Simplifying 3D Polygonal Chains Under the Discrete Fréchet Distance*

Sergey Bereg[1], Minghui Jiang[2], Wencheng Wang[3], Boting Yang[4], and Binhai Zhu[5]

[1] Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA
besp@utdallas.edu
[2] Department of Computer Science, Utah State University, Logan, UT 84322-4205, USA
mjiang@cc.usu.edu
[3] Institute of Software, Chinese Academy of Sciences, Beijing 100080, China
whn@ios.ac.cn
[4] Department of Computer Science, University of Regina, Regina,
Saskatchewan, S4S 0A2, Canada
boting@cs.uregina.ca
[5] Department of Computer Science, Montana State University, Bozeman, MT 59717-3880, USA
bhz@cs.montana.edu

Abstract. A well-known measure to characterize the similarity of two polygonal chains is the famous Fréchet distance. In this paper, for the first time, we consider the problem of simplifying 3D polygonal chains under the discrete Fréchet distance. We present efficient polynomial time algorithms for simplifying a single chain, including the first near-linear $O(n \log n)$ time exact algorithm for the continuous min-# fitting problem. Our algorithms generalize to any fixed dimension $d > 3$. Motivated by the ridge-based model simplification we also consider simplifying a pair of chains simultaneously and we show that one version of the general problem is NP-complete.

1 Introduction

Simplifying polygonal chains is a well-studied problem, especially in the plane (and occasionally in 3D and higher dimensional spaces). In short, the problem is to simplify a given chain A with n vertices into A' such that A and A' are close and $|A'| \ll n$. For instance, in 3D we face the problem of simplifying optic nerves in medical studies and simplifying river networks in GIS [21]. Most of the previous researches are focused on simplifying 2D polygonal chains [6,7,10,14,15,16,20,22], with the notable exception of [10,4]. Readers are referred to [4] for a list of complete references on simplifying polygonal chains in all dimensions. In this paper, we first follow the traditional work on simplifying a polygonal chain (a polyline or simply a chain) in 3D, but under a relatively new measure — the discrete Fréchet distance.

Fréchet distance was first defined by Maurice Fréchet in 1906 [11]. While known as a famous distance measure in the field of mathematics (more specifically, abstract

---

* This research is supported by the NSERC grant 261290-03 and grant A13501 at Utah State University.

E.S. Laber et al. (Eds.): LATIN 2008, LNCS 4957, pp. 630–641, 2008.

spaces), it was Alt and Godau who first applied it in measuring the similarity of polygonal curves in early 1990s [2,3].

In 1994, Eiter and Mannila defined the *discrete Fréchet distance* between two polygonal chains A and B (in any fixed dimensions) [9]. Recently, Jiang, Xu and Zhu applied the discrete Fréchet distance in aligning the backbones of proteins (which is called the *protein structure-structure alignment* problem) [17]. In fact, in this application the discrete Fréchet distance makes more sense as the backbone of a protein is simply a polygonal chain in 3D, with each vertex being the alpha-carbon atom of a residue. So if the (continuous) Fréchet distance is realized by an alpha-carbon atom and some other point which does not represent an atom, it is not meaningful biologically. Jiang, *et al.* showed that given two planar polygonal chains the minimum discrete Fréchet distance between them, under both translation and rotation, can be computed in polynomial time. They also applied some ideas therein to design an efficient heuristic for the original protein structure-structure alignment problem in 3D.

Very recently, the discrete Fréchet distance was used to align protein backbones locally. It was shown that given many proteins finding such a local alignment is NP-complete, but when a constant number of chains are given then the problem is polynomially solvable [24]. Notice that finding local alignment between two proteins (or 3D chains) A, B is different from simplifying them. Loosely speaking, a local alignment is to find a subsequence A' of A and a subsequence B' of B such that A' and B' are very close. But A' and A (hence B' and B) could have a huge difference.

While one can claim that the discrete Fréchet distance is a special case of the (continuous) Fréchet distance, the use of discrete Fréchet distance, in many situations, makes more sense. Firstly, the discrete Fréchet distance is more efficient to compute. For instance, Godau used the Fréchet distance to approximate polygonal chains using vertices of the original curve [12] (i.e., *discrete* fitting in our terminology). The running time of his algorithms are $O(n^3)$ for min-# fitting and $O(n^4 \log n)$ for min-ϵ fitting, while using the discrete Fréchet distance these bounds are $O(n^2)$ and $O(n^3)$ respectively. Secondly, as we just mentioned, in many biological applications (continuous) Fréchet distance does not make any sense.

