Walking Your Dog in the Woods in Polynomial Time

Erin Wolf Chambers
Department of Computer Science
University of Illinois, Urbana-Champaign
erinwolf@uiuc.edu

Éric Colin de Verdière
CNRS, Département d'Informatique
Ecole Normale Supérieure, Paris, France
Eric.Colin.de.Verdiere@ens.fr

Jeff Erickson
Department of Computer Science
University of Illinois, Urbana-Champaign
jeffe@cs.uiuc.edu

Sylvain Lazard
INRIA Lorraine, LORIA
Nancy, France
lazard@loria.fr

Francis Lazarus
CNRS, Laboratoire LIS
Institut National Polytechnique de Grenoble
Francis.Lazarus@lis.inpg.fr

Shripad Thite
California Institute of Technology
Center for the Mathematics of Information
shripad@caltech.edu

ABSTRACT

The Fréchet distance between two curves in the plane is the minimum length of a leash that allows a dog and its owner to walk along their respective curves, from one end to the other, without backtracking. We propose a natural extension of Fréchet distance to more general metric spaces, which requires the leash itself to move continuously over time. For example, for curves in the punctured plane, the leash cannot pass through or jump over the obstacles ("trees"). We describe a polynomial-time algorithm to compute the *homotopic* Fréchet distance between two given polygonal curves in the plane minus a given set of obstacles, which are either points or polygons.

Categories and Subject Descriptors: F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—*Geometrical problems and computations*

General Terms: Algorithms, Performance

Keywords: Homotopic Fréchet distance, geodesic leash map

1. INTRODUCTION

Given two input curves, a natural question that arises is how similar the curves are to each other. One common measure is the Hausdorff distance, which simply takes the minimum distance between any two points, one from each curve. While the Hausdorff metric does measure closeness in space, it does not take into account the flow of the curves, which in many applications, such as morphing in computer graphics, is an important property of the curves.

The *Fréchet distance*, sometimes called the *dog-leash distance*, is defined as the minimum length of a leash required to connect a dog and its owner as they walk without backtracking along their respective curves from one endpoint to the other. The Fréchet metric takes into account the flow of the two curves because the pairs of points whose distance contributes to the Fréchet metric sweep continuously along their respective curves. It is therefore possible for two curves to have small Hausdorff distance but large Fréchet distance. Fréchet distance is used as a more accurate measure of similarity in many different applications [2, 1].

When the two curves are embedded in a general metric space, the distance between two points on the curves (the length of the shortest leash joining them) is not the Euclidean distance but a geodesic distance. For instance, this is the case if the two curves lie on a terrain (or any surface) [16] or if the leash is constrained to the interior of a simple polygon [9, 10]. The definition of the ordinary Fréchet distance allows the leash to switch discontinuously, without penalty, from one side of an obstacle or a mountain to another.

In this paper, we introduce a continuity requirement on the motion of the leash. We require that the leash cannot switch, discontinuously, from one geodesic to another; in particular, the leash cannot jump over obstacles and can sweep over a mountain only if it is long enough. We define the *homotopic Fréchet distance* between two curves as the Fréchet distance with this additional continuity requirement. This continuity requirement is satisfied automatically for curves inside a simple polygon, but not in more general environments like convex polyhedra.

The motion of the leash defines a correspondence between the two curves that can be used to morph between the two curves—two points joined by a leash morph into each other. The homotopic Fréchet distance can thus be thought as the minimal amount of deformation needed to transform one curve into the other.

In robotics, the two curves being compared may be two trajectories in the work space which happens to be the plane

SCG'08, June 9–11, 2008, College Park, Maryland, USA.
Copyright 2008 ACM 978-1-60558-071-5/08/06 ...$5.00.

with obstacles. The similarity between the two trajectories is more accurately measured by the homotopic Fréchet distance rather than the ordinary Fréchet distance which ignores obstacles.

Efficiently computing the homotopic Fréchet distance in general metric spaces is a new open problem. We present a polynomial-time algorithm for a special case of this problem, which is to compute the homotopic Fréchet distance between two polygonal curves in the plane minus a set of point or polygonal obstacles.

2. DEFINITIONS

Let S be a fixed Hausdorff metric space. A ***curve*** in S is a continuous function from the unit interval $[0,1]$ to S. We will sometimes abuse notation by using the same symbol to denote a curve $A\colon [0,1] \to S$ and its image in S. A ***reparameterization of*** $\boldsymbol{[0,1]}$ is a continuous, non-decreasing, surjection $\alpha\colon [0,1] \to [0,1]$. A reparameterization of a curve $A\colon [0,1] \to S$ is any curve $A \circ \alpha$, where α is a reparameterization of $[0,1]$. The ***length*** of any curve A, denoted $\operatorname{len}(A)$, is defined by the metric of S; in particular, two reparameterizations of the same curve have the same length.

A ***leash*** between two curves A and B is another curve $\lambda\colon [0,1] \to S$ such that $\lambda(0) = A(s)$ and $\lambda(1) = B(t)$ for some parameters s and t. A ***homotopy*** between curves A and B is a continuous map $h\colon [0,1] \times [0,1] \to S$ such that $h(\cdot,0) = A$ and $h(\cdot,1) = B$. For any $t \in [0,1]$, the one-parameter function $h(t,\cdot)$ is a leash from A to B. A ***leash map*** between curves A and B is a homotopy between some reparameterization of A and some reparameterization of B. Intuitively, a leash map describes the *continuous* motion of a leash between a dog walking along A and its owner walking along B. The ***length*** of a leash map ℓ, denoted $\operatorname{len}(\ell)$, is the maximum length of any leash $\ell(t,\cdot)$. Finally, the ***homotopic Fréchet distance*** between two curves A and B, denoted $\overline{\mathcal{F}}(A,B)$, is the infimum, over all leash maps ℓ between A and B, of the length of ℓ:

$$\overline{\mathcal{F}}(A,B) := \inf_{\text{leash map } \ell:[0,1]^2 \to S} \left(\max_{0 \leqslant t \leqslant 1} \ \operatorname{len}(\ell(t,\cdot)) \right).$$

In contrast, the classical ("leashless") Fréchet distance is defined directly in terms of reparameterizations and distances:

$$\mathcal{F}(A,B) := \inf_{\alpha,\beta:[0,1]\to[0,1]} \left(\max_{0 \leqslant t \leqslant 1} \ d(A(\alpha(t)), B(\beta(t))) \right).$$

In spaces where shortest paths vary continuously as their endpoints move, such as Euclidean space or the interior of a simple polygon, the two definitions are equivalent. In general, however, the homotopic Fréchet distance between two curves could be larger (but never smaller) than the classical Fréchet distance.

A ***homotopy relative to A and B***, or simply ***relative homotopy***, is a continuous function $h\colon [0,1] \times [0,1] \to S$, such that $h(\cdot,0)$ and $h(\cdot,1)$ are respectively of the form $A(u(\cdot))$ and $B(v(\cdot))$, where u and v are re-parameterizations of $[0,1]$.

Two leashes λ and λ' are ***relatively homotopic***, denoted $\lambda \simeq \lambda'$, if there is a relative homotopy h such that $h(0,\cdot) = \lambda$ and $h(1,\cdot) = \lambda'$. It is easy to prove that $\simeq$ is an equivalence relation over the set of leashes. Any leash map is (the transpose of) a relative homotopy; thus, all leashes $\ell(t,\cdot)$ determined by a leash map ℓ lie in the same relative homotopy class.

