On Computing Fréchet Distance of Two Paths on a Convex Polyhedron *

Anil Maheshwari Jiehua Yi

Abstract

We present a polynomial time algorithm for computing Fréchet distance between two simple paths on the surface of a convex polyhedron.

1 Introduction

Distance measures used to match geometric patterns include: Hausdorff-distance, Fréchet-distance, uniform distance, etc. Alt and Godau [1, 2] proposed that Fréchet distance is one of the most fundamental measures to compute the similarity between two polygonal curves. Fréchet distance is often referred to as the dog-leash distance [1]. The unique property is its sensitivity to the order along the two curves. Fréchet distance is the minimum leash distance that can keep the person and the dog walking on their own tracks from the beginning to the end (without retracting). Some issues on similarity related to Fréchet distance have been considered, for example, find a curve that is similar to a given curve [3, 4], and the application of Fréchet distance on protein backbone matching.

This paper focuses on the following problem: Given a convex polyhedron P consisting of n triangular faces, and two simple paths Z and Z' on the surface of P, how can we use Fréchet distance to measure the similarity of Z and Z'? For the sake of simplicity, assume that each segment in these paths is an edge of P, and Z consists of z segments and Z' consists of z' segments. Referring to the dog-leash distance, the person is walking on the path Z and the dog is walking on the path Z'. The leash defines a geodesic path on the surface of P. Here, Fréchet distance is measured using Euclidean shortest path distance on the surface of P.

In this paper, we present a polynomial time algorithm to compute Fréchet distance between the paths Z and Z' on the surface of a convex polyhedron P. To accomplish this we make use of two data structures: (i) a data structure of the visibility diagram that encodes shortest path information for any pair of points on a pair of edges (e_s, e_t), where $e_s \in Z$ and $e_t \in Z'$ and (ii) the data structure of the free space diagram proposed in [1] for paths in plane. The novelty is in adapting the free space diagram with the aid of visibility diagrams for the problem discussed in this paper. In the next section we discuss the visibility diagram data structure and in Section 3 we present an algorithm to compute Frèchet distance for paths on a convex polyhedron.

2 Visibility Diagram

The visibility diagram of a pair of edges, say e_s and e_t, lying on the surface of P is a data structure that concisely represents geodesic distances between any pair of points p and q, where $p \in e_s$ and $q \in e_t$. (Due to the lack of space we cannot discuss the literature regarding the computation of geodesic paths on a convex polyhedron. For detailed discussion on this we refer the reader to [5, 6, 10].) We make use of the algorithm of [6], and that in turn makes calls to the algorithm of [5].

Algorithm 1 *Visibility-Diagram*
(1) Construct the edge sequence tree T of edge e_s using the algorithm of [5].
(2) Identify those edge sequences in tree T which start at e_s and end at the edge e_t. Let the set of these edge sequences be $\mathcal{E}$.
(3) Unfold each of the edge sequence in $\mathcal{E}$ and construct the visibility polygon for each unfolding.
(4) Compute the overlay of the visibility polygons to obtain the visibility diagram. Label each area in the overlay with the corresponding edge sequence.
(5) Output the final visibility diagram.

Details of this algorithm are provided in [9]. This algorithm uses the concept of *edge sequence*. According to the shortest path properties in [7, 8], a shortest path $\Pi(p \in e_s, q \in e_t)$ on P is identified uniquely by its endpoints and the sequence of edges $\{e_s, e_1, e_2, ..., e_k, e_t\}$ that it crosses. This sequence of edges is called an *edge sequence* of P $(\xi(\Pi(p, q)))$. The faces $\{f_1, f_2, ..., f_{k+1}\}$ that the shortest path traversed can be unfolded to a plane, by rotating the face f_1 into the coordinate system of f_2 around the common edge e_1 of f_1 and f_2, and then rotate f_1 and f_2 to the coordinate system of f_3, and so on. Following these steps, all of the faces can be located in the coordinate system of f_{k+1}, and this forms a planar graph. Geodesic paths in the unfolding map to straight line segments. Refer to Figure 1(b).