Now coming back to the second motivation of our research — ridge-based geometric model simplification. A ridge is a critical 3D polygonal chain on a surface whose projection on the XY-plane is a simple (planar) polygonal chain. Ridge simplification and approximation is an interesting problem in geometric modeling, approximation and 3D geometric compression. We refer to Fig. 1 for an example. We have identified two ridges P and Q and wish to simplify them into P' and Q' so as to have a simplified surface between P' and Q'. In this case, however, we not only want P and P' (Q and Q') to be close, but also want that P' and Q' are close. Otherwise, as can be seen from Fig. 1 (II), the large discrete Fréchet distance between P', Q' induces some long skinny triangle anchored at the vertex y. On the other hand, when we simplify P into P'' such that P'' and Q' have a smaller discrete Fréchet distance then the long skinny triangle disappears (Fig. 1 (III) and (IV)).

It turns out that this problem of simultaneously simplifying a pair of chains is much more difficult than the protein local alignment problem. We show that a special case, where we measure the similarity between P, P' (and between Q, Q') using the

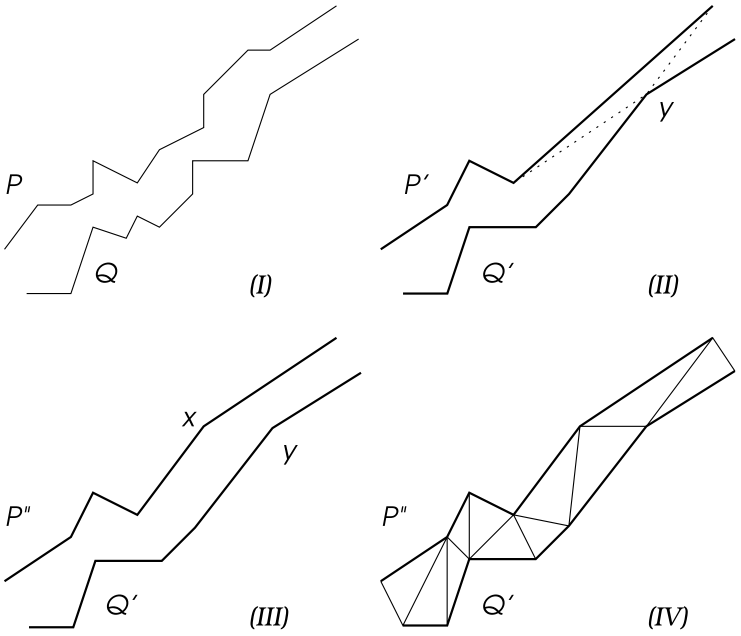

Fig. 1. Large discrete Fréchet distance implies long skinny triangles

Hausdroff distance between vertices while measuring the similarity between P', Q' using the discrete Fréchet distance, is NP-complete. This implies that we should better add Steiner points in this application, which is a popular way to improve the quality of a mesh.

2 Preliminaries

Given two polygonal chains A, B with $|A| = k$ and $|B| = l$ respectively, we aim at aligning the similarity of A and B (sometimes under translation and rotation) such that their distance is minimized under certain measure. Among the various distance measures, the Hausdorff distance is known to be better suited for matching two point sets than for matching two polygonal chains; the (continuous) Fréchet distance is a superior measure for matching two polygonal chains, but it is not quite easy to compute especially when translation/rotation are allowed.

Let X be the Euclidean space R^3; let $d(a, b)$ denote the Euclidean distance between two points $a, b \in X$. The (continuous) Fréchet distance between two parametric curves $f : [0, 1] \to X$ and $g : [0, 1] \to X$ is

$$\delta_{\mathcal{F}}(f, g) = \inf_{\alpha,\beta} \max_{s\in[0,1]} d(f(\alpha(s)), g(\beta(s))),$$

where α and β range over all continuous non-decreasing real functions with $\alpha(0) = \beta(0) = 0$ and $\alpha(1) = \beta(1) = 1$.

Imagine that a person and a dog walk along two different paths while connected by a leash; they always move forward, though at different paces. The minimum possible length of the leash is the Fréchet distance between the two paths. To compute the Fréchet distance between two polygonal curves A and B (in the Euclidean plane) of $|A|$ and $|B|$ vertices, respectively, Alt and Godau [2] presented an $O(|A||B|\log^2(|A||B|))$ time algorithm. Later this bound was reduced to $O(|A||B|\log(|A||B|))$ time [3].