3. PRELIMINARIES

In this paper, we develop a polynomial-time algorithm to compute the homotopic Fréchet distance between two polygonal paths in $\mathcal{E} = \mathbb{E}^2 \setminus P$, for some set P of closed polygons, where the underlying metric is geodesic distance. The polygons P act as obstacles; in any leash map in $\mathcal{E}$, the moving leash can neither touch nor jump over any obstacle.

Specifically, the input to our problem consists of two polygonal curves A and B and a set P of polygons in the Euclidean plane. Curves A and B may (self-)intersect, but neither curve intersects any obstacle in P. To simplify our exposition, we assume that no three vertices of the input (vertices of polygons in P or vertices of A and B) are collinear; this assumption can be enforced algorithmically using standard perturbation techniques [19].

Let $a_0, a_1, \ldots, a_m$ denote the sequence of vertices of A; these points define a standard parameterization $A\colon [0,m] \to \mathcal{E}$ whose restriction to any integer range $[i-1,i]$ is an affine map onto the corresponding edge $a_{i-1}a_i$. Similarly, the vertices $b_0, b_1, \ldots, b_n$ of B define a piecewise-affine parameterization $B\colon [0,n] \to \mathcal{E}$. Let $P_1, P_2, \ldots, P_o$ denote the obstacle polygons in P, and let k denote the total number of vertices in all obstacle polygons. In the special case where every obstacle is a single point, we obviously have $k = o = |P|$. Finally, let $N = n + m + k + 2$ denote the total complexity of the input.

Figure 1 illustrates optimum leash maps for a few sample inputs where P is a discrete set of points.

3.1 Geodesic Leash Maps

To simplify our presentation, we will allow 'paths' in $\mathcal{E}$ to touch obstacles in P. Specifically, we consider ***geodesics***: piecewise-linear curves in the *closure* of $\mathcal{E}$, whose interior vertices are vertices in P. Although geodesics may run along obstacle boundaries, they do not intersect the interior of any obstacle.

In the special case where the obstacles are points (so the closure of $\mathcal{E}$ is the entire plane), we need some additional information to ensure that each geodesic lies in a unique homotopy class. Specifically, we associate a ***turning angle*** with each obstacle point that a geodesic touches. Let C_ε be a circle centered at obstacle point p and radius ε, small enough to exclude every other obstacles in P. A turning angle of θ at an obstacle point p indicates that replacing the portion of γ inside C_ε with a counterclockwise arc of length $\theta\varepsilon$ around C_ε yields a new path homotopic to γ. For example, a path with turning angle zero makes a U-turn at p without enclosing p; a path that goes straight through p with p on its left (resp. right) has turning angle π (resp. $-\pi$); a turning angle of 10π means the path makes a U-turn after winding around the point five times counterclockwise. A geodesic could meet the same obstacle point more than once; we associate a different turning angle with each incidence.

It can be shown that for any two points x and y in $\mathcal{E}$, the shortest path from x to y in any relative homotopy class is a unique geodesic, in which every turning angle at a point obstacle is either at least π or at most $-\pi$. Conversely, every geodesic in which every turning angle is either at least π or at most $-\pi$ is a shortest path in some homotopy class.

A ***geodesic leash map*** is a leash map $\ell\colon [0,1] \times [0,1] \to \mathbb{E}^2$ in which every leash $\ell(t,\cdot)$ is a geodesic, and all these

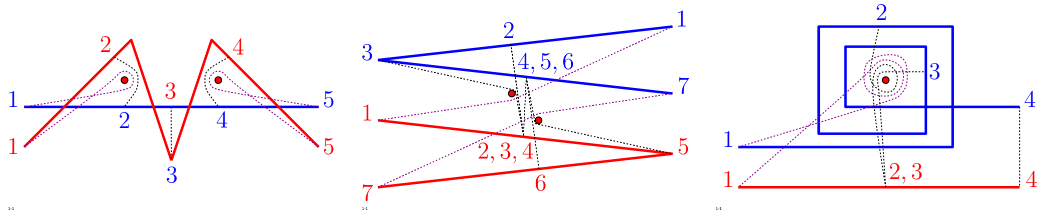

Figure 1. Optimum leash maps for three inputs. Dashed curves between matching numbers represent intermediate leashes.

geodesics are in the same relative homotopy class. We next prove that for any leash map ℓ, there is a shorter geodesic leash map ℓ' in the same homotopy class, with the same parameterizations of A and B.

Lemma 3.1. *Suppose ℓ is a leash map between two curves A and B. There is a geodesic leash map ℓ' between A and B such that, for all $t \in [0,1]$, the leash $\ell'(t,\cdot)$ is the shortest path homotopic to $\ell(t,\cdot)$ with the same endpoints. Additionally, the length of ℓ' is at most the length of ℓ.*

Proof: We lift ℓ to the universal cover $\hat{\mathcal{E}}$ of $\mathcal{E}$, obtaining a leash map $\hat{\ell}$ between the lifts $\hat{A}$ and $\hat{B}$ of A and B respectively. For each $t \in [0,1]$, let $\hat{\ell}'(\cdot,t)$ be the shortest path between the endpoints of $\hat{\ell}(t,\cdot)$ in the same homotopy class. The universal cover $\hat{\mathcal{E}}$ is a simply-connected space with a locally Euclidean metric, so shortest paths in $\hat{\mathcal{E}}$ vary continuously as the endpoints move continuously. It follows that $\hat{\ell}'$ is a continuous function in both arguments, and therefore a (geodesic) leash map in $\hat{\mathcal{E}}$. The projection ℓ' of $\hat{\ell}'$ back to S is a (geodesic) leash map between A and B. For all t, the leash $\ell'(t,\cdot)$ is the shortest path homotopic to $\ell(t,\cdot)$, so $\max_t \operatorname{len}(\ell'(t,\cdot)) \leqslant \max_t \operatorname{len}(\ell(t,\cdot))$. □

This lemma implies that the homotopic Fréchet distance between A and B is the length of a *geodesic* leash map in some homotopy class determined by some reparameterizations of A and B. Thus, the homotopic Fréchet distance can be redefined as the minimum, over all homotopy classes h, of the *classical* Fréchet distance, where distances are defined by shortest paths in relative homotopy class h:

$$\mathcal{F}_h(A,B) := \min_{\alpha,\beta:[0,1]\to[0,1]} \left(\max_{t\in[0,1]} D_h(A(\alpha(t)), B(\beta(t))) \right)$$

$$\overline{\mathcal{F}}(A,B) := \min_{\text{homotopy class } h} \mathcal{F}_h(A,B)$$

(Here, $D_h(u,v)$ denotes the length of the shortest path from u to v in relative homotopy class h.)

We call a relative homotopy class h ***optimal*** if $\overline{\mathcal{F}}(A,B) = \mathcal{F}_h(A,B)$.

For the rest of the paper, we restrict ourselves to geodesic leashes and geodesic leash maps. In Section 4, we provide a characterization of an optimal homotopy class, and we use this characterization to enumerate possible optimal homotopy classes in polynomial time. In Section 5, we describe a polynomial-time algorithm to compute the Fréchet distance within a particular homotopy class. Combining these two subroutines gives us a polynomial-time algorithm to compute homotopic Fréchet distance.

