GEODESIC FRÉCHET DISTANCE WITH POLYGONAL OBSTACLES

Atlas F. Cook IV[†] Carola Wenk[†]

Abstract

We present the first algorithm to compute the geodesic Fréchet distance between two polygonal curves in a plane with polygonal obstacles, where distances between points are measured as the length of a shortest path between them. Using shortest path structures that we call dynamic and static spotlights, we efficiently construct and propagate reachability information through the free space diagram to solve the Fréchet distance. We also show how to construct a novel shortest path map from a line segment source (instead of from a point source). This shortest path map supports geodesic distance queries from any point $s \in \overline{ab}$ to any point $t \in \overline{cd}$ in optimal logarithmic time and permits the shortest path to be reported in output-sensitive fashion.

1. INTRODUCTION

The comparison of geometric shapes is essential in various applications including computer vision, computer aided design, robotics, medical imaging, and drug design. The Fréchet distance is a similarity metric for continuous shapes that is defined using reparametrizations of the shapes. Since it takes the continuity of the shapes into account, it is generally a more appropriate distance measure than the often used Hausdorff distance.

Most previous work assumes an obstacle-free environment where distances between points are measured using an L_p metric. In [3] the Fréchet distance between polygonal curves A and B is computed in arbitrary dimensions for obstacle-free environments in $O(N^2 \log N)$ time, where N is the larger of the complexities of A and B. Rote [19] computes the Fréchet distance between piecewise smooth curves. Buchin et al. [8] show how to compute the Fréchet distance between two simple polygons. Wenk et al. [21] apply the Fréchet distance to the practical domain of map matching. All of these works measure distances between points using an L_p metric.

Recent work has focused on measuring distances between points using shortest paths that avoid a set of obstacles. Maheshwari and Yi [16] show how to compute the geodesic Fréchet distance for polygonal curves A and B on the surface of a convex polyhedron. Efrat et al. [13] apply the geodesic Fréchet distance to morphing by considering the polygonal curves A and B to be obstacles. In [12] the geodesic Fréchet distance is measured for polygonal curves A and B inside a simple polygon obstacle P.

We show how to compute the geodesic Fréchet distance between polygonal curves A and B in a plane filled with polygonal obstacles. Our approach requires $O(N^2k^4 \log k \log Nk)$ expected time and $O(Nk + k^4)$ space, where N is the complexity of A and B, and k is the complexity of the polygonal obstacles. The motivation for this study is that Euclidean geodesics are a natural distance measure for complex environments such as simple polygons and polygonal domains (polygons with holes). Like [12], distances between points are measured by shortest paths. Unlike the $O(N^4k^3 \log kN)$ time Fréchet distance algorithm of [9], we measure distances between points by shortest paths without enforcing any specific homotopy class.

A byproduct of this investigation is the ability to construct a shortest path map between two line segments. Although a traditional shortest path map supports shortest path queries from a fixed point, this structure cannot be used to compute shortest paths from a *continuous* set of source points $s \in \overline{ab}$. Our shortest path

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

[†]Department of Computer Science, University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249-0667. {acook,carola}@cs.utsa.edu

map structure remedies this problem. A related result by Chiang and Mitchell [10] constructs a two-point shortest path map from any source point in the plane to any target point in the plane in $O(k^{11})$ time and space, where k is the total complexity of the polygonal obstacles. Our work differs in that we restrict the source and target points to lie on line segments instead of being allowed to lie anywhere in the plane. This restriction allows us to achieve optimal $O(\log k)$ query time after preprocessing in $O(k^5 \log k)$ expected time and $O(k^5)$ space.

A core insight of this paper is that shortest path structures called dynamic and static spotlights (see section 3) represent all possible shortest paths between any point $s \in \overline{ab}$ and any point $t \in \overline{cd}$. We use these spotlights not only to compute the geodesic Fréchet distance in a plane with polygonal obstacles (section 4) but also to compute a shortest path map (section 5) that can answer shortest path queries between any $s \in \overline{ab}$ and $t \in \overline{cd}$ in logarithmic time and can report the shortest path in an output-sensitive manner.

2. PRELIMINARIES

The Fréchet distance is often illustrated by a person walking a dog on a leash [3]. The idea is that a person walks forward on one curve, and a dog walks forward on the other curve. As the person and dog move along their respective curves, a leash is maintained to keep track of the separation between them. The maximum separation attained during the walk defines the required leash length, and the length of the *shortest* leash over all possible walks defines the Fréchet distance. A short Fréchet distance (i.e., leash length) means the curves are similar; a large Fréchet distance means the curves are different.

Formally, the Fréchet distance for two curves $A, B : [0,1] \rightarrow \mathbb{R}^l$ is defined as

$$\delta_F(A,B) = \inf_{\alpha,\beta:[0,1]\rightarrow[0,1]} \sup_{t\in[0,1]} d'(\, A(\alpha(t)), B(\beta(t))\,)$$

where α and β range over continuous non-decreasing reparametrizations and d' is a distance metric for points, usually the L_2 distance, and in our setting the geodesic distance. For a given $\varepsilon > 0$ the *free space* is defined as $FS_\varepsilon(A,B) = \{(s,t) \mid d'(A(s),B(t)) \leq \varepsilon\} \subseteq [0,1]^2$. A free space cell $C \subseteq [0,1]^2$ is the parameter space $\overline{ab} \times \overline{cd}$ defined by two line segments $\overline{ab} \in A$ and $\overline{cd} \in B$, and the free space inside the cell is $FS_\varepsilon(\overline{ab},\overline{cd}) = FS_\varepsilon(A,B) \cap C$.