We now define the discrete Fréchet distance following [9].

Definition 1. *Given a polygonal chain (polyline) in 3D $P = \langle p_1, \ldots, p_k \rangle$ of k vertices, an m-**walk** along P partitions the path into m (disjoint) non-empty subchains $\{\mathcal{P}_i\}_{i=1..m}$ such that $\mathcal{P}_i = \langle p_{k_{i-1}+1}, \ldots, p_{k_i} \rangle$ and $0 = k_0 < k_1 < \cdots < k_m = k$.*

*Given two 3D polylines $A = \langle a_1, \ldots, a_k \rangle$ and $B = \langle b_1, \ldots, b_l \rangle$, a **paired walk** along A and B is an m-walk $\{\mathcal{A}_i\}_{i=1..m}$ along A and an m-walk $\{\mathcal{B}_i\}_{i=1..m}$ along B for some m, such that, for $1 \le i \le m$, either $|\mathcal{A}_i| = 1$ or $|\mathcal{B}_i| = 1$ (that is, either $\mathcal{A}_i$ or $\mathcal{B}_i$ contains exactly one vertex). The **cost** of a paired walk $W = \{(\mathcal{A}_i, \mathcal{B}_i)\}$ along two paths A and B is*

$$d_F^W(A,B) = \max_i \max_{(a,b)\in\mathcal{A}_i\times\mathcal{B}_i} d(a,b).$$

*The **discrete Fréchet distance** between two polylines A and B is*

$$d_F(A,B) = \min_W d_F^W(A,B).$$

*The paired walk that achieves the discrete Fréchet distance between two paths A and B is also called the **Fréchet alignment** of A and B.*

Consider the scenario in which the person walks along A and the dog along B. Intuitively, the definition of the paired walk is based on three cases:

1. $|\mathcal{B}_i| > |\mathcal{A}_i| = 1$: the person stays and the dog moves forward;
2. $|\mathcal{A}_i| > |\mathcal{B}_i| = 1$: the person moves forward and the dog stays;
3. $|\mathcal{A}_i| = |\mathcal{B}_i| = 1$: both the person and the dog move forward.

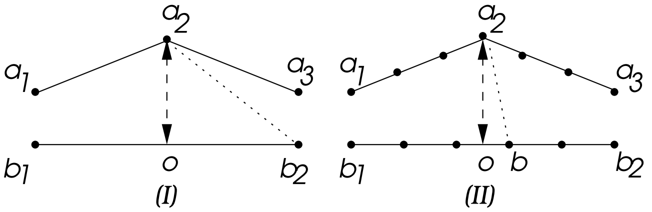

Fig. 2. The relationship between the discrete and continuous Fréchet distances

Eiter and Mannila presented a simple dynamic programming algorithm to compute $d_F(A,B)$ in $O(|A||B|) = O(kl)$ time [9]. The recent result of Jiang, *et al.* shows that in 3D the minimum discrete Fréchet distance between A and B under translation can be computed in $O(k^4 l^4 \log(k+l))$ time, and under both translation and rotation it can be computed in $O(k^7 l^7 \log(k+l))$ time using the ideas presented in [23,17]. They are significantly faster than the corresponding bounds for the continuous Fréchet distance. In 3D, Wenk showed that given two chains with sum of length $N = k + l$, the minimum Fréchet distance between them can be computed in $O(N^{3f+2} \log N)$ time, where f is the degree of freedom for moving the chains [23]. So with translation alone this minimum Fréchet distance can be computed in $O(N^{11} \log N)$ time, and when both translation and rotation are allowed the corresponding minimum Fréchet distance can be computed in $O(N^{20} \log N)$ time [23].

We comment that while the discrete Fréchet distance could be arbitrarily larger than the corresponding continuous Fréchet distance (e.g., in Fig. 2 (I), they are $d(a_2, b_2)$ and $d(a_2, o)$ respectively), by adding sample points on the polylines, one can easily obtain a close approximation of the continuous Fréchet distance using the discrete Fréchet distance (e.g., one can use $d(a_2, b)$ in Fig. 2 (II) to approximate $d(a_2, o)$). This fact was also pointed out in [9]. Moreover, the discrete Fréchet distance is a more natural measure for matching the geometric shapes of biological sequences such as proteins. As we mentioned in the introduction, in such applications, the continuous Fréchet distance does not make much sense to biologists.