3.2 Homotopic Shortest Paths

Our algorithm relies on observations by Hershberger and Snoeyink [13] about shortest homotopic paths in the punctured plane; see also [12, 5, 11, 3, 4]. Suppose we already know a shortest path (a leash) λ between points $a \in A$ and $b \in B$, such as, for instance, a straight-line segment ab. To compute the geodesic leash between some other pair of points in the same homotopy class as λ, we follow the continuous evolution of the geodesic as the points a and b move along their respective curves. The sequence of obstacle vertices on the leash behaves like a *double-ended queue* or *deque*. A new vertex is pushed onto one end of the deque whenever the first or last segment of λ collides with an obstacle vertex. Conversely, a vertex is popped off one end of the deque when the first or last two segments of λ become collinear, and if their common vertex is a point obstacle, the turning angle at that point is either π or $-\pi$.

4. OPTIMAL HOMOTOPY CLASSES

4.1 Minimality

Let $\operatorname{len}(\lambda)$ denote the length of any geodesic leash λ, and let $\operatorname{turn}(\lambda)$ denote the sum of the absolute values of the turning angles at any point obstacles on λ. (Again, if λ meets the same point obstacle more than once, each incidence separately contributes to $\operatorname{turn}(\lambda)$. If none of the obstacles are points, then $\operatorname{turn}(\lambda) = 0$.) For any pair of leashes λ and λ', we write $\lambda \preceq \lambda'$ if and only if either (a) $\operatorname{len}(\lambda) < \operatorname{len}(\lambda')$, or (b) $\operatorname{len}(\lambda) = \operatorname{len}(\lambda')$ and $\operatorname{turn}(\lambda) \leq \operatorname{turn}(\lambda')$. We write $\lambda \prec \lambda'$ whenever $\lambda \preceq \lambda'$ but $\lambda' \not\preceq \lambda$.

We can extend this partial order to homotopy classes as follows. For any relative homotopy class h and any $s,t \in [0,1]$, let $\sigma_h(s,t)$ denote the shortest path in h between points $A(s)$ and $B(t)$. For any two homotopy classes h and h', we write $h \preceq h'$ if and only if $\sigma_h(s,t) \preceq \sigma_{h'}(s,t)$ for all parameters s and t. We write $h \prec h'$ whenever $h \preceq h'$ but $h' \not\preceq h$.

Lemma 4.1. *For any relative homotopy classes h and h', if $h \preceq h'$, then $\mathcal{F}_h(A,B) \leqslant \mathcal{F}_{h'}(A,B)$.*

Proof: Let ℓ' be an optimum leash map in homotopy class h', so that $\operatorname{len}(\ell') = \mathcal{F}_{h'}(A,B)$. For some reparameterizations α and β, we have $\ell'(t,\cdot) = \sigma_{h'}(\alpha(t),\beta(t))$ for all t. Let ℓ be the geodesic leash map in homotopy class h defined by the same reparameterizations: $\ell(t,\cdot) = \sigma_h(\alpha(t),\beta(t))$ for all t. The definition of $\preceq$ implies that $\operatorname{len}(\ell(t,\cdot)) \leqslant \operatorname{len}(\ell'(t,\cdot))$ for all t. It follows that $\mathcal{F}_h(A,B) \leqslant \operatorname{len}(\ell) \leqslant \operatorname{len}(\ell') = \mathcal{F}_{h'}(A,B)$. □

A relative homotopy class h is *minimal* if $h' \preceq h$ implies $h \preceq h'$.

Lemma 4.2. *For any relative homotopy class h, there is a minimal relative homotopy class h' such that $h' \preceq h$.*

Proof: Assume, for the sake of contradiction, that there is no minimal relative homotopy class h' such that $h' \preceq h$. Then, by induction, we can construct an infinite descending chain of relative homotopy classes $h = h_0 \succ h_1 \succ h_2 \succ \cdots$. To simplify notation, let $\sigma_n = \sigma_{h_n}(0,0)$.

Consider the ordered list of obstacle points on each path σ_n. There are finitely many such ordered lists, because $\text{len}(\sigma_n) \leqslant \text{len}(\sigma_0)$ for each n. Thus, up to taking a subsequence, we may assume that every path σ_n visits the same sequence of obstacle points. This assumption implies that all paths σ_n are geometrically equivalent and thus have equal length. Thus, by definition of $\preceq$, we have $\text{turn}(\sigma_n) < \text{turn}(\sigma_0)$ for all n. There are finitely many relative homotopy classes with a given ordered list of vertices and with bounded total absolute turning angle. (Specifically, since $\text{turn}(\sigma_n) - \text{turn}(\sigma_0)$ is always a multiple of 2π, there are at most $\lfloor \text{turn}(\sigma_0)/2\pi \rfloor + 1$ such classes.) □

The two previous lemmas immediately imply that there is a *minimal* optimal homotopy class.

In the next two subsections, we characterize minimal homotopy classes and describe how to enumerate them efficiently, first for point obstacles and then for polygonal obstacles.

4.2 Point Obstacles

Suppose P is a fixed finite set of points. A *proper line segment* is a line segment in $\mathcal{E}$ joining a point in A to a point in B.

Proposition 4.3. *A relative homotopy class is minimal if and only if it contains a proper line segment.*

Proof: One direction of the proof is straightforward. Let h be the relative homotopy class of the proper line segment σ from $A(s)$ to $B(t)$. For any relative homotopy class $h' \neq h$, the shortest path $\sigma_{h'}(s,t)$ must be longer than σ, so $\sigma_{h'}(s,t) \not\preceq \sigma = \sigma_h(s,t)$, which implies that $h' \not\preceq h$. We conclude that h is minimal.

Now let h be an arbitrary minimal homotopy class. Let $\hat{A}$ and $\hat{B}$ be lifts of A and B in the universal cover $\hat{\mathcal{E}}$, such that for all s and t, the shortest path $\hat{\sigma}_h(s,t)$ between $\hat{A}(s)$ and $\hat{B}(t)$ is a lift of $\sigma_h(s,t)$. Let $\hat{P}$ denote the set of *all* lifts of points in P; these lifted obstacle points lie on the boundary of $\hat{\mathcal{E}}$.

We prove that h contains a proper line segment in two stages. First, we prove that no lifted obstacle point $\hat{p} \in \hat{P}$ lies on *every* path $\hat{\sigma}_h(s,t)$. Next, we construct a relative homotopy from the initial leash $\sigma_h(0,0)$ to a proper line segment.

Stage 1: No common corner. For the sake of deriving a contradiction, suppose there is a lifted obstacle point $\hat{p} \in \hat{P}$ such that for all s and t, the path $\hat{\sigma}_h(s,t)$ passes through $\hat{p}$. For all s and t, the path $\hat{\sigma}_h(s,t)$ is a shortest path, so its turning angle at $\hat{p}$ must lie outside the open interval $(-\pi,\pi)$. This turning angle is a continuous function of s and t, so we can assume without loss of generality that it is always greater than π. In other words, we assume that every path $\hat{\sigma}_h(s,t)$ winds counterclockwise around $\hat{p}$.

Now $\hat{p}$ is a lift of some obstacle $p \in P$, and $\hat{\sigma}_h(s,t)$ similarly projects to a geodesic $\sigma_h(s,t)$. For each s and t, let $\tau(s,t)$ denote the path with the same vertices and turning angles as $\sigma_h(s,t)$, except that the turning angle at p is reduced by 2π. All paths $\tau(s,t)$ belong to a single relative homotopy class, which we denote h'.

