Computing the Fréchet Distance between Piecewise Smooth Curves ⋆

Günter Rote

Freie Universität Berlin, Institut für Informatik, Takustraße 9, 14195 Berlin, Germany, rote@inf.fu-berlin.de.

Abstract

We consider the Fréchet distance between two curves which are given as a sequence of $m+n$ curved pieces. If these pieces are sufficiently well-behaved, we can compute the Fréchet distance in $O(mn\log(mn))$ time. The decision version of the problem can be solved in $O(mn)$ time.

1. Introduction

The Fréchet distance is a distance measure between curves.

Definition 1 (Fréchet distance)
Let $f\colon I=[l_I,r_I]\to\mathbb{R}^2$ and $g\colon J=[l_J,r_J]\to\mathbb{R}^2$ be two planar curves, and let $\|\cdot\|$ denote the Euclidean norm. Then the Fréchet distance $\delta_F(f,g)$ *is defined as*

$$\delta_F(f,g) := \inf_{\substack{\alpha\colon [0,1]\to I\\ \beta\colon [0,1]\to J}} \max_{t\in[0,1]} \|f(\alpha(t))-g(\beta(t))\|.$$

where α and β range over continuous and non-decreasing reparameterizations with $\alpha(0)=l_I$, $\alpha(1)=r_I$, $\beta(0)=l_J$, $\beta(1)=r_J$.

In contrast to other common distance measures like the Hausdorff distance, the Fréchet distance respects the one-dimensional structure of the curves and doesn't just treat them as a point set.

The study of the Fréchet distance from a computational point of view has been initiated by Alt and Godau [2]. The *decision problem* is the problem to decide, for a given ε, whether the Fréchet distance between two curves is at most ε.

Alt and Godau [2] treated the case of two polygonal curves. For two curves of m and n pieces, respectively, they showed how to solve the decision problem in $O(mn)$ time and the optimization problem in $O(mn\log(mn))$ time. Some related problems have also been considered, like minimizing the Fréchet distance under translations [3], or a generalized Fréchet distance between a curve and a *graph* [1]. In all cases, however, the objects are piecewise linear.

In this paper, we explore the Fréchet distance between more general curves. We assume that each input curve is given as a sequence of smooth curve pieces that are"sufficiently well-behaved", such as circular arcs, parabolic arcs, or some class of spline curves. Our algorithm will perform certain operations on these curves, like intersecting them with a circle.

We will show that the *combinatorial complexity*, i. e., the number of steps, for solving the decision problem is not larger than for polygonal paths, $O(mn)$. The complexity of the individual operations (the *algebraic complexity*) depends of course on the nature of the curves. Under the stronger assumption that the curves consist of algebraic pieces whose degree is bounded by a constant, we can solve the optimization problem in $O(mn\log(mn))$, thus matching the running time for the polygonal case. The elementary operations, however, are algebraic operations of higher degree.

We assume that each curve is given as a sequence of pieces which are connected at their endpoints. Every piece is a smooth curve of class C^2, i. e., the curvature is defined everywhere and varies continuously within a piece. We will not make any assumptions how the curves are given; it is only important that the necessary geometric operations

⋆ Partially supported by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-2000-26473 (ECG—Effective Computational Geometry for Curves and Surfaces).

can be carried out.

We need curves whose turning angle is bounded by π. Curves of larger bounding angle must be subdivided. For solving the decision problem with parameter ε, we subdivide the curve at all points where the curvature is $1/\varepsilon$, in order to ensure that in each piece of the curve, the curvature is either uniformly smaller or bigger than $1/\varepsilon$.

We have omitted most proofs, but we state one auxiliary lemma in order to illustrate the elementary arguments on which the results are based.