3 Min-# Fitting with a Given Error Bound

In this section, we discuss min-# fitting (simplification) with a given error bound; namely, given a chain A and an error bound δ, we want to simplify A into another chain C with the minimum number of vertices such that $d_F(A, C) \le \delta$. This is a traditional problem on polygonal chain simplification, except that almost all the previous work are all focused on different measures, for instance, the ϵ-*tolerance zone* error measure [4]. With the (continuous) Fréchet error measure, for the 2D problem, Guibas, *et al.* obtained an $O(n^2 \log^2 n)$ time algorithm [13]. We show that in 3D this problem can be solved in $O(n \log n)$ time using the discrete Fréchet error measure.

Let $A = A[1..n]$ be the given chain A of n vertices. Let $A[i..j]$ be the (contiguous) subchain of A starting from the index i to the index j. We call $A[1..i]$ a prefix of A. Let $A \circ B$ be the concatenation of two chains A and B (by connecting the last vertex of A and the first vertex of B).

Given a discrete Fréchet distance (error) δ, we wish to simplify $A = A[1..n]$ using a simple chain C such that $d_F(A, C) \le \delta$ and the size of C is minimized. Apparently, we have two cases; the vertices of C could be arbitrary or could only be the vertices of A. We call them the *continuous* and *discrete* cases respectively. It turns out that the two cases can be solved differently, with the greedy method and dynamic programming respectively. We cover the continuous case first.

For the continuous case, we can see that following the definition of the discrete Fréchet distance, the paired walk between A and C, $\mathcal{A}_i$ and $\mathcal{C}_i$, must satisfy the property that $|\mathcal{A}_i| \ge |\mathcal{C}_i| = 1$ for all i (otherwise, we can simply delete some vertices in C to obtain a better simplification). Then, if $|\mathcal{A}_i| \ge |\mathcal{C}_i| = 1$ for all i but $\mathcal{A}_i$ is not maximal, we can merge $\mathcal{A}_i$ with a prefix y of $\mathcal{A}_{i+1}$ which is at most distance δ away from $\mathcal{C}_i$ to obtain a new $\mathcal{A}'_i = \mathcal{A}_i \circ y$, without affecting the size of C.

So we can use a greedy method to find the first *breakpoint* (the largest index j) such that (all the vertices on) $A[1..j]$ can be covered by a ball centered at a point b_j with radius δ, $\mathcal{B}(b_j, \delta)$, but $A[1..j+1]$ cannot be covered by any ball of radius δ. Given $A[1..j]$, we can decide whether it can be covered by $\mathcal{B}(b_j, \delta)$ in $O(n)$ time. In fact, one can simply compute the smallest enclosing ball for the vertices of $A[1..j]$ to locate the point b_j in linear time [19]. So b_j will be the first vertex on the simplified chain C. Repeating this greedy process at most $m^* = O(n)$ times, where m^* is the optimal solution value for the problem, we can obtain a chain C with m^* vertices. Because we are in 3D, a simple perturbation on the vertices of A can easily eliminate any self-intersection in C. It is easy to see that this greedy method solves the continuous min-# fitting problem in $O(n^2)$ time.

We can use binary search to repeatedly find the breakpoints, so the problem can in fact be solved in $O(m^* n \log n)$ time (which seems tough to beat at the first sight). However, we present the following Algorithm $CMN(A[1..n], \delta)$ which solves the problem in $O(n \log n)$ time. We use the *doubling search* method, which has been used before in [18,5,1].

Algorithm. $CMN(A[1..n], \delta)$

(1) Search with $t = 1, 2, 3, ...$ the first t such that $A[1..2^{t-1}]$ can be covered by a ball with radius δ but $A[1..2^t]$ cannot. Then find the first breakpoint k_1 in $A[2^{t-1}..2^t]$ using binary search.

(2) Repeat the above process on $A[k_1{+}1..n]$ to compute all of the m^*-1 breakpoints.

