ultimately requires testing whether each algebraic curve intersects each of the $O(M)$ (fixed) core edges. To improve this technique from $O(M)$ time per algebraic curve to $O(\log M)$ time per algebraic curve, Agarwal et al. [1] combine triangulation and upper envelope techniques to ensure that each algebraic curve defines at most two maximal shortest path edge sequences. A modified version of their approach is described below.

Agarwal et al. [1] triangulate the region of the core that is directly visible to core vertex v_i such that each triangle Δ has apex v_i. The dual graph D of the (fixed) core for the edgelet α is a *tree* [1] that defines candidate edge sequences. Let the portion of D inside a fixed triangle Δ be the subtree D_Δ. By [1], the maximum degree of any vertex in D_Δ is three, and there is at most one degree three vertex. This follows for two reasons (see Figure 2). First, a triangle Δ cannot contain a core vertex in its interior, so each core edge is a *chord* of Δ. Second, a degree three vertex must have three core edges defining its face, and this happens at most once.

Agarwal et al. [1] compute each subtree D_Δ in $O(M)$ time. In the following lemma, we improve this process to $O(\log M)$ time.

Lemma 1. *A subtree D_Δ can be computed in $O(\log M)$ time.*

Proof. Assume Δ has vertices v_i, v_j, v_k (see Figure 2). Since D_Δ has at most one degree three vertex and the maximum degree of any vertex in D_Δ is three, D_Δ

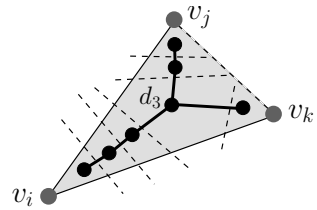

Figure 2: The subtree D_Δ has at most one degree three vertex d_3. Core edges are dashed.

consists of one path from the face containing v_i to the face containing v_j and a second path from the face containing v_i to the face containing v_k (see Figure 2). Point location in the core can identify the endpoints of these two paths in the tree D in $O(\log M)$ time, and these paths define a representation of D_Δ. □

After computing the subtree D_Δ for each triangle Δ, Agarwal et al. [1] use polar coordinates centered at core vertex v_i to compute an upper envelope μ of the algebraic curves defining the kinetic Voronoi cell φ_i. This upper envelope is then refined into a set of curve segments such that each curve segment is contained in some triangle Δ. For each curve segment, a binary search is performed on the two paths in D_Δ. The deepest edge on each of these two paths that is intersected by a curve segment defines a *maximal* shortest path edge sequence. Repeating this technique for all core vertices defined by all edgelets yields $\Theta(M^3)$ *maximal* shortest path edge sequences. The set of all prefixes of these maximal sequences defines all $\Theta(M^4)$ shortest path edge sequences of $\mathcal{P}$ [1].

Theorem 4. *The exact set of $\Theta(M^4)$ shortest path edge sequences for a convex polyhedral surface $\mathcal{P}$ with M vertices can be constructed in $O(M^5 2^{\alpha(M)} \log M)$ time and $O(M^4 2^{\alpha(M)})$ space.*

Proof. Let n_i be the total number of parameterized Voronoi vertices over all edgelets, and let t_Δ be the time to process each triangle Δ. There are $O(M^5)$ possible triangles Δ because each of the $O(M^3)$ edgelets defines $O(M)$ core vertices, and each of these vertices defines $O(M)$ triangles. The technique of Agarwal et al. [1] requires $O(n_i \log M + M^5 t_\Delta)$ time. Since they assume $n_i \in O(M^6 2^{\alpha(M)})$ and $t_\Delta \in O(M)$, this yields $O(M^6 2^{\alpha(M)} \log M)$ time.

We improve this runtime as follows. By Theorem 3, $n_i \in O(M^5 2^{\alpha(M)})$ over all $O(M)$ edges of $\mathcal{P}$. By Lemma 1, $t_\Delta \in O(\log M)$ time. Thus, we achieve $O(M^5 2^{\alpha(M)} \log M)$ total time. The space bound follows by storing the kinetic Voronoi diagram for only one edge at a time. □

3.2. Diameter

The *diameter* of a polyhedral surface is the largest shortest path distance between any pair of points on the surface. Agarwal et al. [1] have previously shown how to compute the diameter for a *convex* polyhedral surface in $O(M^8 \log M)$ time. They compute shortest paths between all pairs of vertices, and these shortest paths induce an arrangement of $O(M^4)$ *ridge-free regions* on the surface. A benefit of this construction is that all source points in a ridge-free region have the same combinatorial star unfolding. For each ridge-free region, a parameterized lower envelope with $O(M^4)$ vertices can be built that corresponds to a kinetic Voronoi diagram [1]. This structure is parameterized to allow the source point to vary continuously over an entire ridge-free region. The below approach computes the diameter a linear factor faster than [1] by taking advantage of small combinatorial changes between adjacent ridge-free regions.

Theorem 5. *The diameter of a convex polyhedral surface $\mathcal{P}$ with M vertices can be constructed in $O(M^7 \log M)$ time and $O(M^4)$ space.*

Proof. Pick an initial ridge-free region and construct its kinetic Voronoi diagram in $O(M^4)$ time as in [1]. Process the remaining ridge free regions in depth-first order so that the current ridge-free region r_c is always adjacent to a previously processed region r_p. This ordering takes advantage of small combinatorial changes between adjacent ridge-free regions. In particular, the star unfolding for r_c can be obtained from the star unfolding for r_p by removing and inserting one source image site. This implies that the kinetic Voronoi diagram for r_c involves only $M-1$ pairs of sites that are not present in r_p. Each of these $M-1$ pairs defines $O(M^2)$ Voronoi vertices by [1], so each ridge-free region r_c defines $O(M^3)$ new Voronoi vertices. This yields a total of $O(M^7)$ parameterized Voronoi vertices over all $O(M^4)$ ridge-free regions. Each parameterized Voronoi vertex v can be associated with a function $f(v)$ that represents the distance from v to its defining source image, and the maximum value for each $f(v)$ can be determined [1] in $O(\log M)$ time. The diameter of $\mathcal{P}$ is the largest distance defined by any of these functions. □

3.3. Fréchet Distance

This section contains algorithms to compute the Fréchet distance on both convex and non-convex polyhedral surfaces. Let $\delta_C(A, B)$ (resp. $\delta_N(A, B)$) denote the Fréchet distance between polygonal curves A and B on a convex (resp. non-convex) polyhedral surface. We define M as the number of vertices in a triangulated version of the surface such that every edge of A and B appears as a set of edges in the triangulation.

Maheshwari and Yi [33] have previously shown how to compute $\delta_C(A, B)$ in $O(M^7 \log M)$ time by enumerating all edge sequences. However, their approach relies on [31] whose key claim "has yet to be convincingly established" [1]. By contrast, we use the star unfolding from section 3.1 to compute $\delta_C(A, B)$ in $O(M^6 \log^2 M)$ time and $O(M^2)$ space. We build a *free space diagram* [5] to measure the distance $d(s, t)$ between all pairs of points $s \in A$ and $t \in B$. Each *cell* in our free space diagram is the parameter space defined by an edgelet $\alpha \in A$ and either a core edge or an anti-core edge in the combinatorial star unfolding for α. A cell is always interior-disjoint from all other cells.