Lemma 2 *Let f be a smooth curve of turning angle at most π, and let c by a circle of radius r.*

(a) *If the curvature of f is at most $1/r$ everywhere, the curve can intersect c at most twice. If it intersects c twice, then its endpoints lie outside c or on the boundary, and the middle piece between the two intersections lies inside c.*

(b) *If the curvature of f is at least $1/r$ everywhere, the curve can intersect c at most twice.*

The full version of this paper is available as a technical report [5].

2. The Free Space Diagram

The main tool of the algorithm is the *free space diagram* which was introduced in [2]. It is a two-dimensional representation of all pairs of points on the two curves, together with the identification of those pairs which are closer than ε.

Definition 3 *Let $f\colon I \to \mathbb{R}^2$, $g\colon J \to \mathbb{R}^2$ be two curves, $I, J \subseteq \mathbb{R}$. The set*

$$F_\varepsilon(f,g) := \{\, (s,t) \in I \times J : \|f(s) - g(t)\| \le \varepsilon \,\}$$

denotes the free space *of f and g. the partition of $I \times J$ into the free space and its complement is called the* free space diagram.

Points in F_ε are called *feasible* or *free*, and they are usually drawn in white. The other points are called *forbidden points* or *obstacles*, see Figure 1. The following simple observation from [2] is crucial.

Lemma 4 *Let $f\colon I = [l_I, r_I] \to \mathbb{R}^2$, $g\colon J = [l_J, r_J] \to \mathbb{R}^2$ be two curves. Then $\delta_F(f,g) \le \varepsilon$ if and only if there exists a curve within $F_\varepsilon(f,g)$ from (l_I, l_J) to (r_I, r_J) which is monotone in both coordinates.* □

As f and g consist of several pieces, the free space diagram decomposes naturally into a grid of rectangular *cells*.

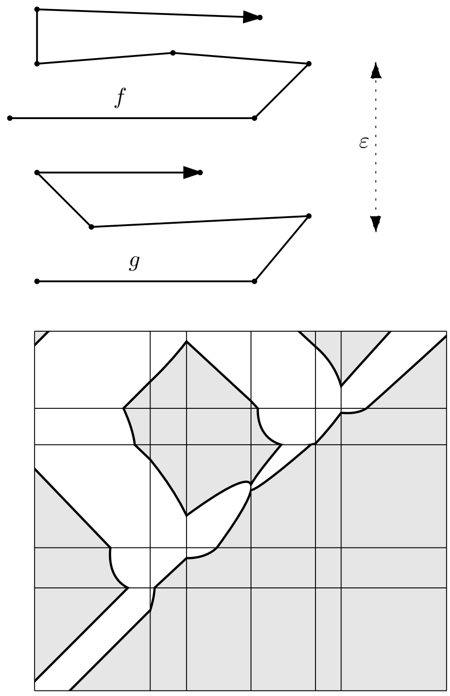

Fig. 1. Two polygonal curves and their free space diagram. The scale of the free space diagram is 50% reduced with respect to the curves.

3. Critical points

We regards as *critical points* on the boundary of F_ε those points which are local extrema in the horizontal or vertical direction. There are eight classes of critical points, shown in Figure 2.

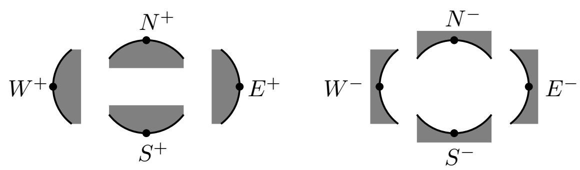

Fig. 2. The eight types of critical points. N, S, E, W refers to the direction in which the point is extreme, and the superscript tells whether the area in this direction is feasible (+) or forbidden (−).

In terms of the curves f and g, these points correspond to situations where a circle c of radius ε around a point of one curve is tangent to the other curve. For example, a critical point of type W^+ occurs in the situation where g touches the circle c of radius ε around a point x on f from inside. As x proceeds further away from g, a portion of g begins to stick out from c.