Theorem 1. *The CMN procedure solves the continuous min-# fitting problem for a 3D polyline under the discrete Fréchet distance in* $O(n \log n)$ *time and* $O(n)$ *space.*

Proof. Clearly k_1 can be found in $O(k_1 \log k_1)$ time. This is due to that when k_1 is in $A[2^{t-1}..2^t]$, then 2^t is at most $2k_1$. Let $n_1, n_2, ..., n_{m^*}$ be the sizes of the subchains determined by the $m^* - 1$ breakpoints (note that $n_1 = k_1$). The overall running time of CMN is

$$O(n_1 \log n_1) + O(n_2 \log n_2) + \cdots + O(n_{m^*} \log n_{m^*}),$$

which is $O(n \log n)$, due to $n_1 + n_2 + \cdots + n_{m^*} = n$. □

We remark that the greedy method in Theorem 1 is similar to that in [4,1]. In [1], Agarwal, *et al.* considered the similar discrete problem of min-# fitting (simplification) with a given continuous Fréchet error. An approximation algorithm, which returns at most twice the size of an optimal simplified curve within half of the error, was presented in [1]. An open question on a better near-linear time approximation was also raised in [1]. The above theorem shows that, using the discrete Fréchet distance, the continuous version of the problem can be solved exactly with a near-linear time algorithm. For the same discrete problem, we will use the discrete Fréchet measure and present an $O(n^2)$ time solution to solve it exactly.

Regarding the discrete min-# fitting problem under the discrete Fréchet distance, i.e., when the vertices of the simplified chain, C', must come from A, it turns out that there is a dramatic difference compared with the continuous case. We refer to Fig. 3, in which we have a chain A with five vertices, and when $\delta = 1$ the optimal simplified chain

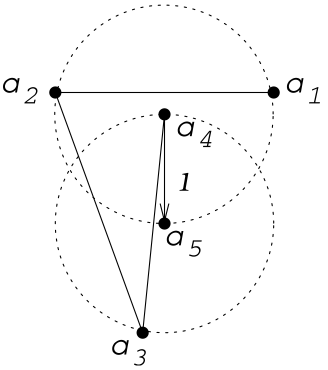

Fig. 3. Given a chain A, when $\delta = 1$, the optimal simplified chain is $\langle a_4, a_5 \rangle$

is $C' = \langle a_4, a_5 \rangle$. In other words, $A[1..2]$ is not covered by a vertex on the subchain $A[1..2]$. This is also completely different from the situation in [4].

Due to the unfavorable non-local property of the discrete problem, we solve the problem using a dynamic programming method. Without loss of generality, we only show how to compute the optimal size of C'. The actual chain C' can be constructed easily by modifying the algorithm.

Define $T[i, s]$ as the maximum index $j, j \geq i$, such that the ball with radius δ and centered at $A[s]$ covers $A[i..j]$. $T[i, s] = i - 1$ if $d(A[i], A[s]) > \delta$; otherwise, $T[i, s] = T[i + 1, s]$. For each s, $T[-, s]$ can be computed in $O(n)$ time. So $T[-, -]$ can be computed in $O(n^2)$ time.

Define $N[i, s]$ as the minimum number of balls with radius δ and centered at $A[s..n]$ that cover $A[i..n]$. We have

$$N[i, s] = \min\{N[i, s + 1], N[j + 1, s + 1] + 1\},$$

where $j = T[i, s]$. The boundary cases when $i = n$ or $s = n$ can be handled easily. So $N[-, -]$ can be computed in $O(n^2)$ time and space.

Theorem 2. *The discrete min-# fitting problem for a 3D polyline under the discrete Fréchet distance can be solved in $O(n^2)$ time and $O(n^2)$ space.*

We remark that using the continuous Fréchet distance the discrete min-# fitting problem can be solved in $O(n^3)$ time [12]. In [1] whether this bound can be reduced was listed as an open problem. Our above theorem shows again that using the discrete Fréchet distance the problem can be solved more efficiently, in fact, in quadratic time.

4 Min-ϵ Fitting with m-chains

The *min-ϵ fitting with m-chains* problem is defined as follows. Given a 3D chain $A = \langle a_1, a_2, \ldots, a_n \rangle$ and a positive integer m, we wish to simplify A into a polyline $B = \langle b_1, b_2, \ldots, b_m \rangle$ such that $d_F(A, B)$ is minimized. Again, we have two cases: the *continuous* case (when the vertices of B are arbitrary ones) and the *discrete* case (when the vertices of B must come from A). We first show how to solve the continuous version of the problem using CMN as a subroutine.