Fix parameters s and t, and consider the turning angles of $\sigma_h(s,t)$ and $\tau(s,t)$ at p. If the turning angle of $\sigma_h(s,t)$ at p is strictly between π and 3π, then the turning angle of $\tau(s,t)$ at p is strictly between $-\pi$ and π. In this case, $\tau(s,t)$ cannot be the shortest path from s to t in this homotopy class, so $\text{len}(\sigma_{h'}(s,t)) < \text{len}(\tau(s,t)) = \text{len}(\sigma_h(s,t))$.

On the other hand, if the turning angle of $\sigma_h(s,t)$ at p is at least 3π, then the turning angle of $\tau(s,t)$ at p is at least π, which implies that $\tau(s,t)$ is the shortest path from s to t in h'. In this case $\sigma_h(s,t)$ and $\sigma_{h'}(s,t) = \tau(s,t)$ are geometrically equivalent and thus have equal length, but $\text{turn}(\sigma_{h'}(s,t)) = \text{turn}(\sigma_h(s,t)) - 2\pi < \text{turn}(\sigma_h(s,t))$.

Hence $\sigma_{h'}(s,t) \prec \sigma_h(s,t)$ for all s and t, which contradicts our assumption that h is a minimal homotopy class. We conclude that no lifted obstacle point $\hat{p}$ lies on every shortest path $\hat{\sigma}_h(s,t)$.

Stage 2: Homotopy construction. If the shortest path $\hat{\sigma}_h(0,0)$ is a proper line segment, then the geodesic $\sigma_h(0,0)$ is also a proper line segment, and the proof is complete. Thus, we assume that $\hat{\sigma}_h(0,0)$ passes through at least one point in $\hat{P}$.

Let $\hat{p}_1,\ldots,\hat{p}_k$ be the sequence of lifted obstacle points on the shortest path $\hat{\sigma}_h(0,0)$. (The points $\hat{p}_i$ are distinct, although their projections back into $\mathcal{E}$ might not be.) Our previous argument implies that for each i, there is a pair of parameters (s_i,t_i) such that $\hat{\sigma}_h(s_i,t_i)$ does not pass through $\hat{p}_i$.

We consider a continuous motion of the parameter point (s,t), starting at $(s,t) = (0,0)$ and then moving successively to each point (s_i,t_i). Specifically, we define two continuous functions $s\colon [0,k] \to [0,m]$ and $t\colon [0,k] \to [0,n]$ such that $s(0) = t(0) = 0$, and for any integer i, we have $s(i) = s_i$ and $t(i) = t_i$. To simplify our notation, we write $\hat{\sigma}(\tau)$ to denote the shortest path $\hat{\sigma}_h(s(\tau),t(\tau))$.

As the parameter τ ('time') increases, points in $\hat{P}$ are inserted into and deleted from the deque of vertices of $\hat{\sigma}(\tau)$. If the deque is empty at any time τ, then the shortest path $\hat{\sigma}(\tau)$ is a proper line segment, which implies that the projected path $\sigma(\tau)$ is a proper line segment as well, concluding the proof. Thus, we assume to the contrary that the deque is never empty. Each vertex $\hat{p}_1,\ldots,\hat{p}_k$ must be deleted from the deque at some time during the motion (but may be reinserted later).

Suppose $\hat{p}$ is the *last* point among $\hat{p}_1,\ldots,\hat{p}_k$ to be removed from the deque for the *first* time. Without loss of generality, we assume $\hat{p}$ is first removed from the front of the deque at time τ_1. Let $\hat{q}$ denote the second point in the deque just before $\hat{p}$ is removed; this point must exist, because the deque is never empty. The point $\hat{p}$ lies on the first segment $\hat{a}\hat{q}$ of $\hat{\sigma}(\tau_1)$, where $\hat{a} = \hat{A}(s(\tau_1))$.

By definition of $\hat{p}$, point $\hat{q}$ must have been pushed onto the *back* of in the deque at some earlier time $\tau_2 < \tau_1$. Just after $\hat{q}$ is inserted, the last two points in the deque are $\hat{p}$ and $\hat{q}$, in that order. Moreover, $\hat{q}$ lies on the last segment $\hat{p}\hat{b}$ of $\hat{\sigma}(\tau_2)$, where $\hat{b} = \hat{B}(t(\tau_2))$.

Thus, there is an *improper* line segment $\hat{a}\hat{b}$ between a point in $\hat{A}$ and a point in $\hat{B}$. Since all line segments in $\hat{\mathcal{E}}$ are shortest paths, $\hat{a}\hat{b}$ is the shortest path $\hat{\sigma}_h(\tau_1, \tau_2)$. Thus, the path $\sigma_h(\tau_1, \tau_2)$ in $\mathcal{E}$ is an *improper* line segment in relative homotopy class h. Finally, for sufficiently small $\varepsilon > 0$, one of the four paths $\sigma_h(\tau_1 \pm \varepsilon, \tau_2 \pm \varepsilon)$ is a proper line segment (because no three vertices of the input are collinear). □

Proposition 4.3 implies that we can enumerate the set of minimal relative homotopy classes in polynomial time as follows. Call a line segment ab with $a \in A$ and $b \in B$ *extremal* if it satisfies one of the following conditions:

(1) The endpoints of ab are vertices of A and B.

(2) One endpoint is a vertex of A or B and the segment contains one point in P.

(3) The segment contains two points in P.

Every proper line segment is relatively homotopic to at least one extremal line segment. Conversely, every extremal line segment carries paths in at most four minimal relative homotopy classes, distinguished by assigning turning angles of π or $-\pi$ at the obstacle vertices that lie on the segment. Thus, to enumerate the minimal relative homotopy classes, it suffices to enumerate the extremal line segments.

There are $O(mn)$ extremal segments of type (1), which we can easily enumerate in $O(mn)$ time by brute force. Each vertex $a \in A$ and point $p \in P$ determine at most n extremal segments of type (2), one for each intersection between $\overrightarrow{ap}$ and B. Similarly, each vertex $b \in B$ and point $p \in P$ determine at most n extremal segments of type (2). Thus, there are $O(mnk)$ extremal segments of type (2); again, we can easily enumerate these in $O(mnk)$ time. Finally, any two points $p, q \in P$ determine $O(mn)$ extremal segments of type (3), distinguished by the intersection points of $\overleftrightarrow{pq}$ with A and B, so there are $O(mnk^2)$ type-(3) extremal segments in total. For any obstacle points p and q, we can compute the intersection points $\overleftrightarrow{pq} \cap A$ and $\overleftrightarrow{pq} \cap B$ in $O(m+n)$ time, and then enumerate the extremal segments that contain both p and q in $O(mn)$ time, again by brute force.

Altogether, we enumerate $O(mnk^2)$ extremal line segments, and therefore $O(mnk^2)$ minimal homotopy classes, in $O(mnk^2)$ time.

There are polygonal curves and point sets that admit $\Omega(mnk^2)$ distinct minimal relative homotopy classes; see Figure 2 for an example. Thus, any improvement in this portion of the algorithm will require a finer characterization of optimal relative homotopy classes.

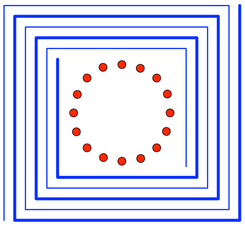

Figure 2. An input with $\Omega(N^4)$ minimal relative homotopy classes.