The decision problem to check whether the Fréchet distance is at most a given $\varepsilon > 0$ is solved by Alt and Godau [3] using the *free space diagram* which consists of all free space cells for all pairs of line segments of A and B. Their dynamic programming algorithm checks for the existence of a monotone path in the free space from $(0,0)$ to $(1,1)$ by propagating reachability information cell by cell through the free space. *Reachable space* is the set of free space points that are reachable by a monotone path through the free space that originates at $(0,0)$. A variant of the Fréchet distance that lacks the monotonicity requirement for the path in the free space is called the *weak Fréchet distance* [3].

This paper investigates the geodesic Fréchet distance in a plane containing polygonal obstacles. Let $\mathcal{O} = \{o_1, o_2, ..., o_R\}$ be the union of the $O(k)$ obstacle vertices. In this paper, a geodesic is a globally shortest path that avoids obstacles.[1] Let $\pi(s,t)$ denote a geodesic between the points s and t, and let $d(s,t)$ be the Euclidean length of $\pi(s,t)$. $||s-t||$ denotes the Euclidean distance between s and t, and $\circ$ denotes concatenation; for example, $\pi(s,o_i) \circ t$ is the geodesic from s to o_i concatenated with the line segment from o_i to t. When s has line of sight to t, we write $\pi(s,t) = s \circ t$.

A traditional *shortest path map* is a partition of the plane into a set of faces such that all points in a given face have the same combinatorial shortest path to a fixed source [17]. This means that all points in a given face have the same shortest path except possibly for the start and end points of the path. The traditional fixed-source shortest path map SPM(s) [14, 17] allows queries to be answered from a fixed-source s to any destination in $\mathbb{R}^2$ in logarithmic time and can be stored in $O(k)$ space after $O(k \log k)$ time and space preprocessing. We define a new type of shortest path map SPM($\overline{ab}, \overline{cd}$) that can handle queries from any source point $s \in \overline{ab}$ to any destination point $t \in \overline{cd}$. Throughout this paper, N is the complexity

[1] A *geodesic* is sometimes defined as an obstacle-avoiding path that cannot locally be shortened by slight perturbations [18]. Such a path is not necessarily a globally shortest path.

of two polygonal curves A and B, k is the complexity of a set of polygonal obstacles in the plane, and $f, g : [0, 1] \to \mathbb{R}^2$ are parametrizations of line segments $\overline{ab}$ and $\overline{cd}$ such that $f(s) = (1 - s) \cdot a + s \cdot b$ and $g(t) = (1 - t) \cdot c + t \cdot d$ for all $s, t \in [0, 1]$.

3. Dynamic and Static Spotlights

A shortest path (i.e., a *geodesic*) between points $s \in \overline{ab}$ and $t \in \overline{cd}$ can have two forms. Line of sight geodesics $\pi(s, t) = s \circ t$ are represented by a structure that we call a *dynamic spotlight*. Geodesics with their final turn at an obstacle vertex o_j have the form $\pi(s, o_j) \circ t$ and are represented by a structure that we call a *static spotlight*. Together, the dynamic and static spotlights completely describe all shortest paths from any $s \in \overline{ab}$ to any $t \in \overline{cd}$.

3.1. Dynamic Spotlights.

Geodesics that proceed directly from $s \in \overline{ab}$ to $t \in \overline{cd}$ are represented by a structure that we call a *dynamic spotlight*. Let $\mathcal{I}_{ab}$ be an interval on $\overline{ab}$ and let $\triangle_{s,o_i,o_j}$ be a triangle with apex s, sides supported by $\overline{so_i}$ and $\overline{so_j}$, and base on $\overline{cd}$ (see Figure 3.1a). A dynamic spotlight is defined as $\mathcal{L}_{\mathcal{D}}(\mathcal{I}_{ab}, o_i, o_j) = \{(s, t) \mid \pi(s, t) = s \circ t,\ s \in \mathcal{I}_{ab},\ t \in \Delta_{s,o_i,o_j} \cap \overline{cd},\ \Delta_{s,o_i,o_j}$ contains no obstacles in its interior$\}$.[2] The dynamic spotlight $\mathcal{L}_{\mathcal{D}}$ encodes line of sight information from $s \in \mathcal{I}_{ab}$ to $t \in \Delta_{s,o_i,o_j} \cap \overline{cd}$. As s varies in $\mathcal{I}_{ab}$, the interval $\Delta_{s,o_i,o_j} \cap \overline{cd}$ that is directly visible from s changes continuously and defines $\mathcal{L}_{\mathcal{D}}$ in the parameter space $\overline{ab} \times \overline{cd}$ (see Figure 3.1c).