Be reminded that in the continuous case, the vertices of B do not have to come from A. We first design a procedure $CME(A[i..j], m)$ which covers $A[i..j]$ with m balls of the smallest radius. Let $\delta(i, j)$ be the radius of the smallest ball covering $A[i..j]$.

Algorithm. $CME(A[i..j], m)$

```
(1) If $i \geq j$ then return 0.
(2) If $m = 1$ then return $\delta(i, j)$.
(3) Find $k$ such that $CMN(A[i..j], \delta(i, k)) > m \geq CMN(A[i..j], \delta(i, k+1))$
and return $\min\{\delta(i, k+1), CME(A[k+1..j], m-1)\}$.
```

Apparently CME is a recursive procedure and we call $CME(A[1..n], m)$ the first time. We have the following theorem.

Theorem 3. *The continuous min-ϵ fitting problem under the discrete Fréchet distance can be solved in $O(mn \log n \log(n/m))$ time.*

Proof. We first sketch the correctness proof of CME. Recall that $\delta(i, j)$ is the radius of the smallest enclosing ball of $A[i, j]$. Denote by $\delta(i, j, k)$ the minimum radius of k uniform balls covering $A[i, j]$. The following properties are not difficult to prove.

Let x be the smallest index such that $A[x+1, n]$ can be covered by $m-1$ balls of radius $\delta(1, x)$; that is, $\delta(x, n, m-1) > \delta(1, x-1)$ and $\delta(x+1, n, m-1) \leq \delta(1, x)$. Then we have the recursion

$$\delta(1, n, m) = \min\{\delta(1, x), \delta(x, n, m-1)\}.$$

This corresponds to two cases in CMN: (1) Cover $A[1, x]$ with a ball of radius $\delta(1, x)$ and $A[x+1, n]$ with $m-1$ balls of radius at most $\delta(1, x)$; and (2) cover $A[1, x-1]$ with a ball of radius $\delta(1, x-1)$ and $A[x, n]$ with $m-1$ balls of radius $\delta(x, n, m-1)$.

Note that as we use CMN, which takes $O(n \log n)$ time, as a subroutine and we have to recurse CME m times, the crucial question is how to find k quickly at each recursion. A naive binary search would find each k in $O(\log n)$ time hence giving us a total running time of $m \times O(\log n) \times O(n \log n) = O(nm \log^2 n)$ time. However, we can use the same doubling search idea in Theorem 1 so that at the i-th recursion the corresponding k_i can be found in $O(\log n_i \times n \log n)$ time, for $i = 1, 2, ..., m$, where n_i is the size of the subchain covered by the i-th vertex of B (with optimal radius/error). So the running time of the algorithm is

$$\sum_{1 \leq i \leq m} O(\log n_i \times n \log n),$$

which is $O(mn \log n \log(n/m))$, due to that $\sum_{1 \leq i \leq m} n_i = n$. □

We now consider the discrete case, i.e., the vertices of B must come from A. In this problem, following Fig. 3 (when $m = 2$) we can again see that in the optimal solution $A[1..i]$ is not necessarily covered by a ball centered at a vertex on $A[1..i]$. Similar to the discrete min-# fitting problem, we again follow the dynamic programming method.

Define $R[i, j, s]$ as the minimum radius of a ball centered at $A[s]$ that covers $A[i..j]$. We have $R[i, j, s] = \max\{d(A[i], A[s]), R[i + 1, j, s]\}$. So $R[-, j, s]$ can be computed in $O(n)$ time and the whole table $R[-, -, -]$ can be computed in $O(n^3)$ time.

Define $E[i, s, z]$ as the minimum radius of z uniform balls centered at $A[s..n]$ that cover $A[i..n]$. Define $J[i, s, z]$ as the minimum index $j \geq i$ such that $R[i, j, s] \geq E[j + 1, s + 1, z - 1]$. $E[i, s, z]$ can be updated in two cases: (1) s is used as a center for a uniform ball, and (2) s is not used as a center for a uniform ball. Therefore,

$$E[i, s, z] = \min\{E[i, s + 1, z], R[i, j, s], E[j, s + 1, z - 1]\},$$

where $j = J[i, s, z]$. Again, the boundary cases when $i = n$ or $s = n$ or $z = 0$ can be handled easily.