4.3 Polygonal Obstacles

The preceding argument breaks down if we allow polygonal obstacles; indeed, it is easy to construct instances where there are *no* proper line segments. Even if the instance allows proper line segments, the optimal homotopy class may not include one. Thus, we require a more complex characterization of minimal homotopy classes in this case.

We say that obstacle vertices p and q ***pin*** a geodesic γ if the globally shortest path from p to q (in the closure of $\mathcal{E}$) is a subpath of γ; note that p and q may be the same point. A relative homotopy class h is *pinned* if some pair of obstacle vertices pins every geodesic in h. The intersection of all geodesics within a pinned homotopy class is a shortest path between obstacle vertices, which we call the ***pinned subpath*** of h.

Lemma 4.4. *Every minimal relative homotopy class either contains a proper line segment or is pinned.*

Proof: As in the proof of Proposition 4.3, let h be an arbitrary minimal homotopy class. Let $\hat{A}$ and $\hat{B}$ be lifts of A and B in the universal cover $\hat{\mathcal{E}}$, such that for all s and t, the shortest path $\hat{\sigma}_h(s,t)$ between $\hat{A}(s)$ and $\hat{B}(t)$ is a lift of $\sigma_h(s,t)$. Let $\hat{P}$ denote the set of all lifts of the *vertices* of obstacles in P; again, every point in $\hat{P}$ lies on the boundary of $\hat{\mathcal{E}}$.

Let $\hat{\pi}_h$ denote the intersection of all shortest paths $\hat{\sigma}_h(s,t)$. If $\hat{\pi} = \varnothing$, then by Stage 2 in the proof of Proposition 4.3, h contains a proper line segment.

Otherwise, $\hat{\pi}_h$ is a shortest path between some pair of lifted obstacle vertices $\hat{p}$ and $\hat{q}$. (In the special case where $\hat{\pi}_h$ is a single point, we have $\hat{p} = \hat{q} = \hat{\pi}_h$.) Now $\hat{p}$ and $\hat{q}$ are lifts of obstacle vertices p and q (which may be the same point, even if $\hat{p}$ and $\hat{q}$ are not), and $\hat{\pi}_h$ is similarly a lift of some path π_h from p to q.

Let σ_h denote the globally shortest path from p to q, and suppose that $\pi_h \neq \sigma_h$. For each s and t, let $\tau(s,t)$ be the curve obtained from $\sigma_h(s,t)$ by replacing π_h with σ_h. All paths $\tau(s,t)$ belong to the same relative homotopy class, which we will denote h'. We now easily confirm that $h' \prec h$, contradicting our assumption that h is minimal. We conclude that π_h is the shortest path from p to q, which implies that h is a pinned relative homotopy class, and π_h is its pinned subpath. □

We call a geodesic γ ***direct*** if it consists of a proper line segment from A to some obstacle vertex p, the shortest path from p to an obstacle vertex q, and a proper line segment from q to B.

Lemma 4.5. *Every pinned relative homotopy class contains a direct geodesic.*

Proof: Let h be a pinned homotopy class, and let p and q denote the start and end of the pinned subpath π_h. Any geodesic $\sigma_h(s,t)$ consists of a geodesic $\alpha_h(s)$ from A to p, the pinned subpath π_h, and a geodesic $\beta_h(t)$ from q to B.

If $\alpha_h(0)$ is a proper segment, our claim is proved. Thus, we assume that $\alpha_h(0)$ is not a direct geodesic, which implies that the lifted path $\hat{\alpha}_h(0)$ passes through at least one obstacle vertex other than its endpoint $\hat{p}$. Let $\hat{p}^-$ be the last lifted obstacle vertex on $\hat{\alpha}_h(0)$ before $\hat{p}$. Let s_0 be the largest value such that $\hat{\alpha}_h(s)$ contains $\hat{p}^-$ for all $0 \leqslant s \leqslant s_0$. Because $\hat{p}^-$ is not on the pinned subpath, it is not on every geodesic $\hat{\alpha}_h(s)$, which implies that $s_0 < m$. For sufficiently small $\varepsilon > 0$, the geodesic $\alpha_h(s_0 + \varepsilon)$ is a proper line segment.

A similar argument implies that $\beta_h(t)$ is a proper line segment for some t. □

We can compute all homotopy classes of proper line segments using only a small modification our our earlier algorithm. In fact, we begin by running the earlier enumerate algorithm using only the vertices of polygons in P as obstacles. We then discard any line segment that intersects the interior of any obstacle polygon.

Lemma 4.5 implies that we can enumerate a superset of of the minimal relative homotopy classes in polynomial time. We begin by computing the shortest paths between every pair of obstacle vertices [14]. Next, for every pair p and q of obstacle vertices, we extend the ray $\overrightarrow{pq}$ until it reaches the interior of an obstacle (or infinity), and then compute all $O(m+n)$ intersections between the resulting line segment and the curves A and B. Finally, we concatenate all $O(mk)$ initial segments, $O(k^2)$ shortest paths, and $O(nk)$ final segments to obtain $O(mnk^4)$ pinned paths in $O(mnk^4) = O(N^6)$ time. Every pinned relative homotopy class contains one of these paths.

Unlike Proposition 4.3, we do not have an exact characterization of minimal homotopy classes for polygonal obstacles—direct geodesics do not necessarily lie in pinned homotopy classes, and not all pinned homotopy classes are minimal. However, there are problem instances with $\Omega(mnk^4)$ pinned homotopy classes; for example, replace each obstacle point in Figure 2 with a small triangle.

4.4 Non-Polygonal Obstacles

We can further generalize our characterization of minimal homotopy classes to arbitrary non-polygonal obstacles. If we replace 'obstacle vertex' with 'obstacle boundary point' in the definitions, Lemmas 4.4 and 4.5 are still true in this more general setting, although the proofs are somewhat more involved. We omit details from this extended abstract.

5. ONE HOMOTOPY CLASS

In this section, we describe an algorithm to compute the Fréchet distance $\mathcal{F}_h(A, B)$ in some relative homotopy class h. Our algorithm is a direct adaptation of Alt and Godau's algorithm for computing the classical Fréchet distance between polygonal paths in the plane [2].

As in the previous section, for any $s \in [0, m]$ and $t \in [0, n]$, let $\sigma_h(s, t)$ denote the shortest path from $A(s)$ to $B(t)$ in homotopy class h, and let $d_h(s, t) = \operatorname{len}(\sigma_h(s, t))$. For any $\varepsilon > 0$, let $F_\varepsilon \subseteq [0, m] \times [0, n]$ denote the ***free space*** $\{(s, t) \mid d_h(s, t) \leqslant \varepsilon\}$. Our goal is to compute the smallest value of ε such that F_ε contains a monotone path from $(0, 0)$ to (m, n); this is precisely the Fréchet distance $\mathcal{F}_h(A, B)$.

The parameter space $[0, m] \times [0, n]$ decomposes naturally into an $m \times n$ grid; let $C_{i,j} = [i - 1, i] \times [j - 1, j]$ denote the grid cell representing paths from the ith edge of A to the jth edge of B. Our generalization of Alt and Godau's algorithm requires that the restriction of the function d_h to any grid cell $C_{i,j}$ is convex. We prove this fact in Section 5.1 (Proposition 5.3).