Define the domain $z = e_s \times e_t$ as a unit square and it is an affine mapping of the edges e_s and e_t on

*Research supported by NSERC. School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6, {anil,jyi}@scs.carleton.ca

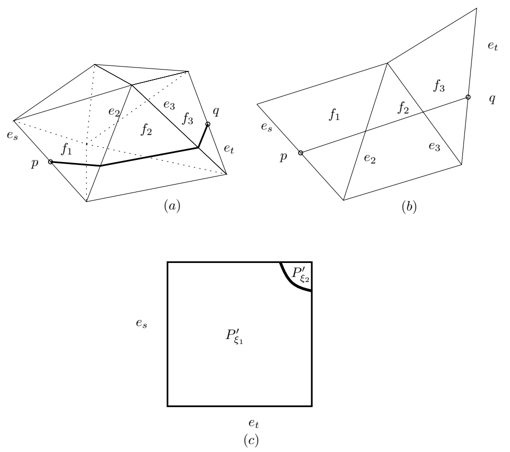

Figure 1: (a): A sequence of edges and faces that a shortest path from $p \in e_s$ to $q \in e_t$ passes through; (b): The planar unfolding relative to the edge sequence; (c): The visibility diagram of e_s and e_t. P'_{ξ_1} corresponds to the outer boundary of the unfolded edge sequence in (b), which is a polygon. P'_{ξ_2} corresponds to another unfolded edge sequence.

$[0, 1]$, see Figure 1(c). The visibility diagram is defined as the partition of the domain z, each partition corresponds to an unfolded edge sequence starting at e_s and ending at e_t. (Mount [8] has shown that the number of such edge sequences is $O(n^2)$.) In a partition the pair of points are visible to one another in their corresponding unfolded edge sequence. During the construction of the visibility diagram, if the pair of points in a partition can see each other in more than one unfolded edge sequence, then this partition is further subdivided, until each partition corresponds to one unfolded edge sequence. It can be shown that the boundary between each pair of partition in the visibility diagram is a hyperbolic curve, and each partition is a polygon. Observe that geodesic path for any pair of points in e_s and e_t can be computed from their corresponding unfolded edge sequence in the visibility diagram. Thus, the visibility diagram for a pair of edges can be computed by simultaneously overlaying $O(n^2)$ visibility polygons corresponding to each of the unfolded edge sequence; it can be computed in $O(n^3 \log n)$ time.

3 Algorithm to compute Fréchet Distance

First we briefly outline Fréchet Diagram for two polygonal curves Z and Z' in plane as described in [1]. They used affine mapping to represent a continuous and piecewise linear curve. If the curve Z is a line segment and similarly the curve Z' is a line segment then the set

$$F_\epsilon = \{(s,t) \in [0,1]^2 \mid d(Z(s), Z'(t)) \leq \epsilon\} \qquad (1)$$

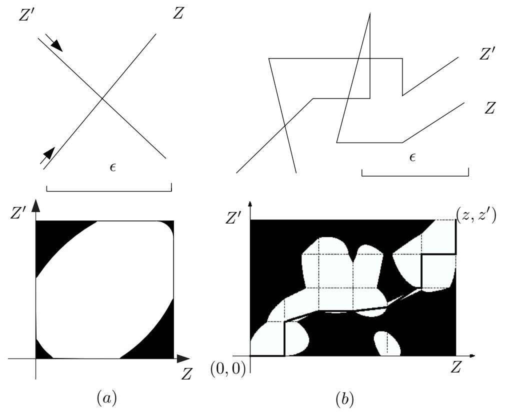

Figure 2: (a): Two segments Z and Z' and their free space diagram for a given ϵ. (b): Two polygonal curves Z and Z' and their free space diagram for a given ϵ. A monotone path from $(0, 0)$ to (z, z') in the free space diagram.