To compute $\delta_C(A, B)$, we determine for a given constant $\varepsilon \geq 0$ all points $\{(s, t) \mid s \in A,\ t \in B,\ d(s, t) \leq \varepsilon\}$ that define the *free space* [5]. The star unfolding $\mathcal{S}$ maps a fixed source point $s \in A$ to a set $s_1, ..., s_M$ of source image points in $\mathcal{S}$ and maps the polygonal curve B to a set $\beta_1, ..., \beta_{O(M^2)}$ of core and anti-core edges in $\mathcal{S}$. Since s maps to multiple images in $\mathcal{S}$, free space is defined by the union of a set of disks $d_1, ..., d_M$, where each disk d_i has radius ε and is centered at s_i (see Figure 3). This follows by [9, 13] because all L_2 distances in the star unfolding for a *convex* polyhedral surface are at least as large as the shortest path between those two points (even when the L_2 path does not stay inside the boundary of the star unfolding). As the source point s varies continuously over an edgelet $\alpha \in A$, the core is fixed and each s_i is parameterized along a line segment l_i in the star unfolding [1]. This is illustrated in Figures 3a and 3b. The below $\delta_C(A, B)$ *decision problem* decides whether the Fréchet distance between polygonal curves A and B on a convex polyhedral surface is at most some given constant $\varepsilon \geq 0$.

Theorem 6. *The $\delta_C(A, B)$ decision problem can be computed in $O(M^6 \log M)$ time and $O(M^2)$ space.*

Proof. Partition the polygonal curve A into $O(M^3)$ edgelets and maintain a star unfolding for these edgelets in $O(M^4)$ total time by Theorem 1. Let β be an *anti-core* edge that lies in the anti-core region containing the source image s_i. The anti-core edge β changes length inside its triangular anti-core region as s varies continuously over an edgelet $\alpha \in A$ (see Figure 3b). In the free space diagram, this motion defines a constant complexity algebraic cell C that represents the parameter space for the edgelet $\alpha \in A$ and the anti-core edge $\beta \in B$. *Free space* in the cell C is defined by the intersection of C with the ellipse $E_{l_i,\beta} = \{(s, t) \mid s \in l_i,\ t \in \beta,\ ||s - t|| \leq \varepsilon\}$ [5]. This follows because the anti-core edge β is always at least as close to the source image s_i as to any other source image $s_{j \neq i}$ [13]. Since the free space defined by each anti-core edge has constant complexity and is interior-disjoint from all other cells, the $O(M^5)$ anti-core edges contribute $O(M^5)$ total complexity to the free space diagram.

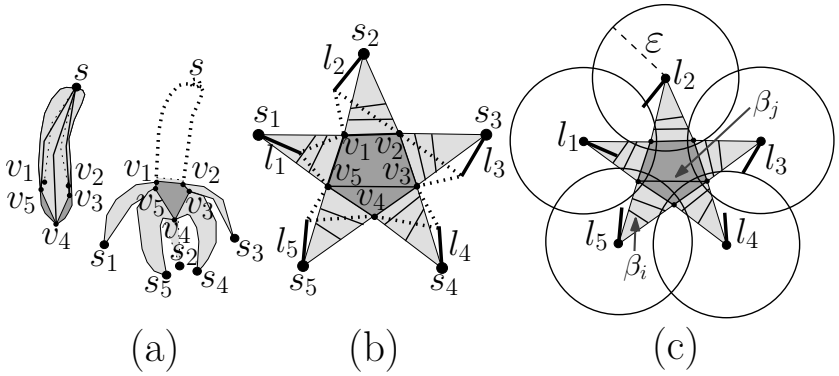

Figure 3: **(a)** A source point s on a polyhedral banana with vertices $v_1, ..., v_5$ defines a star unfolding $\mathcal{S}$ with source images $s_1, ..., s_5$. **(b)** As s varies continuously along an edgelet α, the darkly-shaded core of $\mathcal{S}$ is fixed while the images of s vary along line segments $l_1, ..., l_5$ [1]. Although the lightly-shaded triangles in the anti-core vary continuously with respect to $s \in \alpha$, their movement does not change the combinatorial structure of $\mathcal{S}$. **(c)** Inside the core, there are $O(M)$ images of the edges of $\mathcal{P}$ (e.g., β_j) that touch two vertices of $\mathcal{P}$. Inside the anti-core, there are $O(M^2)$ images of the edges of $\mathcal{P}$ (e.g., β_i) that were cut during the unfolding process. Free space for edges in the anti-core is completely described by a single disk (e.g., the disk centered on l_5 is always closest to β_i). Free space for edges in the core (e.g., β_j) is defined by the union of all $O(M)$ disks.

The $O(M)$ *core edges* have fixed positions as s varies continuously over an edgelet α, and the free space for any core edge γ is defined by the union of the M ellipses $E_{l_1,\gamma}, ..., E_{l_M,\gamma}$ (see Figure 3c). The union of these ellipses has $O(M^2)$ complexity and can be computed for each core edge in $O(M^2 2^{\alpha(M)})$ time [2]. Thus, the free space defined by all $O(M)$ core edges has $O(M^3)$ total complexity per edgelet and $O(M^6)$ complexity over all edgelets. Reachability information can be propagated through the free space diagram via plane sweep [19] in $O(M^6 \log M)$ time. By storing one star unfolding, one cell, and one vertical line segment of the free space diagram at a time, $O(M^2)$ space is sufficient.[2] The decision problem is affirmative if and only if the upper right corner of the free space diagram is reachable. □

For a *non-convex* polyhedral surface, the star unfolding can overlap itself [15], and shortest paths can turn at vertices in the star unfolding [35]. By Chen and Han [15], a shortest path $\pi(s,t)$ is either a line segment in the star unfolding from

[2]One vertical line segment of the free space diagram is defined by all distances between a fixed point $s \in A$ and the polygonal curve B. The free space defined by s and the polygonal curve B has $O(M^2)$ complexity because each of the $O(M^2)$ anti-core edges is intersected with a single fixed ellipse and each of the $O(M)$ core edges is intersected with $O(M)$ fixed ellipses.

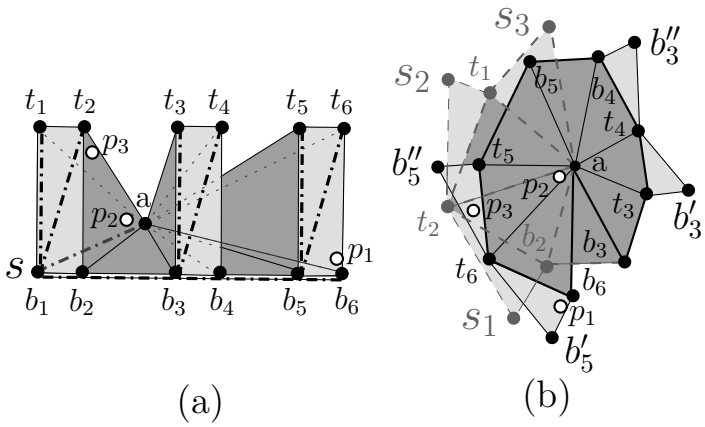

Figure 4: (a) A non-convex polyhedral surface can define (b) a star unfolding with an unfolded core that overlaps itself. Vertices on the boundary of the core are shown in the sequence a, b_6, t_6, t_5, b_5, b_4, t_4, t_3, b_3, b_2, t_2, t_1. The core begins to overlap itself in this ordering after vertex b_3, and these edges are drawn dashed. The three unfolded images of s are labeled s_1, s_2, and s_3.

some source image s_i to an unfolded image of t, or $\pi(s,t)$ is the concatenation of $\pi(s,v_j)$ with $\pi(v_j,t)$, where v_j is a vertex of the star unfolding.