As input to our problem, we are given a path $\sigma_h(s_0, t_0)$ in relative homotopy class h; based on the results of the previous section, this is either a proper line segment or a direct geodesic. Without loss of generality, we assume that the endpoints $A(s_0)$ and $B(t_0)$ are vertices of A and B; otherwise, we insert them as new vertices and reparameterize.

5.1 Geodesic Distance Is Convex

Let $A, B\colon [0, 1] \to \mathcal{E}$ be linear motions in the plane with obstacles, with (constant) derivatives $\vec{a}$ and $\vec{b}$ respectively, and let h be a relative homotopy class. For each $t \in [0, 1]$, let $\sigma(t)$ be the shortest path from $A(t)$ to $B(t)$ in relative homotopy class h, and let $d(t)$ be the length of $\sigma(t)$. The goal of this section is to prove that the function d is convex.

We omit the easy proof of the following lemma.

Lemma 5.1. *Let o be a fixed point in the plane, and let $\varphi\colon \mathbb{R}^2 \to \mathbb{R}$ be the function $p \mapsto \|\overrightarrow{op}\|$. The function φ is convex everywhere, and of class C^1 everywhere except at o. The gradient of φ at any point $p \neq o$ is $\overrightarrow{op}/\|\overrightarrow{op}\|$.*

Fix $t \in [0, 1]$. The shortest path $\sigma(t)$ is a polygonal line. Let $\vec{u}(t)$ be the unit vector representing the direction of the first line segment of $\sigma(t)$ (at its initial point $A(t)$). Similarly, we denote by $\vec{v}(t)$ the unit vector representing the direction of the last line segment of $\sigma(t)$.

Recall that, as t increases, the shortest path $\sigma(t)$ encounters a finite number of *events*, between which the set of vertices of the obstacles at which it bends is the same.

Lemma 5.2. *Between any two consecutive events, d is convex and of class C^1. In particular, $d'(t) = \vec{b} \cdot \vec{v}(t) - \vec{a} \cdot \vec{u}(t)$, where $\cdot$ denotes inner product.*

Proof: Fix two consecutive events t_0 and t_1.

Assume first that for all t between t_0 and t_1, the path $\sigma(t)$ is *not* a line segment. Then for every $t \in [t_0, t_1]$, $\sigma(t)$ is the concatenation of a line segment from $A(t)$ to a fixed obstacle vertex p, a geodesic from p to another fixed obstacle vertex q, and the line segment from q to $B(t)$. It follows that $d(t)$ equals a constant plus $\|\overrightarrow{pA(t)}\| + \|\overrightarrow{qB(t)}\|$. Our result is now a consequence of Lemma 5.1. Specifically, d is the sum of two convex functions, and is therefore convex. Since A and B do not meet the obstacle vertices M and N, the function d is C^1 in the interval $[t_0, t_1]$. The chain rule implies the claimed expression for d'.

If $\sigma(t)$ is a line segment whenever $t_0 \leqslant t \leqslant t_1$, then $d(t) = \|\overrightarrow{A(t)B(t)}\|$. Since the function $t \mapsto \overrightarrow{A(t)B(t)}$ is affine, Lemma 5.1 also implies that d is convex and of class C^1, and that

$$d'(t) = (\vec{b} - \vec{a}) \cdot \overrightarrow{A(t)B(t)}/\|\overrightarrow{A(t)B(t)}\|.$$

Finally, we observe that

$$\vec{u}(t) = \vec{v}(t) = \overrightarrow{A(t)B(t)} \,/\, \|\overrightarrow{A(t)B(t)}\|,$$

which completes the proof. □

Proposition 5.3. *The function d is convex.*

Proof: Lemma 5.2 implies that consecutive events, the function d' is continuous and non-decreasing. Let t_0 be an arbitrary event. Since the functions $t \mapsto \vec{u}(t)$ and $t \mapsto \vec{v}(t)$ are continuous at t_0, Lemma 5.2 implies that d' is also continuous at t_0. Thus, d' is non-decreasing over the entire interval $[0, 1]$, which implies that d is convex. □

Proposition 5.3 implies that the *bivariate* shortest-path distance function $D(u, v)$ between $A(u)$ and $B(v)$ is also convex, as follows. For any $u, u', v, v', t \in [0,1]$ we denote by $d_{u,v,u',v'}(t)$ the shortest-path distance between $A((1-t)u + tu')$ and $B((1-t)v + tv')$. In other words, we define

$$d_{u,v,u',v'}(t) = D((1-t)(u,v) + t(u',v')).$$

Proposition 5.3 implies that the univariate function $d_{u,v,u',v'}$ is convex. It follows that

$$\begin{aligned} D((1-t)(u,v) + t(u',v')) &= d_{u,v,u',v'}((1-t)\cdot 0 + t \cdot 1) \\ &\leqslant (1-t)d_{u,v,u',v'}(0) + td_{u,v,u',v'}(1) \\ &= (1-t)D(u,v) + tD(u',v'), \end{aligned}$$

which expresses the convexity of D.

The convexity of D immediately implies the following corollary.

Corollary 5.4. *For all integers i and j, the restriction of d_h to any grid cell $C_{i,j}$ is convex.*

5.2 Preprocessing for Distance Queries

The only significant difference between our algorithm and Alt and Godau's is that we require additional preprocessing to compute several *critical distances* and an auxiliary data structure to answer certain *distance queries.* (If there are no obstacles, each critical distance can be computed, and each distance query can be can be answered, in constant time.)

There are three types of critical distances:

- ***endpoint distances*** $d_h(0,0)$ and $d_h(m,n)$,
- ***vertex-edge distances*** $d_h(i,[j-1,j]) = \min\{d_h(i,t) \mid t \in [j-1,j]\}$ for all integers $i \in [0,m]$ and $j \in [1,n]$, and
- ***edge-vertex distances*** $d_h([i-1,i],j) = \min\{d_h(s,j) \mid s \in [i-1,i]\}$ for all integers $i \in [1,m]$ and $j \in [0,n]$.

Given integers i and j and any real value ε, a ***horizontal distance query*** asks for all values of $t \in [j-1,j]$ such that $d_h(i,t) = \varepsilon$, and a ***vertical distance query*** asks for all values of $s \in [i-1,i]$ such that $d_h(s,j) = \varepsilon$. The convexity of d_h within any grid cell implies that any distance query returns at most two values.

We first describe how to preprocess a single vertical edge in the parameter grid to answer distance queries; computation of the critical values will be done automatically during the preprocessing. Obviously a similar result applies to horizontal grid edges.

Lemma 5.5. *Suppose we are given a point p and a line segment $\ell = \overline{xy}$, parameterized over $[0,1]$, as well as the geodesic $\sigma_h(p,x)$ and its length $d_h(p,x)$. In $O(k \log k)$ time, we can build a data structure of size $O(k)$ such that for any ε, all values $t \in [0,1]$ such that $d_h(p, \ell(t)) = \varepsilon$ can be computed in $O(\log k)$ time. We also report the critical vertex-edge distance $d_h(p,\ell)$, the path $\sigma_h(p,y)$, and its length $d_h(p,y)$.*

Proof: We first compute a constrained Delaunay triangulation of the polygons P, the segment ℓ, and point p in time $O(k \log k)$. This triangulation includes ℓ and the edges of polygons in P as edges.