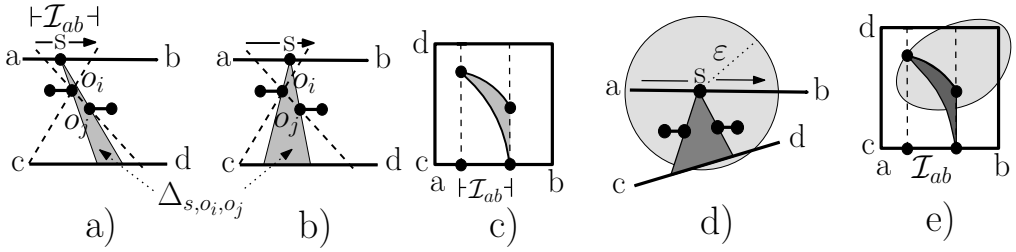

FIGURE 3.1. a,b) A dynamic spotlight $\mathcal{L}_{\mathcal{D}}(\mathcal{I}_{ab}, o_i, o_j)$ encodes line of sight information from $s \in \mathcal{I}_{ab}$ to $t \in \Delta_{s,o_i,o_j} \cap \overline{cd}$. As s varies over $\mathcal{I}_{ab}$, $\Delta_{s,o_i,o_j} \cap \overline{cd}$ changes continuously and defines c) $\mathcal{L}_{\mathcal{D}}$ in the parameter space $\overline{ab} \times \overline{cd}$. d,e) Free space inside $\mathcal{L}_{\mathcal{D}}$ is the intersection of the lightly-shaded free space ellipse with the darkly-shaded dynamic spotlight.

Lemma 1. *The free space defined by the dynamic spotlights for a cell C has complexity $O(k^2)$ and can be constructed in $O(k^2 \log k)$ time and $O(k^2)$ space.*

Proof. We compute $O(k^2)$ disjoint dynamic spotlights in C as follows. For every vertex o_j, perform an angular sweep around o_j to determine a partition of $[0, 2\pi]$ that encodes the angular ordering of vertices that are visible from o_j as well as intervals with line of sight to $\overline{ab}$ or to $\overline{cd}$. Overlaying this partition with a copy shifted by π yields a partition that encodes bidirectional line of sight information along a line through o_j. Each interval in this partition that encodes line of sight to *both* $\overline{ab}$ *and* to $\overline{cd}$ defines a dynamic spotlight. The partition for each o_j has complexity $O(k)$ and can be computed in $O(k \log k)$ time. After computing the $O(k^2)$ dynamic spotlights in C, we intersect each spotlight with the free space ellipse that corresponds to the Euclidean free space between $\overline{ab}$ and $\overline{cd}$ (see Figure 3.1e). □

[2]The endpoints c and d are also candidates for o_i and o_j in this case.

3.2. **Static Spotlights.** Paths from $s \in \overline{ab}$ to $t \in \overline{cd}$ that have their final turn at an obstacle vertex o_j and have the form $\pi(s, o_j) \circ t$ are represented by a structure that we call a *static spotlight* (see Figure 3.2). Let $\mathcal{I}_{ab}$ be an interval on $\overline{ab}$ that results from intersecting SPM(o_j) with $\overline{ab}$ (so that $\pi(s, o_j)$ has the same combinatorial structure for all $s \in \mathcal{I}_{ab}$). SPM(o_j) has $O(k)$ complexity [17], so there are $O(k)$ choices of $\mathcal{I}_{ab}$ for each o_j. Let $\mathcal{I}_{cd}$ be an interval on $\overline{cd}$ with line of sight to o_j. A static spotlight is defined as $\mathcal{L}_{\mathcal{S}}(\mathcal{I}_{ab}, o_j, \mathcal{I}_{cd}) = \{(s,t) \mid s \in \mathcal{I}_{ab},\ t \in \mathcal{I}_{cd},\ \pi(s, o_j) \circ t$ is a path connecting s to t such that o_j has line of sight to $t\}$.

Although dynamic spotlights are disjoint and represent only *shortest* paths, static spotlights from o_i and o_j can overlap and represent candidate paths that are not all necessarily shortest. This follows because a static spotlight for o_j forces a path from s to t to go through o_j even when $\pi(s,t)$ does not have its final turn at o_j. However, every shortest path $\pi(s,t)$ must be represented by at least one dynamic or static spotlight because shortest paths can only turn at obstacle vertices [5, 15]. The free space for $\mathcal{L}_{\mathcal{S}}(\mathcal{I}_{ab}, o_j, \mathcal{I}_{cd})$ satisfies the inequality $d(f(s), o_j) + ||o_j - g(t)|| \leq \varepsilon$ for all $f(s) \in \mathcal{I}_{ab}$ and $g(t) \in \mathcal{I}_{cd}$.