Figure 4b illustrates a star unfolding for a non-convex polyhedral surface. Notice that only some of the lightly-shaded anti-core regions contain a vertex that is a source image. The remaining anti-core regions contain a vertex v_j that is similar to a source image but is additively weighted by $d(s,v_j)$ [15]. For example, the shortest path $\pi(s,p_1)$ can be obtained from the star unfolding in Figure 4b as the concatenation of $\pi(s,b'_5)$ with $\pi(b'_5,p_1)$.

Although it was previously known [15] that anti-core regions for a non-convex polyhedral surface could overlap other anti-core regions, it was not previously known whether the core could overlap itself. The below lemma asserts that the core can overlap itself.

Lemma 2. *The core of the star unfolding for a non-convex polyhedral surface can overlap itself.*

Proof. Figure 4 illustrates a non-convex polyhedral surface whose star unfolding has an overlapping core. The three-dimensional surface consists of three rectangular faces and eleven triangular faces. The three rectangular faces all lie in a common plane and are defined by the set of vertices $V = \{b_1, ..., b_6, t_1, ..., t_6\}$. The eleven triangular faces are all defined by a common apex vertex a and two vertices from V. The eleven angles incident to a are $\angle b_1at_1$, $\angle t_1at_2$, $\angle t_2ab_2$, $\angle b_2ab_3$, $\angle b_3at_3$, $\angle t_3at_4$, $\angle t_4ab_4$, $\angle b_4ab_5$, $\angle b_5at_5$, $\angle t_5at_6$, and $\angle t_6ab_6$. The sum of these angles can be greater than 2π by making the three rectangular faces sufficiently tall

(because this causes the six angles $\angle b_i a t_i$ for $i = 1, ..., 6$ to exceed $\frac{\pi}{3}$). Shortest paths from the source point $s = b_1$ to each vertex are shown as thick dash-dotted line segments. Note that the shortest path $\pi(s, a)$ is the line segment from s to a that overlaps the edge $\overline{b_1 a}$. Cutting along these shortest paths and unfolding into the plane yields the star unfolding for this surface. The shortest paths $\pi(s, t_4)$ and $\pi(s, t_6)$ turn at a vertex because all vertices in V lie in the same plane, and the $\frac{\pi}{2}$ angles defining the rectangular faces prevent shortcuts through the interiors of "alley" faces such as $b_2 a b_3$.

The rectangular faces and the hidden face $b_1 a t_1$ in Figure 4a map to anti-core regions in the star unfolding because each of these faces can be associated with one maximal shortest path edge sequence. We now justify why the remaining heavily-shaded points map to the core in the star unfolding. Consider the points p_2, p_3 in the interior of the face $t_2 a b_2$ in Figure 4a. The shortest path $\pi(s, p_2)$ travels from s to a to p_2 while $\pi(s, p_3)$ travels from s to edge $\overline{t_2 b_2}$ to p_3. These two unique shortest path edge sequences imply that the face $t_2 a b_2$ must lie in the core. The remaining heavily-shaded faces lie in the core by analogous reasoning. Figure 4b illustrates that faces in the core such as $b_6 a t_6$ and $b_2 a t_2$ partially overlap when the core is unfolded into the plane. This follows because the sum of the eleven angles incident to a is greater than 2π. Thus, the core of the star unfolding for a non-convex polyhedral surface can overlap itself. □

Even though the star unfolding of a non-convex polyhedral surface can overlap itself, the star unfolding can still be defined by an angularly ordered set of non-crossing shortest path cuts from the source to every vertex [15, 35], and a core can still be defined by a polygonal equator with $O(M)$ complexity that connects adjacent endpoints in the ordering.

Since a shortest path can only turn at vertices in the star unfolding [35], an anti-core region must have an *hourglass* shape [27, 28, 18]. This follows because an anti-core region is bounded by a point or line segment source, a line segment, and two shortest paths in the unfolded plane. For example, an hourglass in Figure 5 is bounded by l_1, $\overline{v_2 v_{15}}$, and shortest paths from the endpoints of l_1 to v_2 and v_{15}. Another hourglass is bounded by v'_{10}, $\overline{v_{12} v_{14}}$, $\pi(v'_{10}, v_{12})$, and $\pi(v'_{10}, v_{14})$.[3] Such hourglasses encode all shortest paths from a source to any point in its anti-core region. The free space defined by an hourglass is a connected, piecewise hyperbolic shape with $O(M)$ complexity by [18]. The below $\delta_N(A, B)$ *decision problem* decides whether the Fréchet distance between polygonal curves A and B on a non-convex polyhedral surface is at most some given constant $\varepsilon \geq 0$.

[3]The hourglass containing v'_{10} is actually a simplified type of hourglass called a "funnel" [27, 28, 18].

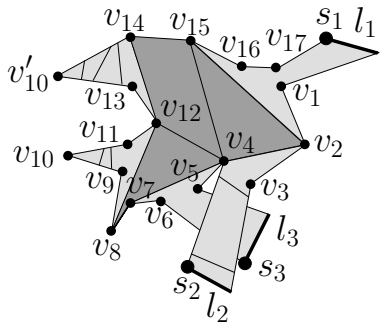

Figure 5: In general, the star unfolding for a *non-convex* polyhedral surface can overlap itself but still has a heavily-shaded core and a lightly-shaded anti-core. Each of the anti-core regions is an hourglass.

Theorem 7. *The $\delta_N(A, B)$ decision problem can be computed in $O(M^7 \log M)$ time and $O(M^3)$ space.*

Proof. Partition the polygonal curve A into $O(M^3)$ edgelets and maintain a star unfolding for these edgelets in $O(M^4)$ total time by Theorem 1. Let C be the constant complexity parameter space for an edgelet and either a core edge or an anti-core edge. Free space for an *anti-core edge* is the intersection of C with the $O(M)$ complexity free space for *one* hourglass. Free space for a *core edge* is the intersection of C with the union of the free spaces for $O(M)$ hourglasses. This union has $O(M^3)$ complexity because the free space for any pair of hourglasses intersects $O(M)$ times. Since each core edge is a chord of the core, the dual graph of the core is a tree. Consequently, the $O(M)$ hourglasses for a core edge γ can be defined by iteratively extending an hourglass from every vertex in the star unfolding [15] through the dual graph of the core to γ.

In total, the free space for each edgelet has $O(M^4)$ complexity because it involves $O(M^2)$ anti-core edges and $O(M)$ core edges, and this free space can be computed in $O(M^4 \log M)$ time [2]. Thus, each of the $O(M^3)$ edgelets contributes $O(M^4)$ complexity to the free space diagram. A plane sweep [19] can be used to answer the decision problem in $O(M^7 \log M)$ time. By storing one star unfolding, one cell, and one vertical line segment of the free space diagram at a time, $O(M^3)$ space is sufficient. □

Critical values [5] are candidate values of ε that are caused by a geometric configuration change of the free space. The smallest critical value ε^* that causes the decision problem to return true defines the exact value of the Fréchet distance. The standard approach [5] to find ε^* is to apply parametric search with Cole's [17] sorting trick. An alternative to parametric search is to run the decision problem

once for every bit of accuracy that is desired [44]. In the following theorem, we apply parametric search to a set of parameterized free space vertices.