4. The structure of a single cell

The free space may be arbitrarily complicated even inside a cell. For example, if ε is very small, F_ε will contain isolated islands of free space for all intersections between f and g. However, we will show that the reachable points can be computed in a constant number of elementary geometric operations.

We have subdivided the curves, and consequently, the parameter intervals I and J into m and n pieces, respectively. Correspondingly, we cut the rectangle $I \times J$ into mn cells. On the boundaries of these cells, we compute all points which are *reachable* from the lower left corner (l_I, l_J) of the rectangle by a path in free space which is monotone in both directions. We do this incrementally from the lower left cell to the uppermost right cell.

A vertical line in the free-space diagram corresponds to a fixed point $f(s)$ on f. The points in F_ε on this line correspond to the points of g which lie inside a circle c of radius ε around $f(s)$. The boundary of F_ε corresponds to the intersections of c with g, and hence we can apply Lemma 2.

Lemma 5 *Inside a cell, a vertical or horizontal line intersects the boundary of F_ε at most twice.*

A vertical tangent line trough a critical point of type E or W or a horizontal tangent line trough a critical point of type N or S does not cross the boundary of F_ε in any other point. □

Lemma 6 *A curve forming a component of the boundary of the free space inside a cell can contain at most four critical points.* □

This lemma implies that there is a limited number of possibilities for such a boundary, the most complicated being an "s-shaped" path between between the left edge and the right edge of the rectangle, containing two critical points S^+ and N^-.

5. Processing a cell

We are given the reachable points on the left and bottom edge, and we compute the points on the right edge and on the top edge which are reachable from there.

On each edge of the rectangle there are at most two intervals of free points, by Lemma 5. Inside each interval of free points, there is only a single interval of reachable points because from every free point, everything which is to the right or to the top in the same free interval is reachable directly.

We will illustrate how to compute the *leftmost* reachable point in each free interval on the top edge from a given interval X on left edge. Other cases are similar.

We are given the lowest reachable point B in X. The upper end of X may be the upper left corner of the rectangle, or it may be a forbidden point which belongs to a component O of forbidden points. Similarly, the left endpoint F of Y may be part of a component of forbidden points, which we denote by O_2. (O and O_2 are not necessarily different, see Figure 3a.)

Lemma 7 *The leftmost point U in Y reachable from X depends only on the presence and the relative locations of O and O_2 and the horizontal line through B.*

PROOF. We have to show that any other "obstacles" of forbidden points do not play any role in this question. We show this by giving an algorithm for constructing U in all cases.

If the horizontal line through B intersects O or O_2, it is clear that one cannot reach Y, see for example the interval X_1 in Figure 3a or the interval X_2 in connection with Y_2 in Figure 3b. Otherwise, we claim that the desired point U lies directly above the rightmost point of O or of O_2, whichever is further to the right.

The monotone path from X to Y has to pass to the right of O and O_2. Thus, no point in Y left of U is reachable from X. To see that U is reachable, consider first the case that O exists, see the example of the interval X_1 in Figure 3b. Let A be the rightmost point of O. A can lie on the upper edge, or it can be a critical point of type E^+.

Assume first that A is a critical point of type E^+. The vertical line a through A lies completely in the free space, by Lemma 5, and O is the only obstacle left of a. By assumption, the horizontal line b from the lowest reachable point B in X does not intersect O before reaching a, and there are no other obstacles in this range. Thus, A is reachable from B, and the upper end A' of a is the leftmost reachable point on the top edge. If it lies in Y, we can take it as our point U, and we are done. (This is the case for the intervals X_1 and Y_1 in Figure 3b.) If Y lies left of a, we are done as well, as no points in Y are reachable from X. So let us deal with the