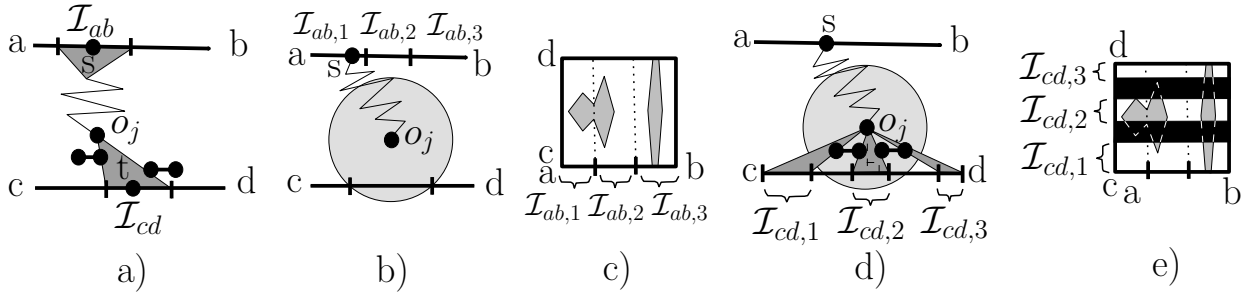

FIGURE 3.2. **a)** Static spotlights represent paths of the form $\pi(s, o_j) \circ t$. **b)** SPM(o_j)$\cap\overline{ab}$ induces $O(k)$ intervals $\mathcal{I}_{ab,1}, ..., \mathcal{I}_{ab,O(k)} \in \overline{ab}$. Since free space satisfies $d(f(s), o_j)+||o_j - g(t)|| \leq \varepsilon$, a circle centered at o_j with radius $\varepsilon - d(f(s), o_j)$ defines the free space for the static spotlight. **c)** In a free space cell, each of the intervals $\mathcal{I}_{ab,1}, ..., \mathcal{I}_{ab,O(k)} \in \overline{ab}$ defines a vertical slab. Within each slab, the free space inequality $d(f(s), o_j) + ||o_j - g(t)|| \leq \varepsilon$ describes a (lightly-shaded) semi-algebraic set of constant degree that is parametrized based on ε. **d)** Line of sight from o_j can be restricted to visible intervals $\mathcal{I}_{cd,1}, \mathcal{I}_{cd,2}, ..., \mathcal{I}_{cd,O(k)} \in \overline{cd}$ by cutting out **e)** horizontal slabs (darkly-shaded) from the semi-algebraic sets in (c).

Lemma 2. *The free space contributed by all $O(k^3)$ static spotlights in a cell can be described by an arrangement of semi-algebraic sets that has $O(k^3)$ complexity and can be constructed in $O(k^3)$ time.*

Proof. There are $O(k^3)$ static spotlights in a cell because there are $O(k)$ choices for each of $\mathcal{I}_{ab},\ \mathcal{I}_{cd},\ o_j$ in $\mathcal{L}_{\mathcal{S}}(\mathcal{I}_{ab}, o_j, \mathcal{I}_{cd})$. For a fixed obstacle vertex o_j, consider the partition of $\overline{ab}$ that is induced by all the intervals $\mathcal{I}_{ab}$ for which there exists a static spotlight $\mathcal{L}_{\mathcal{S}}(\mathcal{I}_{ab}, o_j, \mathcal{I}_{cd})$. This partition induces $O(k)$ vertical slabs in the free space cell because the point-source shortest path map SPM(o_j) intersected with $\overline{ab}$ has $O(k)$ complexity [17]. Within each slab, the free space inequality $d(f(s), o_j) + ||o_j - g(t)|| \leq \varepsilon$ describes a semi-algebraic set of constant degree that is parametrized based on ε (see Figure 3.2c). For two obstacle vertices o_i and o_j, the intersections of all such semi-algebraic sets can be computed by first overlaying the respective vertical slabs of $\overline{ab}$ for o_i and o_j. This overlay has $O(k)$ complexity, and within each slab there is one semi-algebraic set for o_i and one semi-algebraic set for o_j. Given two semi-algebraic sets in a slab, we can compute in constant time all of their $O(1)$ intersection points (note that these intersection points are parametrized based on ε). Hence, over all pairs o_i, o_j and their overlayed vertical slabs there are $O(k^2 \cdot k)$ parametrized intersection vertices that can be computed in $O(k^3)$ total time.

The points described by the semi-algebraic sets, however, are only a superset of the free space. These sets do not yet capture the line of sight information from each o_j onto all maximal intervals $\mathcal{I}_{cd,1}, \mathcal{I}_{cd,2}, ..., \mathcal{I}_{cd,O(k)} \in$

$\overline{cd}$ that are visible from o_j (see Figures 3.2d and 3.2e). The non-free space points can be removed by cutting out a horizontal slab from the semi-algebraic set for each interval on $\overline{cd}$ that is not visible from o_j. For each o_j, there are $O(k)$ such horizontal slabs. Each of the $O(k)$ slabs can intersect each of the $O(k)$ semi-algebraic sets for o_j $O(1)$ times, for a total of $O(k^2)$ vertices per o_j, and $O(k^3)$ vertices in total. The arrangement can be computed using a randomized incremental algorithm [1] in $O(k^3)$ expected time. □

4. GEODESIC FRÉCHET DISTANCE FOR POLYGONAL OBSTACLES

Let δ_G be the geodesic Fréchet distance for two polygonal curves A, B in a plane with polygonal obstacles. Let $\mathcal{F}$ be the free space diagram defined by the parameter space $A \times B$. The *free space* in $\mathcal{F}$ is the set of all points $(s,t) \in \mathcal{F}$ such that $d(s,t) \leq \varepsilon$ (for some $\varepsilon > 0$). The *decision problem* determines whether $\delta_G \leq \varepsilon$ by testing if a *monotone* free space path exists from the lower left corner of $\mathcal{F}$ to the upper right corner of $\mathcal{F}$. To test if such a monotone path exists, dynamic programming is commonly used [3] to propagate reachability information. This reachability information defines the set of points $(s,t) \in \mathcal{F}$ that are reachable by a monotone path from the lower left corner of $\mathcal{F}$. After propagating reachability information, $\delta_G \leq \varepsilon$ if true if and only if the upper right corner of $\mathcal{F}$ is in the set of reachable points. A variant of δ_G called the *weak* geodesic Fréchet distance $\tilde{\delta}_G$ differs in that the path through $\mathcal{F}$ need not be monotone. A key idea of this section is that since dynamic and static spotlights can be used to encode all shortest paths $\pi(s,t)$, these structures can be used to represent the free space in $\mathcal{F}$.