Theorem 8. *The Fréchet distance can be computed on a convex polyhedral surface in $O(M^6 \log^2 M)$ time and $O(M^2)$ space and on a non-convex polyhedral surface in $O(M^7 \log^2 M)$ time and $O(M^3)$ space, where M is the total complexity of the surface and the polygonal curves A, B. Furthermore, the free space diagram for a non-convex polyhedral surface can have $\Omega(M^4)$ complexity.*

Proof. Represent each of the $O(M^6)$ (resp. $O(M^7)$) free space vertices from Theorems 6 and 7 as a (possibly partially defined) algebraic curve $\rho_i(\varepsilon)$ that has constant degree and description complexity. There are three types of critical values. Type (a) critical values are values of ε such that some $\rho_i(\varepsilon)$ touches a corner of the free space diagram. Type (b) critical values occur when two $\rho_i(\varepsilon)$ intersect or when free space becomes tangent to a cell boundary. Monotonicity-enforcing type (c) critical values occur when a pair of intersection points lie on a horizontal/vertical line.

Given the $O(M^6 \log M)$ (resp. $O(M^7 \log M)$) decision problem runtimes from Theorems 6 and 7, parametric search with Cole's [17] sorting trick can be applied to the $\rho_i(\varepsilon)$ functions to compute the Fréchet optimization problem in $O(M^6 \log^2 M)$ (resp. $O(M^7 \log^2 M)$) time. The space requirements are identical to the decision problems.

We now show that the free space diagram for a non-convex polyhedral surface can have $\Omega(M^4)$ complexity. Consider the parameter space C defined by $\overline{ab} \in A$ and $\overline{cd} \in B$. Figure 6 illustrates a two-dimensional situation where $\Omega(M^2)$ points in C have the same shortest path distance, and all other distances in C are larger. Hence, for an appropriate choice of ε, the free space in C is defined by $\Omega(M^2)$ disjoint regions. Define the polygonal curves by their vertices as $A = \{a, b, a, b, ..., a, b\}$ and $B = \{c, d, c, d, ..., c, d\}$. Since the $\frac{M^2}{4}$ parameter spaces defined by $\overline{ab}$ and $\overline{cd}$ can each have $\Omega(M^2)$ complexity, the free space diagram can have $\Omega(M^4)$ complexity. □

3.4. Shortest Path Maps

This section develops shortest path maps on convex and non-convex polyhedral surfaces. Throughout this section, M denotes the complexity of either a convex or non-convex polyhedral surface.

Theorem 9. *A shortest path map SPM($\overline{ab}$,$\mathcal{P}$) can be built for a* convex *polyhedral surface $\mathcal{P}$ in $O(M^4 2^{\alpha(M)} \log M)$ time and $O(M^4 2^{\alpha(M)})$ space. For all points*

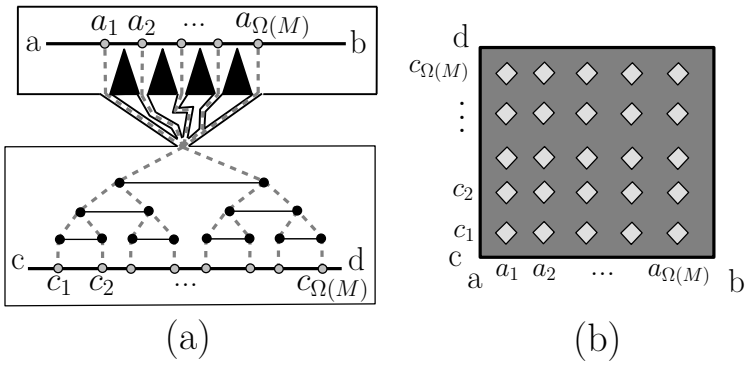

Figure 6: (a) A non-convex polyhedral surface can define a (b) parameter space with $\Omega(M^2)$ complexity. Dashed lines depict shortest paths.

$s \in \overline{ab} \in \mathcal{P}$ and $t \in \mathcal{P}$, SPM($\overline{ab}$,$\mathcal{P}$) can return $d(s,t)$ in $O(\log^2 M)$ time and $\pi(s,t)$ in $O(\log^2 M + K)$ time, where K is the complexity of any returned path.[4]

Proof. The line segment $\overline{ab}$ can be partitioned into $O(M^2)$ edgelets, and a kinetic Voronoi diagram can be maintained for these edgelets in $O(M^4 2^{\alpha(M)} \log M)$ total time and $O(M^4 2^{\alpha(M)})$ space by Theorem 3. Point location queries in this kinetic Voronoi diagram take $O(\log^2 M)$ time by [24]. □

Our next theorem uses the star unfolding [1] and the hourglass structure of [27] to encode all shortest paths between two line segments. Such an hourglass defines a piecewise hyperbolic free space that has $O(M)$ complexity [18].

Theorem 10. *A shortest path map SPM($\overline{ab}$,$\mathcal{P}_N$) can be built for a* non-convex *polyhedral surface $\mathcal{P}_N$ in $O(M^{9+\kappa})$ time and $O(M^9)$ space for any constant $\kappa > 0$. For all points $s \in \overline{ab} \in \mathcal{P}_N$ and $t \in \mathcal{P}_N$, SPM($\overline{ab}$,$\mathcal{P}_N$) can return $d(s,t)$ in $O(\log M)$ time and $\pi(s,t)$ in $O(\log M + K)$ time, where K is the complexity any returned path.*

Proof. Let α represent one of the $O(M^2)$ edgelets on $\overline{ab}$ (see section 3.1). A shortest path between a point $s \in \alpha$ and any fixed point in the *anti-core* can be resolved using one hourglass (cf. section 3.3). Distance and shortest path queries in this hourglass can be resolved in the desired query times by [28, 27]. By contrast, all shortest paths between $s \in \alpha$ and a fixed point in a face of the *core* are defined by $O(M)$ hourglasses (see section 3.3). To support logarithmic query time for

[4] $O(\log M)$ time queries are also possible by [24] but at the cost of essentially squaring both the time and space preprocessing bounds.

all points in a fixed face of the core, we can form $O(M^3)$ constant complexity distance functions from these hourglasses and compute their lower envelope and a vertical decomposition structure in $O(M^{6+\kappa})$ time and $O(M^6)$ space, for any constant $\kappa > 0$ [2]. Repeating this procedure for all $O(M)$ faces in the core yields $O(M^{7+\kappa})$ time per edgelet and $O(M^{9+\kappa})$ time over all $O(M^2)$ edgelets. □

4. Link Distance Problems on a Polyhedral Surface

This section investigates link distance problems on two kinds of polyhedral surfaces. We define $\mathcal{C}$ as a special type of polyhedral surface such that every face of $\mathcal{C}$ is convex and no two adjacent faces are coplanar. We define $\mathcal{N}$ as a special type of polyhedral surface such that no two adjacent faces are coplanar. $\mathcal{N}$ can be created from any polyhedral surface by triangulating the surface and merging adjacent triangles that are coplanar. The essential property of both $\mathcal{C}$ and $\mathcal{N}$ is that a path must turn whenever it enters a new face. The difference between $\mathcal{C}$ and $\mathcal{N}$ is that the faces of $\mathcal{C}$ are convex while the faces of $\mathcal{N}$ need not be convex and can contain holes. Throughout this section, we assume that s, t are points and $\overline{ab}, \overline{cd}$ are line segments on either $\mathcal{C}$ or $\mathcal{N}$. The link distance between s and t is denoted by $d_L(s,t)$, and $\pi_L(s,t)$ denotes a link distance path between s and t. We define M as the complexity of $\mathcal{C}$ and $\mathcal{N}$.