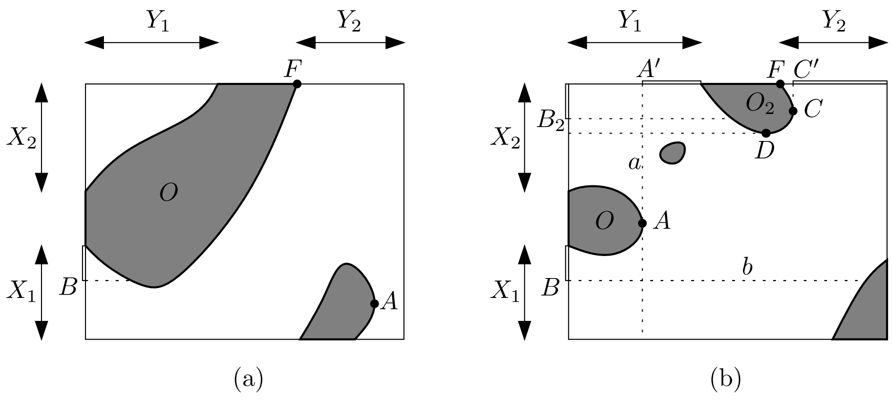

Fig. 3. Determining the reachable points on the top edge

only remaining case that Y lies to the right of a, and a is separated from Y by the obstacle O_2.

The lowest point D of O_2 must lie above O, and the horizontal line through D intersects a, which is reachable. Therefore D is reachable. From D we can reach the rightmost point C of O_2, which is either a critical point of type E^+ or the point F. In either case, we can indeed reach the point C' vertically above C as the leftmost point U.

The cases when O does not exist, or when the rightmost point A of O lies on the upper edge, can be treated similarly. □

Will have described our procedure in terms of geometric operations in the free space diagram, like finding the right-most point in a component of forbidden points. By working out what these operations mean in terms of the curves f and g, we obtain the following theorems.

Theorem 8 *Given the reachable points on the bottom edge and the left edge of a cell, the reachable points on the top edge and the right edge of the cell can be computed in a constant number of the following operations:*

- *Intersecting a circle of radius ε with one of the curves*
- *Finding the first intersection of one curve with an offset curve of the other curve at distance ε.*

In both cases, we must be able to find the parameter values on the respective curves, corresponding to the points that we have computed. □

Theorem 9 *Given two curves consisting of m and n pieces, respectively, where each piece has a turning angle at most π and has curvature $\geq \varepsilon$ or $\leq \varepsilon$ throughout, we can decide $O(m+n)$ space and in $O(mn)$ primitive operations of the type described in Theorem 8 whether their Fréchet distance is at most ε, for a given parameter ε.* □

6. The Minimization Problem

The minimization problem of *computing* the Fréchet distance can be solved by Megiddo's parametric search technique [4], closely following the approach of [2] for polygonal curves. The technical details are more involved, and we have to make some stronger assumptions on the curves.

Theorem 10 *Given two curves consisting of m and n pieces, respectively, of smooth algebraic curves of fixed maximum degree we can compute their Fréchet distance in $O(nm)$ space and in $O(mn \log(mn))$ algebraic operations, i.e., degree comparison between two real solutions of algebraic equations of bounded degree.* □

References

[1] H. Alt, A. Efrat, G. Rote, and C. Wenk, *Matching planar maps*, Journal of Algorithms **49** (2003), 262–283.

[2] H. Alt and M. Godau, *Computing the Fréchet distance between two polygonal curves*, Internat. J. Comput. Geom. Appl. **5** (1995), 75–91.

[3] H. Alt, C. Knauer, and C. Wenk, *Matching polygonal curves with respect to the Fréchet distance*, STACS 2001 (A. Ferreira and H. Reichel, eds.), Lect. Notes Comp. Sci., vol. 2010, Springer-Verlag, 2001, pp. 63–74.

[4] N. Megiddo, *Applying parallel computation algorithms in the design of serial algorithms*, J. Assoc. Comput. Mach. **30** (1983), 852–865.

[5] Günter Rote, *Computing the fréchet distance between piecewise smooth curves*, Tech. Report ECG-TR-241108-01, 2003.