It is well known that a non-geodesic free space cell has a *convex* free space that can be computed in $O(1)$ time [3]. By contrast, the free space in a geodesic cell C is not convex in general because it can have $O(k)$ free space intervals on a boundary edge (see Figure 4.1).

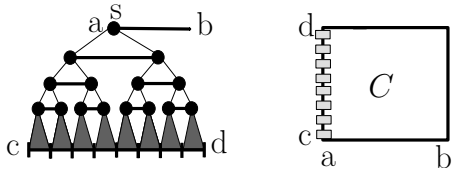

FIGURE 4.1. The Euclidean distance function $d(s, g(t))$ from a fixed point s to $g(t) \in \overline{cd}$ has $O(k)$ minima at perpendiculars from obstacle vertices onto $\overline{cd}$. Hence, the free space cell C can have $O(k)$ free space intervals (lightly shaded) on a boundary edge.

The non-convexity of the free space in C means that we cannot simply compute the boundaries of C to propagate reachability information (as is common practice in the non-geodesic case [3]). Instead, we use the arrangement of the dynamic and static spotlights defined in section 3 to construct the free space inside C. Once this arrangement has been calculated, a planesweep can propagate reachability information through C, and this is sufficient to solve the δ_G decision problem. The δ_G optimization problem is solved using parametric search.

4.1. **Decision Problem.**

The geodesic decision problem can be solved by propagating reachability through the free space. However, the free space in each geodesic cell is not convex in general, so we explicitly construct the free space using the dynamic and static spotlights from sections 3.1 and 3.2.

Lemma 3. *Free space in any cell C for the geodesic Fréchet distance δ_G has complexity $O(k^4)$ and can be constructed in $O(k^4)$ expected time for any $\varepsilon > 0$.*

Proof. We compute the free space in a cell C by overlaying the dynamic spotlight arrangement (Lemma 1) with the static spotlight arrangement (Lemma 2). Within each of the $O(k)$ vertical $\mathcal{I}_{ab}$ slabs for a fixed o_j, each of the $O(k^2)$ dynamic spotlights can intersect the semi-algebraic set for o_j's free space $O(1)$ times, for a total of $O(k \cdot k^3)$ intersections of this type over all o_j. Each of the $O(k^2)$ dynamic spotlights can also intersect each of the $O(k^2)$ line of sight enforcing horizontal slabs $O(1)$ times (see Figure 3.2). Hence, the

arrangement of the dynamic and static spotlights has $O(k^4)$ complexity. The arrangement can be computed using a randomized incremental algorithm (see [1]) in $O(k^4)$ expected time. □

Lemma 4. *Given any $\varepsilon \geq 0$, a free space cell C can have reachability information propagated from its left and bottom boundary edges to its right and top boundary edges in $O(k^4 \log k)$ expected time for the strong geodesic Fréchet distance δ_G and in $O(k^4)$ expected time for the weak geodesic Fréchet distance $\tilde{\delta}_G$. Both approaches use $O(k^4)$ space.*

Proof. By Lemma 3, the free space in a cell has $O(k^4)$ complexity and can be constructed in $O(k^4)$ expected time. For $\tilde{\delta}_G$, reachability information can be propagated by simply marking certain connected components of the free space as reachable.

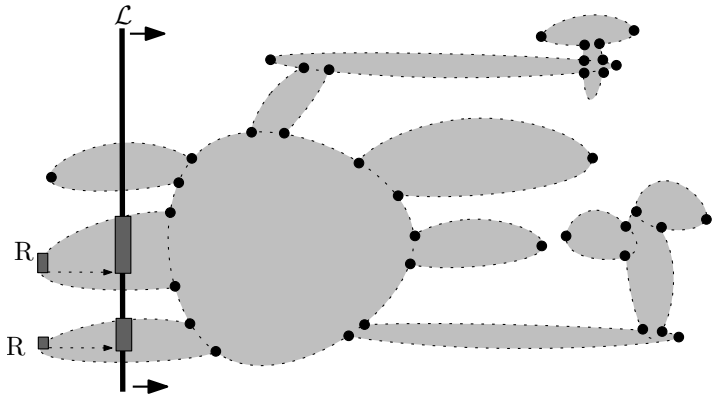

FIGURE 4.2. Reachability information can be propagated through C by a planesweep of a vertical line $\mathcal{L}$. The input reachability points are denoted by R. Free space is lightly shaded. Reachable space on $\mathcal{L}$ is heavily shaded.

For δ_G, path monotonicity must be enforced, and reachability information can be propagated by sweeping a vertical line $\mathcal{L}$ from left to right through C and maintaining *free space* and *reachable space* information (see Figure 4.2). The events handled by the sweep are the $O(k^4)$ vertices of the free space, and each event can be processed in $O(\log k)$ time. □

Theorem 1. *The geodesic Fréchet δ_G* decision problem *for two polygonal curves A and B in a plane with polygonal obstacles can be solved for $\varepsilon \geq 0$ in $O(N^2k^4 \log k)$ expected time, where N is the complexity of A and B, and k is the complexity of the obstacles. The weak geodesic Fréchet $\tilde{\delta}_G$* decision problem *can be solved in $O(N^2k^4)$ expected time. Both approaches use $O(Nk + k^4)$ space.*