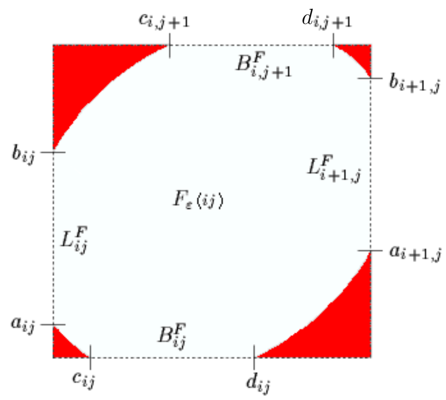

Figure 3: The boundary values of a unit cell in the free space diagram that must be calculated

describes all of the pairs of points in the affine mapping of Z and Z', whose Euclidean distance is at most ϵ. Figure 2(a) shows line segments Z and Z', and a distance $\epsilon > 0$; F_ϵ is the white area within the unit square, which is an ellipse [1], subsequently called as the *free space diagram*. Figure 2(b) shows polygonal curves Z and Z' with z and z' segments, respectively, and their free space diagram F_ϵ. This is obtained by combining the free space diagrams for each pair of segments of Z and Z'.

Lemma 1 *[1] For polygonal curves Z and Z' in plane, their Fréchet distance, $\delta_F(Z, Z') \leq \epsilon$, only if there exists a curve within the corresponding free space diagram F_ϵ from $(0, 0)$ to (z, z') that is monotone in both coordinates.*

In order to ensure that there is a passage for the path between neighboring cells in the diagram, Figure 3 illustrates certain boundary values that needs to be calculated. These values correspond to the in-

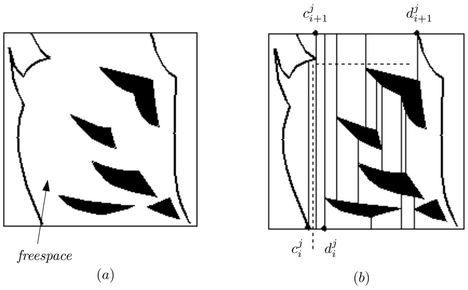

Figure 4: The trapezoidation of a free space after union. (a) : The free space (white areas). (b) : After trapezoidation and executing BFS (the dashed line is one of the paths), the upper boundary value is $c^j_{i+1}d^j_{i+1}$.

tersections of the ellipse with the boundary of the cell. Above lemma provides a mechanism to check whether Fréchet distance is at most ϵ. In [1] it is shown that the exact value of the distance is determined by an ϵ corresponding to one of the following cases: (i) $(0,0) \in F_\epsilon$ and $(z,z') \in F_\epsilon$, (ii) $L^F_{i,j}$ or $B^F_{i,j}$ becoming nonempty for some pair (i,j), or (iii) $a_{i,j} = b_{k,j}$ or $c_{i,j} = d_{i,k}$ for some i,j,k. Therefore, to determine the exact distance, one needs to apply Lemma 1 for only the set of critical values of ϵ as determined by the above cases. It turns out that the total number of critical values is $O(z^2z' + zz'^2)$, and hence Fréchet distance between two paths Z and Z' in the plane can be computed in $O(zz'\log(zz'))$ time.

In the rest of this paper we sketch how we can adapt the free space diagram with the aid of the visibility diagrams for the case of convex polyhedron. Let $Z : [0..z]$ and $Z' : [0..z']$ be the two paths on the boundary of convex polyhedron consisting of z and z' segments, respectively. Moreover, $Z(i-1)Z(i)$ and $Z'(j-1)Z'(j)$, for $1 \le i \le z, 1 \le j \le z'$, represents those segments. For a fixed ϵ, we construct a free space cell for a pair of edges $Z(i-1)Z(i)$ and $Z'(j-1)Z'(j)$. The outer boundary of the free space cell is a unit square and it is exactly the same square as the boundary of the visibility diagram of those two edges. But the main difficulty arises due to the fact that the polygons in the visibility diagram belong to different unfolded edge sequences, and each unfolded edge sequence has its own coordinate system in the plane. Because of this, the free spaces of $Z(i-1)Z(i)$ and $Z'(j-1)Z'(j)$ in the unfolded edge sequences are different from one another. Thus, we compute the intersection of the free space cell with the visibility polygon, where both the cell and the polygon correspond to the same unfolded edge sequence. For each pair of points in the intersection area, their shortest path distance is at most ϵ. The free space diagram of the unit cell of $Z(i-1)Z(i)$ and $Z'(j-1)Z'(j)$ is obtained by taking the union of the free spaces for all the