4.1. Shortest Path Maps, Voronoi Diagrams, and Diameters

This section explores link-based shortest path maps, Voronoi diagrams, and diameters on $\mathcal{C}$ and $\mathcal{N}$. All of the shortest path maps in this section support link-based queries in $O(\log M + K)$ time, where K is the complexity of any returned path.

Since each face of $\mathcal{C}$ is convex, link-based paths only turn in $\mathcal{C}$ when they enter a new face. Hence, a breadth-first search is sufficient to mark the link distance of each face to a nearest source. Such a breadth-first search can be used to obtain the following three results.

Corollary 1. *A link-based shortest path map SPM($\overline{ab}$, $\mathcal{C}$) can be constructed on $\mathcal{C}$ in $\Theta(M)$ time and space. This structure supports $d_L(s,t)$, $\pi_L(s,t)$ queries from any $s \in \overline{ab}$ to any $t \in \mathcal{C}$.*

Corollary 2. *A link-based Voronoi diagram for M point or line segment sites can be computed on $\mathcal{C}$ in $\Theta(M)$ time and space provided that the line segment sites are interior-disjoint.*[5]

[5]This runtime requires that each site has previously been associated with the face that contains

Corollary 3. *The link diameter of $\mathcal{C}$ can be computed in $O(M^2)$ time and $O(M)$ space.*

Note that the above Voronoi diagram's line segment sites are interior-disjoint merely to avoid having $\Omega(M^2)$ intersection points. The link diameter of $\mathcal{C}$ is the largest link distance between any pair of faces. We now investigate data structures on $\mathcal{N}$.

Theorem 11. *Link-based shortest path maps SPM(s, $\mathcal{N}$) and SPM($\min_{s\in\overline{ab}}$, $\mathcal{N}$) can be constructed on $\mathcal{N}$ in $\Theta(M^4)$ time and space. SPM(s, $\mathcal{N}$) supports queries from a fixed source point s to any point $t \in \mathcal{N}$. SPM($\min_{s\in\overline{ab}}$, $\mathcal{N}$) supports queries $\min_{s\in\overline{ab}} d_L(s,t)$ and $\min_{s\in\overline{ab}} \pi_L(s,t)$ from a fixed line segment $\overline{ab} \in \mathcal{N}$ to any point $t \in \mathcal{N}$.* Implicit *representations of SPM(s, $\mathcal{N}$) and SPM($\min_{s\in\overline{ab}}$, $\mathcal{N}$) can be built in $O(M^{\frac{7}{3}} \log^{3.11} M)$ time and $O(M)$ space to support exact queries in $O(M)$ time and queries accurate to within one link in $O(\log M + K)$ time.*

Proof. Each face of $\mathcal{N}$ is a *polygonal domain* (a two-dimensional polygon with polygonal holes). Mitchell, Rote, and Woeginger [36] compute a shortest path map in a polygonal domain by iteratively constructing the set of points at link distance $1, 2, ..., M$ from the source in $\Theta(M^4)$ [43] time and space. This approach can be modified to work on the surface $\mathcal{N}$ by propagating link distance paths into adjacent faces at each step. The technique of [36] also supports exact $O(M)$ time queries and approximate queries to within one link of optimal in $O(\log M)$ time. □

Notice that SPM($\min_{s\in\overline{ab}}$, $\mathcal{N}$) only supports $\min_{s\in\overline{ab}} d_L(s,t)$ queries. As an alternative, we can also construct a shortest path map SPM($\overline{ab}, \overline{cd}$) that supports all possible queries between any $s \in \overline{ab} \in \mathcal{N}$ and $t \in \overline{cd} \in \mathcal{N}$. The below observation and lemma will be useful to construct SPM($\overline{ab}, \overline{cd}$).

Observe that all link distances between a pair of line segments $\overline{ab}$, $\overline{cd}$ on any polyhedral surface must equal i, $i+1$, or $i+2$, for some fixed integer $i \geq 0$. This is true because any link distance path between $s \in \overline{ab}$ and $t \in \overline{cd}$ can be extended by at most two extra links into a path that connects any $s' \in \overline{ab}$ and $t' \in \overline{cd}$. See Figure 7.

Observation 1. *All link distances $d_L(s,t)$ for any points $s \in \overline{ab} \in \mathcal{N}$ and $t \in \overline{cd} \in \mathcal{N}$ equal either i, $i+1$, or $i+2$, where $i = \min_{s\in\overline{ab},\ t\in\overline{cd}} d_L(s,t)$.*

The below lemma efficiently computes the intersection of a line segment with our $\Theta(M^4)$ complexity shortest path maps from Theorem 11.

---

it. When this is not the case, either point location techniques [22] or a brute force approach may be used to associate each site with the face that contains it.

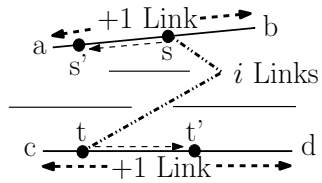

Figure 7: All distances $d_L(s,t)$ for $s \in \overline{ab} \in \mathcal{N}$ and $t \in \overline{cd} \in \mathcal{N}$ equal either i, $i+1$, or $i+2$.

Lemma 3. *The intersection of a line segment $\overline{cd}$ with the $\Theta(M^4)$ complexity shortest path maps SPM(s, $\mathcal{N}$), SPM($\min_{s\in\overline{ab}}$, $\mathcal{N}$) can be constructed in $O(M^2\alpha(M)\log^2 M)$ time and $O(M^2)$ space (without precomputing either shortest path map).*

Proof. By Observation 1, all link distances between $\overline{ab}$ and $\overline{cd}$ equal either i, $i+1$, or $i+2$, where $i = \min_{s\in\overline{ab},\ t\in\overline{cd}} d_L(s,t)$. Suri and O'Rourke [43] have demonstrated that all points with link distance i, $i+1$, or $i+2$ from $\overline{ab}$ can be represented by the union of $O(M^2)$ triangles. These triangles intersect $\overline{cd}$ in $O(M^2)$ intervals that can be constructed in $O(M^2\alpha(M)\log^2 M)$ time and $O(M^2)$ space by modifying the algorithm of [36] so that it propagates link distance paths into adjacent faces of the surface $\mathcal{N}$ at each step. □

We use the following terminology of Arkin, Mitchell, and Suri [7] to construct SPM($\overline{ab}, \overline{cd}$). The combinatorial type of a (link-based) shortest path map is a listing of the combinatorial types of its edges. The combinatorial type of a shortest path map *edge* $\mathcal{E}$ is a vertex-edge pair (v,e) such that $\mathcal{E}$ has one endpoint at a vertex v and has its other endpoint on an edge e. As the source point s varies continuously along $\overline{ab}$, the position of $\mathcal{E}$'s endpoint on e is parameterized homographically by $g(s) = \frac{A+Bs}{C+Ds}$ for constants A, B, C, D. We also define an *edgelet* α as a line segment such that the shortest path map for every source point $s \in \alpha$ has the same combinatorial structure.[6]