Proof. The decision problem can be answered by propagating reachability through $O(N^2)$ cells using the dynamic programming approach of Alt and Godau [3]. The total time complexity follows by applying Lemma 4 for each cell. Since a horizontal or vertical line segment in a cell can be represented by a fixed-source shortest path map with $O(k)$ complexity [14, 17], there are at most $O(k)$ *free space* and *reachable space* intervals on any cell boundary. Consequently, all cell boundaries in the two rows required by the dynamic programming algorithm of [3] can be stored in $O(Nk)$ space. An additional $O(k^4)$ storage is needed to propagate reachability through a single cell. □

Lemma 5. *The free space diagram for the geodesic Fréchet distance between polygonal curves A and B in a plane with polygonal obstacles has $\Omega(N^2k^2)$ complexity, where N is the complexity of A and B, and k is the complexity of the obstacles.*

Proof. Figure 4.3 illustrates a situation where $\Omega(k^2)$ points (a_i, c_j) in the free space cell $\overline{ab} \times \overline{cd}$ have the same shortest path distance $d(a_i, c_j)$. All other distances in the cell are larger than $d(a_i, c_j)$. Hence, the free space cell $\overline{ab} \times \overline{cd}$ has $\Omega(k^2)$ complexity.

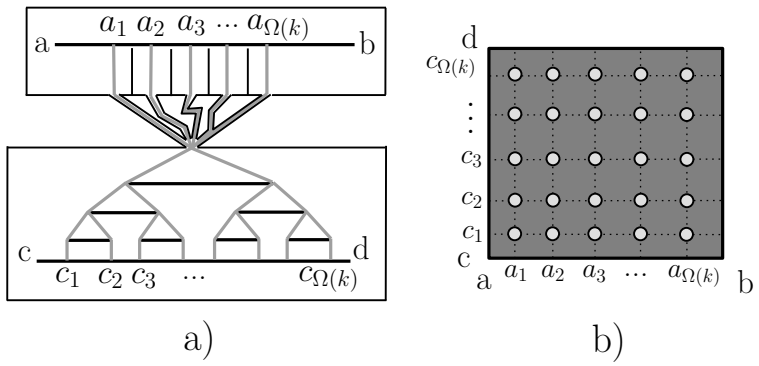

FIGURE 4.3. a) Two line segments in a plane with polygonal obstacles can define b) a single free space cell with $\Omega(k^2)$ complexity for $\varepsilon = d(a_i, c_j)$. Gray lines depict shortest paths. The obstacles are constructed such that $d(a_i, c_j)$ has the same value over all $a_1, ..., a_{\Omega(k)} \in \overline{ab}$ and $c_1, ..., c_{\Omega(k)} \in \overline{cd}$, and all other distances between $s \in \overline{ab}$ and $t \in \overline{cd}$ are greater than $d(a_i, c_j)$.

Define the polygonal curves by their endpoints as $A = \{a, b, a, b, ..., a, b\}$ and $B = \{c, d, c, d, ..., c, d\}$. Since the $\frac{N^2}{4}$ cells defined by $\overline{ab}$ and $\overline{cd}$ each have $\Omega(k^2)$ complexity, the free space diagram has $\Omega(N^2k^2)$ complexity. □

4.2. **Optimization Problem.** Critical values [3] are 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 [3] to find ε^* is to apply parametric search with Cole's [11] sorting trick.

Theorem 2. *The (strong) geodesic Fréchet distance δ_G for two polygonal curves A and B in a plane with polygonal obstacles can be computed in $O(N^2k^4 \log k \log Nk)$ expected time, where N is the complexity of A and B, and k is the complexity of the obstacles. The* weak *geodesic Fréchet distance $\tilde{\delta}_G$ can be computed in $O(N^2k^4 \log Nk)$ expected time. Both approaches use $O(Nk + k^4)$ space.*

Proof. By Lemma 3, the free space in a cell has $O(k^4)$ complexity. In fact, in the proofs of Lemmas 2 and 3, each of the $O(k^4)$ intersection vertices was described combinatorially as the intersection of two parametrized semi-algebraic sets. Each of these $O(k^4)$ intersection points defines an algebraic curve $\rho_i : [0, \infty) \to \mathbb{R}^2$ of constant degree and description complexity such that $\rho_i(\varepsilon)$ defines the location of the ith intersection point for every $\varepsilon \geq 0$.

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.[3]

Using parametric search with Cole's [11] sorting trick[4] on the $O(k^4)$ $\rho_i(\varepsilon)$ functions for each cell plus the $O(N^2k^4 \log k)$ expected runtime for the δ_G decision problem (see Theorem 1), δ_G can be computed in $O(N^2k^4 \log k \log Nk)$ expected time. Similarly, $\tilde{\delta}_G$ can be found in $O(N^2k^4 \log Nk)$ expected time using the $O(N^2k^4)$ expected runtime for the $\tilde{\delta}_G$ decision problem. The space requirement is identical to the decision problem. □

[3]Note that type (c) critical values can be ignored for $\tilde{\delta}_G$.

[4]An easier to implement alternative to parametric search is to run the decision problem once for every bit of accuracy that is desired [20].

5. TWO-SEGMENT SHORTEST PATH MAP

In this section, we construct a two-dimensional shortest path map SPM($\overline{ab}, \overline{cd}$) that represents all shortest paths in a plane with polygonal obstacles between any source point s on a line segment $\overline{ab}$ and any destination point t on a line segment $\overline{cd}$. Note that our approach can easily be adapted for two infinite lines. The dynamic and static spotlights of section 3 are the fundamental data structures used to construct SPM($\overline{ab}, \overline{cd}$).