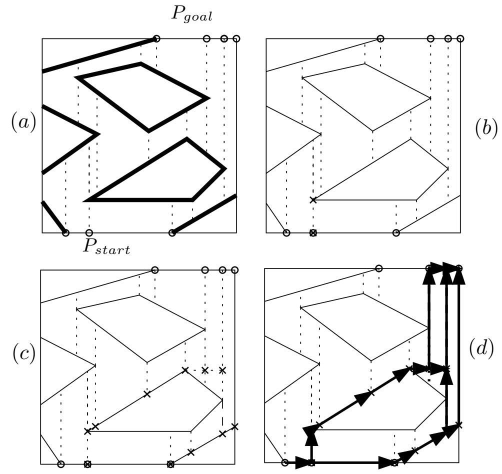

Figure 5: (a)Placing boundary nodes in the trapezoidal map. $(b)(c)$: Illustration of placing interior nodes in the trapezoidal map. (d): Adding arcs between the nodes to form a directed graph. For simplifying the figures, boundaries of the polygons are drawn as straight line segments.

corresponding edge sequences. Refer to Figure 4(a). The free space diagram for Z and Z' for a fixed ϵ is obtained by applying the above computation steps on each pair of segments.

Once we have the free space diagram for a specific value of ϵ, we need to test whether we have a monotone path from $(0,0)$ to (z,z'). Observe that if there is a monotone path then each part of the path in the corresponding cells must be monotone. The algorithm starts from the first pair of edges including $(0,0)$, ends at the last pair of edges including (z,z'), and computes the boundary values for each unit cell in the free space diagram subject to the monotone path restrictions. If (z,z') can be reached, then this particular choice of ϵ results in a valid monotone path in the corresponding free space diagram. The boundary values for each cell are obtained by performing breadth first search on a modified dual map of the trapezoidal decomposition of each cell, refer to Figure 5. The computation of the boundary values for a unit cell proceeds in three steps. First, we place candidate nodes in the trapezoidal map from where a monotone path could pass through. The nodes placed on the boundary of the unit cell are candidates for the boundary values, we use P_{start} and P_{goal} to mark them (marked as " $\circ$ " in Figure 5(a)), respectively. The nodes placed in the interior of the free space cell (marked as " $\times$ " in Figure 5$(b)(c)$) can be reached monotonically from the neighboring nodes placed earlier. Second, connect the neighboring nodes with the directed arcs, the arcs must also be in the free space area. Thus, a directed graph is formed in the free space area of the unit cell. Third, by applying the breadth first search in this directed graph, a monotone path (if it exists) from a starting point P_{start} to a point P_{goal} is identified (see Figure 4(b)). There is a technical detail involved here, and this arises due to the fact that the edges of the polygons in a unit cell can be hyperbolic curves or elliptical arcs, refer to Figure 4(a). Therefore, trapezoidation needs to take this in account, as well as care must be taken while placing the nodes to ensure that the directed arcs reside in the free space. We omit the details of this part here.

Suppose that we have already computed the visibility diagram for a pair of edges in $O(n^3 \log n)$ time. For a fixed ϵ, we can show that the union of the free spaces in the unit cell for the pair of edges can be computed in $O(n^2 \log n)$ time, since there are $O(n^2)$ polygons in the visibility diagram; Computing the boundary value of a unit cell takes $O(n^2 \log n)$ time, which includes the trapezoidal map and the directed acyclic graph construction, and performing breadth first search. Therefore, in $O(zz'n^3 \log n)$ time, we can determine whether Fréchet distance between the paths Z and Z' is at most ϵ. This is summarized in the following lemma, and it can be viewed as the analog of Lemma 1 for the case of convex polyhedron.