Theorem 12. *The link-based shortest path map SPM($\overline{ab}, \overline{cd}$) can have $\Omega(M^4)$ complexity and can be constructed on $\mathcal{N}$ in $O(M^7)$ space and either $O(M^7)$ expected time or $O(M^6\lambda_6(M))$ deterministic time. The SPM($\overline{ab}, \overline{cd}$) structure supports $d_L(s,t)$, $\pi_L(s,t)$ queries for any $s \in \overline{ab} \in \mathcal{N}$ and $t \in \overline{cd} \in N$ in $O(\log M + K)$ time, where K is the complexity of any returned path.*

Proof. SPM($\overline{ab}, \overline{cd}$) is a partition of the parameter space defined by $\overline{ab}$ and $\overline{cd}$ into maximal regions such that for all points (s,t) in a region the shortest path $\pi_L(s,t)$

[6] Arkin, Mitchell, and Suri [7] refer to an edgelet as an *atomic segment*.

has the same combinatorial structure. Edgelet endpoints are defined by positions $s \in \overline{ab}$ where there are at least two distinct shortest paths $\pi_L(s, v)$ to some vertex $v \in \mathcal{N}$ [7]. A total of $O(M^3)$ edgelet endpoints can be defined by computing $\overline{ab}$ $\cap$ SPM(v, $\mathcal{N}$) for every vertex $v \in \mathcal{N}$ in $O(M \cdot M^2 \alpha(M) \log^2 M)$ total time and $O(M^3)$ space (see Lemma 3).

For each edgelet $\alpha \in \overline{ab}$, choose a point $s \in \alpha$, and construct SPM(s, $\mathcal{N}$). A shortest path map edge $\mathcal{E} \in$ SPM(s, $\mathcal{N}$) is a vertex-edge pair (v, e) that is homographically parameterized by $s \in \alpha$ such that $\mathcal{E} \cap \overline{cd}$ defines a constant complexity algebraic curve in the parameter space for $\overline{ab}$ and $\overline{cd}$. Constructing such a curve for each choice of v and e yields $O(M^2)$ curves whose arrangement can be constructed in $O(M^4)$ space and $O(M^4)$ expected time or $O(M^3 \lambda_6(M))$ deterministic time [2]. Since there are $O(M^3)$ edgelets, SPM($\overline{ab}$,$\overline{cd}$) has $O(M^3 \cdot M^4)$ complexity.

Figure 8 illustrates that SPM($\overline{ab}, \overline{cd}$) can have $\Omega(M^4)$ complexity. In Figure 8a, $\overline{ab}$ and $\overline{cd}$ are enclosed in rectangles with tiny openings, and $\Omega(M^2)$ line of sight edges can be drawn between pairs of these openings. Let the intersections of these $\Omega(M^2)$ edges with $\overline{ab}$ be $a_1, ..., a_{\Omega(M^2)}$ and the intersections with $\overline{cd}$ be $c_1, ..., c_{\Omega(M^2)}$. The points $a_1, ..., a_{\Omega(M^2)}$ and $c_1, ..., c_{\Omega(M^2)}$ define a grid of $\Omega(M^2)$ line segments in SPM($\overline{ab}, \overline{cd}$) such that all link distances on these line segments are at most two, and all other link distances are three (see Figure 8b). Thus, SPM($\overline{ab}, \overline{cd}$) can have $\Omega(M^4)$ complexity. □

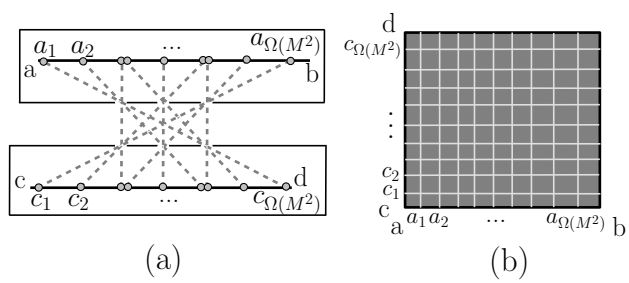

Figure 8: (a) Two line segments on $\mathcal{N}$ can define (b) a link-based shortest path map SPM($\overline{ab}$,$\overline{cd}$) with $\Omega(M^4)$ complexity.

Corollary 4. *The link diameter of $\mathcal{N}$ can be computed in $O(M^{\frac{19}{3}} \log^{3.11} M)$ time and $O(M^3)$ space.*

Proof. Partition each of the $O(M)$ edges of $\mathcal{N}$ into $O(M^3)$ edgelets with Lemma 3. Choose a point s in each of these $O(M^4)$ edgelets. For each of these points, construct the $O(M^2)$ possible (v, e) edges of SPM(s, $\mathcal{N}$) without computing the arrangement of these edges. This can be done in $O(M^{\frac{7}{3}} \log^{3.11} M)$ time per edgelet

by the techniques of [36] and Theorem 11. The largest link distance defined by any of these edges is the link diameter for $\mathcal{N}$. □

The next theorem develops a link-based Voronoi diagram for $\mathcal{N}$.

Theorem 13. *A link-based Voronoi diagram for M point or line segment sites can be computed on $\mathcal{N}$ in $O(M^6)$ time and space and can have $\Omega(M^4)$ complexity.*

Proof. For each point site s on $\mathcal{N}$, construct the $O(M^2)$ possible (v, e) edges that define SPM(s, $\mathcal{N}$) using the techniques of [36] and Theorem 11. Similarly, construct the $O(M^2)$ possible (v, e) edges that define SPM($\min_{s \in \overline{ab}}$, $\mathcal{N}$) for each line segment site $\overline{ab}$ on $\mathcal{N}$. Since there are M sites, this procedure defines $O(M^3)$ edges on $\mathcal{N}$. The arrangement of these edges can be constructed with an output-sensitive algorithm [14] in $O(M^6)$ time and space. Since some edges in this arrangement should not appear in the final Voronoi diagram, a breadth-first postprocessing step should merge adjacent faces with link distance i that are separated by a suboptimal edge with link distance $j > i$. The $\Omega(M^4)$ lower bound follows from the lower bound for SPM(s, $\mathcal{N}$). □

4.2. Similarity Metrics

This section contains link-based algorithms to compute the Hausdorff distance and Fréchet distance on the polyhedral surfaces $\mathcal{C}$ and $\mathcal{N}$. These surfaces were defined at the beginning of section 4.

Theorem 14. *The link-based Hausdorff distance can be computed between two sets of M interior-disjoint line segments on $\mathcal{C}$ in $O(M \log M)$ time and $O(M)$ space.*

Proof. Corollary 2 can be used to build a Voronoi diagram for the line segments in $\Theta(M)$ time and space. Once this Voronoi diagram is known, the Hausdorff distance can be computed in $O(M \log M)$ time and $O(M)$ space by [4]. □

Theorem 15. *The link-based Hausdorff distance can be computed between two sets of M points on $\mathcal{N}$ in $O(M^{\frac{10}{3}} \log^{3.11} M)$ time and $O(M^2)$ space and between two sets of M line segments on $\mathcal{N}$ in $O(M^4 \alpha(M) \log^2 M)$ time and $O(M^3)$ space. In addition, the Hausdorff distance can be approximated to within two links of optimal for line segments on $\mathcal{N}$ in $O(M^{\frac{10}{3}} \log^{3.11} M)$ time and $O(M^2)$ space.*

Proof. For point sets on $\mathcal{N}$, Theorem 11 can be used to construct an implicit shortest path map for every point. These shortest path maps are sufficient to return the Hausdorff distance.