We apply the following observations used in the *funnel* algorithm for computing shortest homotopic paths [6, 15, 13]. The shortest homotopic paths $\sigma_h(p,x)$ and $\sigma_h(p,y)$ may share a common subpath and then split at some vertex v; this vertex is then the apex of two concave chains that form a funnel with base xy. Each concave chain has complexity at most k and intersects a given edge of the triangulation at most twice.

The geodesic from p to x may have complexity greater than $O(k)$, but (as observed above) the concave chain from v to x will have at most $O(k)$ segments. Our goal is to find a vertex w on $\sigma_h(p,x)$ such that the path from w to x contains v. In other words, the chain from w to x along $\sigma_h(p,x)$ will be of complexity $O(k)$ and will contain the concave funnel path.

To find w, walk along the path from x to p. If we find a vertex where the chain is not concave, we must have passed v, so we mark the non-concave vertex as w. If we ever re-cross a segment of the triangulation a second time, we again must have passed the funnel apex v so we can mark the second crossing as w. (We walked along $O(k)$ edges of the chain to find w.) Let π be the portion of $\sigma_h(p,x)$ between p and w, and τ_1 be the portion of $\sigma_h(p,x)$ between w and x.

We know that π is contained in $\sigma_h(p,y)$, since w is before the apex of the funnel v. Let τ_2 be the portion of $\sigma_h(p,y)$ between w and y; this can be computed in $O(k)$ time using the funnel algorithm. Given τ_2, we can then find the apex of the funnel v in $O(k)$ time.

Imagine extending each line segment on the concave chains until it intersects ℓ, the line connecting x and y. Between the two concave chains, the combinatorial description of the distance function changes only at points where the extended lines meet ℓ. To answer distance queries, we will record the $O(k)$ intersections of the extended lines with ℓ. For each of the resulting intervals, record the (fixed) length of the geodesic up to the first vertex in the extended line, as well as the equations of the two lines that bracket the interval. In constant time per interval, we can also compute and store the value $t^* \in [0,1]$ such that $d_h(p, \ell(t^*))$ is minimized, along with the path $\sigma_h(p, \ell(t^*))$; this gives the desired value $d_h(p,\ell)$.

The funnel data structure requires $O(k)$ space to store the $O(k)$ combinatorial changes to the leash as its endpoint sweeps $\ell = \overline{xy}$.

Now given this data structure, we answer distance queries as follows. If the distance queried is smaller than $d_h(p,\ell)$, we return the empty set. If it is equal to $d_h(p,\ell)$, we return $\ell(t^*)$. If it is larger than $d_h(p,\ell)$, we do two binary searches, one on the intervals between x and $\ell(t^*)$ and the other on the intervals between $\ell(t^*)$ and y. □

Lemma 5.6. *In $O(mnk \log k)$ time and using $O(mnk)$ space, we can compute all critical distances, as well as a data structure of size $O(mnk)$ that can answer any horizontal or vertical distance query in $O(\log k)$ time.*

Proof: We preprocess each edge of the parameter grid as described in Lemma 5.5. We start from the vertex (i,j) that is our given input, either a straight line segment or a direct geodesic. We then walk on the edges of the grid, visiting each edge at least once and at most $O(1)$ times. During this walk, at each current vertex (i,j), we maintain the shortest homotopic path $\sigma_h(i,j)$ and its length $d_h(i,j)$. Each time we walk along an edge, we apply Lemma 5.5 to preprocess it and to compute the shortest homotopic path corresponding to the target vertex of that edge. Each step takes $O(k \log k)$ time, and there are $O(mn)$ edges, whence the running-time. As we walk along an edge of the parameter grid, we use a deque to push and pop the obstacle vertices along the leash in constant time per operation. Since at most k vertices are pushed onto the deque for each grid edge, the total size of the deque is $O(mnk)$. □

5.3 Decision Procedure

Like Alt and Godau, we first consider the following *decision problem*: Is $\mathcal{F}_h(A, B)$ at least some given value ε? Equivalently, is there a monotone path in the free space F_ε from $(0, 0)$ to (m, n)? Our algorithm to solve this decision problem is identical to Alt and Godau's, except for the $O(\log k)$-factor penalty for distance queries; we briefly sketch it here for completeness.

For any integers i and j, let $h_{i,j}$ denote the intersection of the free space F_ε with the horizontal edge $([i-1, i], j)$, and let $v_{i,j}$ denote the intersection of F_ε with the vertical edge $(i, [j-1, j])$. In the first phase of the decision procedure, we compute $h_{i,j}$ and $v_{i,j}$ for all i and j, using one distance query (and $O(\log k)$ time) for each edge of the parameter grid.

In the second phase of the decision procedure, we propagate in lexicographic order from $C_{1,1}$ to $C_{m,n}$ and determine which $h_{i,j}$ and $v_{i,j}$ are reachable via a monotone path from $C_{1,1}$. Since the free space in each $C_{i,j}$ is convex, we can propagate through each cell in constant time.

Our decision algorithm returns true if and only if there is a monotone path that reaches (m, n). The total running time is $O(mn \log k)$.

5.4 Computing Fréchet Distance

Finally, we describe how to use our decision procedure to compute the optimum value $\varepsilon^* = \min\{\varepsilon \mid (m, n) \in R_\varepsilon\}$; this is the Fréchet distance $\mathcal{F}_h(A, B)$.

We start by computing critical distances and the distance-query data structure in $O(mnk \log k)$ time, as described in Lemma 5.6. We then sort the $O(mn)$ critical distances. Using the decision procedure, we can compare the optimal distance ε^* with any critical distance ε in $O(mn \log k)$ time. By binary search, we can, repeating this step $O(\log mn)$ times, compute an interval $[\varepsilon^-, \varepsilon^+]$ that contains ε^* but no critical distances.

We then apply Megiddo's *parametric search* technique [17]; see also [7, 18]. Parametric search combines our decision procedure with a 'generic' parallel algorithm whose combinatorial behavior changes at the optimum value ε^*. Alt and Godau observe that one of two events occurs when $\varepsilon = \varepsilon^*$:

- For some integers i, i', j, the bottom endpoint of $v_{i,j}$ and the top endpoint of $v_{i',j}$ lie on the same horizontal line.
- For some integers i, j, j', the left endpoint of $h_{i,j}$ and the right endpoint of $h_{i,j'}$ lie on the same vertical line.

Thus, it suffices to use a 'generic' algorithm that sorts the $O(mn)$ endpoint *values* of all non-empty segments $h_{i,j}$ and $v_{i,j}$, where the value of an endpoint (s, j) of $h_{i,j}$ is s, and the value of an endpoint (i, t) of $v_{i,j}$ is t.

We use Cole's parallel sorting algorithm [8], which runs in $O(\log N)$ parallel steps on $O(N)$ processors, as our generic algorithm. Each parallel step of Cole's sorting algorithm needs to compare $O(mn)$ endpoints. The graph of an endpoint, considered as a function of ε, is convex, monotone, and made of $O(k)$ pieces, each having a simple closed form (see proof of Lemma 5.5). It follows that the sign of a comparison between two endpoints may change at $O(k)$ different values of ε that can be computed in $O(k)$ time. Applying the parametric search paradigm requires the following operations for each parallel step of the sorting algorithm:

- Compute the $O(mnk)$ values of ε corresponding to the changes of sign of the $O(mn)$ comparisons. This can be done in $O(mnk)$ time and $O(mnk)$ space.
- Apply binary search to these values by median finding, calling the decision procedure to discard half of them at each step of the search. This takes $O(mnk + T_d \log(mnk))$ time, where $T_d = O(mn \log k)$ is the running time of our decision procedure. We obtain this way an interval for ε where each of the $O(mn)$ comparisons has a determined sign.
- Deduce in $O(mn \log k)$ time the sign of each of the $O(mn)$ comparisons within the previously computed interval.