Dynamic spotlights encode $\pi(s,t) = s \circ t$ paths, and all dynamic spotlights contribute to the final SPM($\overline{ab}, \overline{cd}$) structure. These spotlights can be constructed in $O(k^2 \log k)$ time and $O(k^2)$ space by Lemma 1.

Unlike dynamic spotlights which are always disjoint and consist entirely of shortest paths, static spotlights can overlap and represent suboptimal paths. The main task to construct SPM($\overline{ab}, \overline{cd}$) is to "clip" the static spotlights into a disjoint set of optimal paths. Regions of overlap between all static spotlights defined by o_j and o_l can be resolved by computing a hyperbolic *bisector* $B_{jl}(s) = \{(s,t) \mid d(s,o_j) + ||o_j - t|| = d(s,o_l) + ||o_l - t||\}$. For any *fixed* value of s, the bisector $B_{jl}(s)$ is a hyperbolic arc that is commonly used in additively weighted Voronoi diagrams [6] (see Figure 5.1a). Hershberger and Suri [14] have shown that for a fixed s, $B_{jl}(s)$ intersects $\overline{cd}$ in at most two points.

Theorem 3. *SPM($\overline{ab}, \overline{cd}$) can be constructed in $O(k^5)$ expected time and $O(k^5)$ space. After this preprocessing, for any $s \in \overline{ab}$ and $t \in \overline{cd}$, geodesic distance queries $d(s,t)$ take $O(\log k)$ time. Geodesic path queries $\pi(s,t)$ take $O(\log k + K)$ time, where K is the complexity of the returned path. There are instances in which SPM($\overline{ab}, \overline{cd}$) has $\Omega(k^2)$ complexity.*

Proof. The goal is to partition $\overline{ab} \times \overline{cd}$ into regions such that for all points (s,t) in a region the shortest path $\pi(s,t)$ has the same combinatorial structure. Note that the arrangement of the dynamic and static spotlights with the static spotlight bisectors defines this partition. Lemma 3 showed that the arrangement of the dynamic and static spotlights without the bisectors has $O(k^4)$ complexity. We now consider the arrangement with the static spotlight bisectors.

For a fixed obstacle vertex o_j, consider the partition of $\overline{ab}$ that is induced by all the intervals $\mathcal{I}_{ab}$ for which there exists a non-empty static spotlight $\mathcal{L}_{\mathcal{S}}(\mathcal{I}_{ab}, o_j, \mathcal{I}_{cd})$. This partition has $O(k)$ complexity and induces $O(k)$ vertical slabs in the free space cell. After overlaying the $O(k)$ vertical slabs for o_j and o_l, $\pi(s,o_j)$ and $\pi(s,o_l)$ have the same combinatorial structure throughout a vertical slab in the overlay. Hence, inside a vertical slab in this overlay the bisector $B_{jl}(s) \cap \overline{cd}$ is a (non-parametrized) semi-algebraic set of constant complexity (see Figure 5.1). Thus, over all $O(k)$ vertical slabs for each pair o_j, o_l, the bisectors define $O(k^3)$ constant-complexity semi-algebraic sets. Each of these $O(k^3)$ semi-algebraic sets intersects:

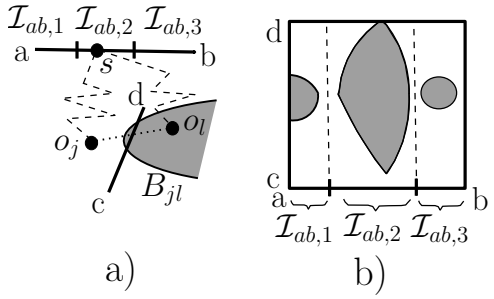

FIGURE 5.1. (a) For a fixed value of s, the bisector $B_{jl}(s)$ is a hyperbolic arc that intersects $\overline{cd}$ at most twice. (b) Inside a vertical slab in the overlay for o_j, o_l, $B_{jl}(s) \cap \overline{cd}$ is a semi-algebraic set with constant complexity. Darkly shaded points satisfy $d(s,o_l) + ||o_l - t|| \leq d(s,o_j) + ||o_j - t||$. Unshaded points satisfy $d(s,o_l) + ||o_l - t|| > d(s,o_j) + ||o_j - t||$.

(1) each of the $O(k^2)$ dynamic spotlights $O(1)$ times (because the intersection of two constant complexity sets has constant complexity) and (2) each of the $O(k^2)$ line of sight enforcing horizontal slabs for the static

spotlights $O(1)$ times. Hence, the arrangement has $O(k^5)$ complexity. A randomized incremental algorithm (see [1]) can build both the arrangement and a vertical decomposition (i.e., point location) structure for $\overline{ab} \times \overline{cd}$ in $O(k^5)$ expected time and $O(k^5)$ space. After computing the arrangement, we also compute $O(k)$ SPM(o_j) structures in $O(k^2 \log k)$ time and $O(k^2)$ space [17].

Queries proceed as follows. The vertical decomposition structure supports $O(\log k)$ deterministic time point location queries to identify the region r_j in the partition of $\overline{ab} \times \overline{cd}$ that contains the query point t. r_j is associated with a predecessor vertex o_j such that either o_j is NIL and $d(s,t) = ||s-t||$, or the precomputed SPM(o_j) can be used to report $d(s,t) = d(s,o_j) + ||o_j - t||$ and $\pi(s,t) = \pi(s,o_j) \circ t$. The lower bound of $\Omega(k^2)$ for the complexity of SPM($\overline{ab}, \overline{cd}$) follows easily from Lemma 5. □

6. Conclusion

We presented the first algorithm to compute the geodesic Fréchet distance in a plane with polygonal obstacles. Our approach uses shortest path structures that we call dynamic and static spotlights.