Line segment sets $A, B \in \mathcal{N}$ can be handled as follows. Intersecting a fixed line segment $\overline{ab} \in A$ with SPM($\min_{t \in \overline{cd}}$, $\mathcal{N}$) for every $\overline{cd} \in B$ produces a set of piecewise-constant distance functions. The lower envelope of these functions defines for every point $s \in \overline{ab}$ the nearest neighbor distance $\min_{t \in B} d_L(s, t)$. Such a lower envelope can be computed for each $\overline{ab} \in A$ in $O(M^3 \alpha(M) \log^2 M)$ time and $O(M^3)$ space by combining Lemma 3 with a plane sweep [20] technique. Repeating this step for each $\overline{ab} \in A$ and returning the largest distance on any lower envelope yields the Hausdorff distance $\delta_H(A, B)$ in $O(M^4 \alpha(M) \log^2 M)$ time. The Hausdorff distance can also be approximated to within two links of optimal for line segment sets on $\mathcal{N}$ by computing the Hausdorff distance of the *endpoints* of the line segments (see Observation 1). □

The below theorem shows how to compute the link-based Fréchet distance on $\mathcal{C}$ a logarithmic factor *faster* than the traditional Fréchet distance in the plane [5].

Theorem 16. *The link-based Fréchet distance between two polygonal curves $A, B \in \mathcal{C}$ can be computed in $O(M^2)$ time and space, and the free space diagram can have $\Omega(M^2)$ complexity. The total complexity of A, B, and $\mathcal{C}$ is M.*

Proof. Although the free space in a cell defined by $\overline{ab} \in A$ and $\overline{cd} \in B$ can be disconnected, we show that reachability information can always be propagated through a cell solely based on the cell's boundary. In the simplest case, $\overline{ab}$ and $\overline{cd}$ are in the same convex face of $\mathcal{C}$, and all link distances in the cell are either zero (at any intersection points of $\overline{ab}$ and $\overline{cd}$) or one. When $\overline{ab}$ and $\overline{cd}$ are in separate faces, all points in the *interior* of a cell have the same link distance; furthermore, all link distances on the *interior* of a boundary edge are the same (see Figure 9). This behavior ensures that reachability information can always be propagated through a cell in constant time solely based on the cell's boundary.

The distances defining all cell boundaries of the free space diagram can be computed in $\Theta(M^2)$ time using shortest path map structures from Corollary 1. The Fréchet distance can be computed by building a planar graph on the free space diagram and applying a linear-time shortest path algorithm (e.g., [29]). □

We now use the shortest path map SPM($\overline{ab}, \overline{cd}$) from Theorem 12 to compute the link-based Fréchet distance on $\mathcal{N}$. The Fréchet *decision problem* [5] decides whether the Fréchet distance $\delta_F(A, B) \leq \varepsilon$ for some constant $\varepsilon \geq 0$.

Theorem 17. *The link-based Fréchet distance between two polygonal curves $A, B \in \mathcal{N}$ can be computed* exactly *in $O(M^9 \log M)$ time and $O(M^4)$ space, and the free space diagram can have $\Omega(M^6)$ complexity. The Fréchet distance can also be approximated to within one link of optimal in $O(M^4 \alpha(M) \log^2 M)$ time or to within*

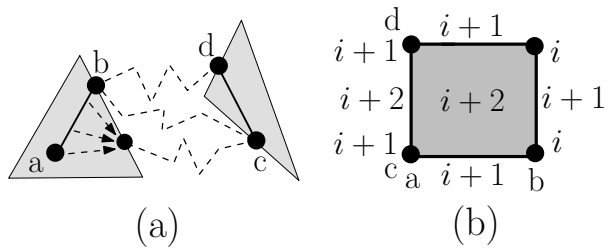

Figure 9: (a) When $\overline{ab}$ and $\overline{cd}$ are in separate faces of $\mathcal{C}$, all distances in the interior of a cell defined by $\overline{ab}$ and $\overline{cd}$ have the same value. (b) Free space in a cell can be disconnected by choosing $\varepsilon = i$ so that all link distances less than or equal to i define the free space.

two links of optimal in $O(M^{\frac{10}{3}} \log^{3.11} M)$ time. Both approximations use $O(M^2)$ space.

Proof. Let $i = \min_{s \in \overline{ab},\, t \in \overline{cd}} d_L(s,t)$. To approximate the Fréchet distance to within one link of optimal, determine the representative value $i+1$ for each of the $O(M^2)$ cells defined by $\overline{ab} \in A$ and $\overline{cd} \in B$ by computing SPM($\min_{t \in \overline{cd}}$, $\mathcal{N}$) $\cap\, \overline{ab}$. This intersection defines for every $s \in \overline{ab}$ the distance $\min_{t \in \overline{cd}} d_L(s,t)$, and the smallest of these distances equals i. Repeating this process for each of the $O(M^2)$ cells in the free space diagram takes $O(M^2 \cdot M^2 \alpha(M) \log^2 M)$ total time and $O(M^2)$ working space by Lemma 3. The approximation to within two links of optimal uses the *implicit* shortest path map from Theorem 11 to compute any link distance i, $i+1$, or $i+2$ to represent a cell (cf. Observation 1). Once a representative value is known for each cell, the approximate Fréchet distance can be computed in $O(M^2)$ time by constructing a directed acyclic graph on the free space diagram and performing a breadth first search.

The exact decision problem is computed by representing each of the $O(M^2)$ cells defined by $\overline{ab} \in A$ and $\overline{cd} \in B$ by the SPM($\overline{ab}$,$\overline{cd}$) structure of Theorem 12. The $O(M^9)$ faces in these shortest path maps form a partition of the free space diagram, and each face has the same link distance throughout its interior. The decision problem can be computed in $O(M^9 \log M)$ time by combining dynamic programming [5] with a plane sweep [20] that propagates reachability information through each cell. $O(M^4)$ storage is sufficient to store one edgelet of SPM($\overline{ab}$,$\overline{cd}$) at a time (see Theorem 12).

After executing either of the above approximation algorithms, the Fréchet distance is known to within $O(1)$ links of its true value. Thus, the exact decision problem can be executed $O(1)$ times to return the exact Fréchet distance in $O(M^9 \log M)$ time. The $\Omega(M^6)$ lower bound for the complexity of the free space diagram follows because each cell can have $\Omega(M^4)$ complexity by Theorem 12. $\square$

5. Conclusion

We develop algorithms to compute edge sequences, Voronoi diagrams, shortest path maps, the Fréchet distance, and the diameter for a polyhedral surface. Despite efforts by Chandru et al. [13] to improve edge sequence algorithms, these runtimes had not improved since 1997. Our work speeds up the edge sequence and diameter approaches of Agarwal et al. [1] by a linear factor and introduces many new shortest path and link distance algorithms that apply to both convex and non-convex polyhedral surfaces. It would be interesting to lower the gaps between our various lower and upper bounds. In particular, future work could attempt to construct the $\Theta(M^4)$ shortest path edge sequences on a convex polyhedral surface in $o(M^5)$ time.