Lemma 2 *For simple paths Z and Z' on the surface of convex polyhedron, their Fréchet distance, $\delta_F(Z, Z') \leq \epsilon$, only if there exists a curve within the corresponding free space diagram F_ϵ from $(0,0)$ to (z, z') that is monotone in both coordinates. Moreover this can be determined in $O(zz'n^3 \log n)$ time where z and z' are the number of the segments in Z and Z', and n is the number of faces on P.*

Next we need to determine what are the critical values of ϵ, and then we can search among them to determine Fréchet distance between the paths. The three kinds of critical values of ϵ, as in the planar case, are also suitable for the convex polyhedron case. But due to the complex nature of cells in this case, we need to extend the third case; the main idea is captured in the following observation. Assume that the minimum value of ϵ is not determined by the first two cases. Then we claim that the minimum value of ϵ that ensures that there is a monotone path in the free space diagram will consists of a segment that is parallel to one of the coordinate axes.

Recall that in the two dimensional case, the third case corresponds to either $a_{i,j} = b_{k,j}$ or $c_{i,j} = d_{i,k}$ for some i, j, k. In the light of the above observation, for the convex polyhedron case this needs to be extended to the case when $a_{i,j}$ and $b_{k,j}$ are located inside the cells, if we use the same labels for the points inside the cells as the labels for the boundary values; see Figure 3. Imagine that the black areas, corresponding to non feasible regions, where $a_{i,j}$ and $b_{i+1,j}$ are located, are moved inside the unit cell. These critical values can be computed from the free space diagrams, and will require solving a degree four equation in ϵ since the boundaries of the corresponding polygons (elliptic or hyperbolic) are defined by degree two equations.

We claim that there is an upper bound of $O((z^2z' + zz'^2)n^4)$ on the number of critical values of ϵ, since each cell has $O(n^2)$ polygons and every two of them need to be tested in the computation of the third kind of critical values. In addition to this there are potentially $O(zz'n^2)$ critical values for ϵ as determined in the first two cases. Sorting the critical values first, then using the binary search, for each value of ϵ, we test whether we can obtain a monotone path in the free space diagram. The smallest ϵ, that results in a monotone path located in the free space diagram, is Fréchet distance between the paths Z and Z'. We summarize the result in the following theorem.

Theorem 3 *Fréchet-distance between two simple paths Z and Z' on the surface of a convex polyhedron, consisting of n triangular faces, can be computed in $O((z^2z' + zz'^2)n^4 \log(zz'n))$ time, where Z consists of z segments and Z' consists of z' segments.*

References

[1] H. Alt, M. Godau. Computing the Fréchet Distance Between Two Polygonal Curves. In *IJCGA*, 5, 1995, pp 75-91.

[2] M. Godau. A Natural Metric for Curves– Computing the Fréchet Distance for Polygonal Chains and Approximation Algorithms. In *STACS, LNCS*, 480, 1991, pp 127-136.

[3] H. Alt,A. Efrat, G. Rote, C. Wenk. Matching Planar Maps. In *J. of Algo.*, 49, 2003, pp 262-283

[4] H. Alt, C. Knauer, C. Wenk. Matching Polygonal Curves with respect to the Fréchet Distance. In *STACS*, 2001, pp 63-74.

[5] J. Chen, Y. Han. Shortest Paths on a Polyhedron. In *IJCGA*, 6, 1996, 127-144.

[6] Y. H.Hwang, R. C.Chang, H. Y.Tu. Finding all Shortest Path Edge Sequences on a Convex Polyhedron. In *WADS, LNCS*, 382, 1989 August, pp 252-266.

[7] M. Sharir, A. Schorr. On Shortest Paths in Polyhedral Spaces. In *SIAM J. of Comp.*, 15, 1986, pp 193-215.

[8] D. M.Mount. The Number of Shortest Path on the Surface of a Polyhedron. In *SIAM J. of Comp.* 19, 1990 August, pp 593-611.

[9] J. Yi. Computing Fréchet Distance of Two Paths on A Convex Polyhedra. In *M.C.S Thesis, Carleton University*, 2004 August.

[10] L. Aleksandrov, A. Maheshwari, J.-R.Sack. Determining approximate shortest paths on weighted polyhedral surfaces. In *JACM*, 50, 2005, pp 25-53.