The shortest path map SPM($\overline{ab}, \overline{cd}$) in a plane with polygonal obstacles was also constructed using dynamic and static spotlights. It supports output-sensitive shortest path queries from any source point $s \in \overline{ab}$ to any destination point $t \in \overline{cd}$. It is interesting to note that the shortest path map structure is closely related to the Fréchet distance. In particular, it is possible to use SPM($\overline{ab}, \overline{cd}$) to construct the free space in a cell by computing the free space of SPM($\overline{ab}, \overline{cd}$). However, since the shortest path map contains additional information that is not required by the Fréchet distance, it is asymptotically faster (by an $O(k)$ factor) to compute the Fréchet distance directly.

An open problem for the non-geodesic Fréchet distance is to close the gap between the current $O(N^2 \log N)$ solution [3] and the $\Omega(N \log N)$ lower bound of [7]. Future work for our geodesic Fréchet distance is to determine a tight bound for the complexity of the free space diagram. Our algorithm constructs a free space diagram with $O(N^2k^4)$ complexity, while our lower bound on the complexity is $\Omega(N^2k^2)$. Likewise, a tighter lower bound than $\Omega(k^2)$ is also unknown for our shortest path map SPM($\overline{ab}, \overline{cd}$).

We are currently working on generalizing our SPM($\overline{ab}, \overline{cd}$) structure to SPM($\mathbb{R}^2, \mathbb{R}^2$). Using a different technique, Chiang and Mitchell [10] have shown that the *Euclidean* SPM($\mathbb{R}^2, \mathbb{R}^2$) can be represented by $O(k^{10})$ combinatorially unique shortest path maps, where $\Omega(k^4)$ are sometimes necessary.

7. Acknowledgment

We wish to thank Helmut Alt for many helpful and jovial conversations.

References

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

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

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

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

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

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

[7] K. Buchin, M. Buchin, C. Knauer, G. Rote, and C. Wenk. How difficult is it to walk the dog? *23rd European Workshop on Computational Geometry*, pages 170–173, 2007. Graz, Austria.

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

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

[10] Y. Chiang and J. S. B. Mitchell. Two-point Euclidean shortest path queries in the plane. *SODA: 10th ACM-SIAM Symposium on Discrete Algorithms*, pages 215–224, 1999.
[11] R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *J. ACM*, 34(1):200–208, 1987.
[12] A. F. Cook IV and C. Wenk. Geodesic Fréchet distance inside a simple polygon. *Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science (STACS), Bordeaux, France*, 2008.
[13] 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.
[14] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.
[15] A. Maheshwari, J.-R. Sack, and H. N. Djidjev. Link distance problems. *Handbook of Computational Geometry*, 1999.
[16] A. Maheshwari and J. Yi. On computing Fréchet distance of two paths on a convex polyhedron. *European Workshop on Computational Geometry (EWCG)*, pages 41–4, 2005.
[17] J. S. B. Mitchell. Geometric shortest paths and network optimization. *Handbook of Computational Geometry*, 1998.
[18] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.
[19] G. Rote. Computing the Fréchet distance between piecewise smooth curves. Technical Report ECG-TR-241108-01, May 2005.
[20] R. van Oostrum and R. C. Veltkamp. Parametric search made practical. *SoCG: 18th Symposium on Computational Geometry*, pages 1–9, 2002.
[21] C. Wenk, R. Salas, and D. Pfoser. Addressing the need for map-matching speed: Localizing global curve-matching algorithms. *Proc. 18th International Conference on Scientific and Statistical Database Management (SSDBM)*, pages 379–388, 2006.

APPENDIX: GEODESIC HAUSDORFF DISTANCE

Geodesic Hausdorff Distance. *Hausdorff distance* is a similarity metric commonly used to compare sets of points or sets of higher dimensional objects such as line segments or triangles. The *directed* Hausdorff distance is defined as $\tilde{\delta}_H(A,B) = \sup_{a\in A}\inf_{b\in B} d(a,b)$, where A and B are compact sets (see [3, 4]). The (*undirected*) geodesic Hausdorff distance is the larger of the two directed distances: $\delta_H(A,B) = \max(\tilde{\delta}_H(A,B),\ \tilde{\delta}_H(B,A))$.

Theorem 4. *The Euclidean geodesic Hausdorff distance in a plane with polygonal obstacles can be computed in $O((k+N)\log(k+N))$ time and space between sets A and B, where A and B are either sets of points or sets of line segments. k is the complexity of the polygonal obstacles, and N is the size of A and B.*

Proof. The algorithm of [14] can be used to compute a geodesic Voronoi diagram in $O((k+N)\log(k+N))$ time and space for a set of points or line segments. For point sets, the Voronoi diagram can be used to compute the nearest neighbor distance $\min_{b\in B} d(a,b)$ for each $a \in A$ in $O(\log(k+N))$ time. For line segment sets, the Voronoi diagram can be used to identify and resolve $O(N)$ candidate values for the Hausdorff distance using a planesweep algorithm described in [2]. Hence, the Hausdorff distance can be computed in $O((k+N)\log(k+N))$ time and space. □