References

[1] P. K. Agarwal, B. Aronov, J. O'Rourke, and C. A. Schevon. Star unfolding of a polytope with applications. *SIAM Journal on Computing*, 26(6):1689–1713, 1997.

[2] P. K. Agarwal and M. Sharir. *Davenport–Schinzel Sequences and Their Geometric Applications*, pages 1–47. Handbook of Computational Geometry, Elsevier, 2000.

[3] G. Albers, J. S. B. Mitchell, L. J. Guibas, and T. Roos. Voronoi diagrams of moving points. *International Journal of Computational Geometry and Applications*, 8:365–380, 1998.

[4] H. Alt, P. Braß, M. Godau, C. Knauer, and C. Wenk. Computing the Hausdorff distance of geometric patterns and shapes. *Discrete and Computational Geometry*, 25:65–76, 2003. Special Issue: The Goodman-Pollack-Festschrift (B. Aronov, S. Basu, J. Pach, M. Sharir eds.).

[5] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 5:75–91, 1995.

[6] H. Alt, C. Knauer, and C. Wenk. Comparison of distance measures for planar curves. *Algorithmica*, 38(1):45–58, 2003.

[7] E. M. Arkin, J. S. B. Mitchell, and S. Suri. Optimal link path queries in a simple polygon. *3rd Symposium on Discrete Algorithms (SODA)*, pages 269–279, 1992.

[8] B. Aronov, S. Har-Peled, C. Knauer, Y. Wang, and C. Wenk. Fréchet distance for curves, revisited. *14th Annual European Symposium on Algorithms (ESA)*, pages 52–63, 2006.

[9] B. Aronov and J. O'Rourke. Nonoverlap of the star unfolding. *7th Symposium on Computational Geometry (SoCG)*, pages 105–114, 1991.

[10] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, 1991.

[11] K. Buchin, M. Buchin, and C. Wenk. Computing the Fréchet distance between simple polygons in polynomial time. *22nd Symposium on Computational Geometry (SoCG)*, pages 80–87, 2006.

[12] E. W. Chambers, É. Colin de Verdiére, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Walking your dog in the woods in polynomial time. *24th Symposium on Computational Geometry (SoCG)*, pages 101–109, 2008.

[13] V. Chandru, R. Hariharan, and N. M. Krishnakumar. Short-cuts on star, source and planar unfoldings. *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 174–185, 2004.

[14] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. ACM*, 39(1):1–54, 1992.

[15] J. Chen and Y. Han. Shortest paths on a polyhedron. *International Journal of Computational Geometry and Applications*, 6(2):127–144, 1996.

[16] Y. Chiang and J. S. B. Mitchell. Two-point Euclidean shortest path queries in the plane. *10th Symposium on Discrete Algorithms (SODA)*, pages 215–224, 1999.

[17] R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *J. ACM*, 34(1):200–208, 1987.

[18] A. F. Cook IV and C. Wenk. Geodesic Fréchet distance inside a simple polygon. *25th Symposium on Theoretical Aspects of Computer Science (STACS)*, 2008.

[19] A. F. Cook IV and C. Wenk. Geodesic Fréchet distance with polygonal obstacles. Technical Report CS-TR-2008-010, University of Texas at San Antonio, 2008.

[20] A. F. Cook IV and C. Wenk. Min-link shortest path maps and Fréchet distance. Technical Report CS-TR-2008-011, University of Texas at San Antonio, 2008.

[21] A. F. Cook IV and C. Wenk. Shortest path problems on a polyhedral surface. *25th European Workshop on Computational Geometry (EuroCG)*, 2009.

[22] M. de Berg and O. Schwarzkopf. Cuttings and applications. *International Journal of Computational Geometry and Applications*, 5(4):343–355, 1995.

[23] E. D. Demaine and J. O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, New York, NY, USA, 2007.

[24] O. Devillers, M. Golin, K. Kedem, and S. Schirra. Queries on Voronoi diagrams of moving points. *Computational Geometry: Theory and Applications*, 6(5):315–327, 1996.

[25] A. Efrat, L. J. Guibas, S. Har-Peled, D. C. Lin, J. S. B. Mitchell, and T. M. Murali. Sweeping simple polygons with a chain of guards. *11th Symposium on Discrete Algorithms (SODA)*, pages 927–936, 2000.

[26] A. Efrat, L. J. Guibas, S. Har-Peled, J. S. B. Mitchell, and T. M. Murali. New similarity measures between polylines with applications to morphing and polygon sweeping. *Discrete and Computational Geometry*, 28(4):535–569, 2002.

[27] L. J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *Journal of Computer and System Sciences*, 39(2):126–152, 1989.

[28] L. J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.

[29] M. R. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *Journal of Computer and System Sciences*, 55(1):3–23, 1997.

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

[31] Y.-H. Hwang, R.-C. Chang, and H.-Y. Tu. Finding all shortest path edge sequences on a convex polyhedron. *Workshop on Algorithms and Data Structures (WADS)*, 1989.

[32] A. Maheshwari, J.-R. Sack, and H. N. Djidjev. Link distance problems. *Handbook of Computational Geometry*, 1999.

[33] A. Maheshwari and J. Yi. On computing Fréchet distance of two paths on a convex polyhedron. *21st European Workshop on Computational Geometry (EuroCG)*, 2005.

[34] J. S. B. Mitchell. Geometric shortest paths and network optimization. *Handbook of Computational Geometry*, 1998.

[35] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.

[36] J. S. B. Mitchell, G. Rote, and G. J. Woeginger. Minimum-link paths among obstacles in the plane. *6th Symposium on Computational Geometry (SoCG)*, pages 63–72, 1990.

[37] D. M. Mount. The number of shortest paths on the surface of a polyhedron. *SIAM Journal on Computing*, 19(4):593–611, 1990.

[38] A. Okabe, B. Boots, K. Sugihara, and S. N.Chiu. *Spatial tessellations: Concepts and applications of Voronoi diagrams*. Probability and Statistics. Wiley, NYC, 2nd edition, 2000. 671 pages.

[39] G. Rote. Computing the Fréchet distance between piecewise smooth curves. Technical Report ECG-TR-241108-01, May 2005.

[40] C. Schevon and J. O'Rourke. The number of maximal edge sequences on a convex polytope. *26th Allerton Conference on Communication, Control, and Computing*, pages 49–57, 1988.

[41] Y. Schreiber and M. Sharir. An optimal-time algorithm for shortest paths on a convex polytope in three dimensions. *Discrete & Computational Geometry*, 39(1-3):500–579, 2008.

[42] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. *Computer Vision and Graphical Image Processing (CVGIP)*, 35(1):99–110, July 1986.

[43] S. Suri and J. O'Rourke. Worst-case optimal algorithms for constructing visibility polygons with holes. *2nd Symposium on Computational Geometry (SoCG)*, pages 14–23, 1986.

[44] R. van Oostrum and R. C. Veltkamp. Parametric search made practical. *18th Symposium on Computational Geometry (SoCG)*, pages 1–9, 2002.