Since the underlying sorting algorithm requires $O(\log mn)$ parallel steps, the resulting parametric search algorithm runs in time $O(mn \log(mn)(k + \log k \log(mnk))) = O(N^3 \log N)$. The distance query data structure requires $O(mnk)$ space. We require an additional $O(mnk)$ space to simulate sequentially each parallel step of the sorting algorithm; we can re-use this space for subsequent parallel steps. Therefore, the total space complexity of our algorithm is $O(mnk) = O(N^3)$.

Lemma 5.7. *Given either a proper line segment or a direct geodesic in relative homotopy class h, the Fréchet distance $\mathcal{F}_h(A, B)$ can be computed in $O(N^3 \log N)$ time and $O(N^3)$ space.*

5.5 Summary

Finally, to compute the homotopic Fréchet distance $\overline{\mathcal{F}}(A, B)$ in the plane minus a set of point obstacles, we compute the Fréchet distance $\mathcal{F}_h(A, B)$ in each of the $O(mnk^2)$ minimal homotopy classes h. Similarly, for polygonal obstacles, we compute $\mathcal{F}_h(A, B)$ for every class h in a set of $O(mnk^4)$ relative homotopy classes, which includes every minimal homotopy class.

Theorem 5.8. *The homotopic Fréchet distance between two polygonal curves in the plane minus a set of points can be computed in $O(N^7 \log N)$ time.*

Theorem 5.9. *The homotopic Fréchet distance between two polygonal curves in the plane minus a set of polygons can be computed in $O(N^9 \log N)$ time.*

6. CONCLUSION

In this paper, we introduced a natural generalization of the Fréchet distance between curves to more general metric spaces, called the homotopic Fréchet distance. We described a polynomial-time algorithm to compute the homotopic Fréchet distance between polygonal curves in the plane with point or polygon obstacles.

Improving the running time of our algorithms is the most immediate outstanding open problem. For point obstacles, we conjecture that the running time can be improved by at least a factor of N by optimizing leash maps in every minimal homotopy class simultaneously. Since shortest paths between the same endpoints but belonging to different homotopy classes are related, we expect to (partially) reuse the results of shortest path computations going from one homotopy class to another. For polygonal obstacles, an exact characterization of minimal homotopy classes would almost certainly lead to a significantly faster algorithm.

Cook and Wenk [9] describe an algorithm for computing geodesic Fréchet distance between two curves within a simple polygon, generalizing earlier results of Efrat *et al.* [10]. Their algorithm is faster than ours by roughly a factor of N, in part because they use a randomized strategy in place of parametric search. Unfortunately, we have not been able to apply their technique to our more general problem, because it seems to require a simply-connected environment. However, similar ideas may simplify and improve our algorithm.

Weak Fréchet distance is a variant of the ordinary Fréchet distance without the requirement that the endpoints move monotonically along their respective curves—the dog and its owner are allowed to backtrack to keep the leash between them short. Alt and Godau [2] gave a simpler algorithm for computing the weak Fréchet distance, using a graph shortest-path algorithm instead of parametric search. A similar simplification of our algorithm computes the weak *homotopic* Fréchet distance between curves in the punctured plane in polynomial time. We omit further details from this extended abstract.

It would be interesting to compute homotopic Fréchet distance in more general spaces. In particular, we are interested in computing the homotopic Fréchet distance between two curves on a convex polyhedron, generalizing the algorithm of Maheshwari and Yi for classical Fréchet distance [16]. The vertices of the polyhedron are 'mountains' over which the leash can pass only if it is long enough. Shortest paths on the surface of a convex polyhedron do *not* vary continuously as the endpoints move, because of the positive curvature at the vertices, so we cannot consider only geodesic leash maps.

Finally, it would also be interesting to consider the homotopic (weak) Fréchet distance between higher-dimensional manifolds; such problems arise with respect to surfaces in configuration spaces of robot systems. Standard Fréchet distance is difficult to compute in higher dimensions, although the *weak* Fréchet distance between two triangulated surfaces can be computed in polynomial time [1].

Acknowledgments

This research was initiated during a visit to INRIA Lorraine in Nancy, made possible by a UIUC/CNRS/INRIA travel grant. Research by Erin Chambers and Jeff Erickson was also partially supported by NSF grant DMS-0528086; Erin Chambers was additionally supported by an NSF graduate research fellowship. Research by Shripad Thite was partially supported by the Netherlands Organisation for Scientific Research (NWO) under project number 639.023.301 and travel by INRIA Lorraine. We thank Hazel Everett and Sylvain Petitjean for useful discussions and great company, and Kira and Nori for several walks in the woods.

References

[1] H. Alt and M. Buchin. Semi-computability of the Fréchet distance between surfaces. *Proc. 21st European Workshop on Computational Geometry*, 45–48, 2005.

[2] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *IJCGA* 5(1–2):75–91, 1995.

[3] S. Bespamyatnikh. Computing homotopic shortest paths in the plane. *Proc. 14th Annu. ACM-SIAM Sympos. Discrete Algorithms*, 609–617, 2003.

[4] S. Bespamyatnikh. Encoding homotopy of paths in the plane. *Proc. LATIN 2004: Theoretical Infomatics*, 329–338, 2004. Lect. Notes Comput. Sci. 2976, Springer-Verlag.

[5] S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. *Discrete Comput. Geom.* 31(1):61–81, 2004.

[6] B. Chazelle. A theorem on polygon cutting with applications. *Proc. 23rd Annu. IEEE Sympos. Found. Comput. Sci.*, 339–349, 1982.

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

[8] R. Cole. Parallel merge sort. *SIAM J. Comput.* 17(4):770–785, 1988.

[9] A. Cook and C. Wenk. Geodesic Fréchet and Hausdorff distance inside a simple polygon. Tech. Rep. CS-TR-2007-004, U. Texas San Antonio, 2007.

[10] 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 Comput. Geom.* 28:535–569, 2002.

[11] A. Efrat, S. G. Kobourov, and A. Lubiw. Computing homotopic shortest paths efficiently. *Comput. Geom. Theory Appl.* 35(3):162–172, 2006.

[12] D. Grigoriev and A. Slissenko. Polytime algorithm for the shortest path in a homotopy class amidst semialgebraic obstacles in the plane. *Proc. Internat. Sympos. Symbolic and Algebraic Computation*, 17–24, 1998.

[13] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Computational Geometry: Theory and Applications* 4:63–67, 1994.

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

[15] D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks* 14:393–410, 1984.

[16] A. Maheshwari and J. Yi. On computing Fréchet distance of two paths on a convex polyhedron. *Proc. 21st European Workshop on Computational Geometry*, 41–44, 2005.

[17] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM* 30:852–866, 1983.

[18] R. van Oostrum and R. C. Veltkamp. Parametric search made practical. *Computational Geometry: Theory and Applications* 28:75–88, 2004.

[19] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.* 19:1–17, 1998.