$E[-, -, -]$ can be computed in $O(mn^2)$ time given $J[-, -, -]$. Note that $E[-, -, z]$ depends on $J[-, -, z]$, and that $J[-, -, z]$ depends on $E[-, -, z - 1]$. Therefore, for each z from 1 to m, we need to compute $J[-, -, z]$ before $E[-, -, z]$.

To compute $J[i - 1, s, z]$, compare $d(A[i - 1], A[s])$ with $R[i, j, s]$, where $j = J[i, s, z]$. If $d(A[i - 1], A[s]) < R[i, j, s]$, then set $J[i - 1, s, z]$ to j. Otherwise, use a sequential search to find the minimum $j' \leq j$ such that $R[i - 1, j', s] \geq E[j' + 1, s + 1, z - 1]$, then set $J[i - 1, s, z]$ to j'. The time is $O(j - j' + 1)$ for filling each $J[i - 1, s, z]$, which adds up to $O(n)$ for $J[-, s, z]$. So $J[-, -, -]$ can be computed in $O(mn^2)$ time. The total running time for constructing $E[-, -, -]$ is $O(n^3) + O(mn^2) = O(n^3)$.

Theorem 4. *The discrete min-ϵ fitting problem under the discrete Fréchet distance can be solved in $O(n^3)$ time and $O(n^3)$ space.*

We comment that the running times in Theorem 1, Theorem 2 (when $m = o(n)$), Theorem 3 are all much faster than the corresponding ones for the ϵ-tolerance zone metric [4]. This might due to the strong 'ordering' property of the discrete Fréchet distance. However, we show in the next section that when we have to simplify a pair of chains simultaneously under the discrete Fréchet distance, one version of the general problem is even NP-complete. No such negative result is known, on any distance measure, in the previous research on chain simplification.

5 Simplifying a Pair of Chains Under the Discrete Fréchet Distance

As we have discussed in the introduction, in this section we investigate the problem of simplifying a pair of chains A, B into A', B' such that the vertices of A', B' must come from A, B respectively, $d'(A, A'), d'(B, B'), d_F(A', B')$ are all bounded. We will show that when $d'(-, -)$ is the Hausdorff distance between the vertices of two chains (denoted as $d_H(-, -)$ henceforth) then the problem for general 3D chains is NP-complete. This indicates that for ridge-based model simplification, we should use Steiner points to ensure the quality of the simplified surface.

Formally, the Chain Pair Simplification (CPS) problem is defined as follows.

Instance: Given a pair of 3D chains A and B in 3D, each with length $O(n)$, an integer K, and three real numbers $\delta_1, \delta_2, \delta_3$.

Problem: Does there exist a pair of chains A', B' each of at most K vertices such that the vertices of A', B' are from A, B respectively, and $d_1(A, A') \leq \delta_1, d_2(B, B') \leq \delta_2, d_F(A', B') \leq \delta_3$?

When $d_1 = d_2 = d_H$, we call the corresponding problem CPS-2H and when $d_1 = d_2 = d_F$, we call the corresponding problem CPS-3F. We have the following theorem.

Theorem 5. *The CPS-2H problem is NP-complete.*

Proof. It is easy to see that CPS-2H belongs to NP. We now reduce 3SAT to the CPS-2H problem. The idea of this reduction is from [8], even though over here we are handling a geometric problem.

Let $\phi = F_1 \bigwedge F_2 \bigwedge \cdots \bigwedge F_m$ be a conjunctive normal form, where each sub-formula F_i is a 3-disjunctive clause like $(x_2 \bigvee x_5 \bigvee \neg x_7)$. Assume that $x_1, x_2, \cdots, x_n$ are the boolean variables in the formula ϕ and each F_i cannot contain both x_k and $\neg x_k$ (otherwise F_i is already true and can be discarded). We construct a triple of points for each F_i as $p_{i1} = (i, i^2, 0), p_{i2} = (i, i^2, \epsilon), p_{i3} = (i, i^2, 2\epsilon)$, for some $0 < \epsilon < 0.1$. We then construct two chains A and B each with $4n-1$ vertices such that ϕ is satisfiable iff A and B can be simplified into A', B' each with $K = 2n-1$ vertices such that $d_H(A, A') \leq 2\epsilon$, $d_H(B, B') \leq 2\epsilon$ and $d_F(A', B') = 0$ (i.e., $\delta_3 = 0$ in our construction).

First we construct $n-1$ points $q_j = (j, 0, 0), 1 \leq j \leq n-1$. For each variable x_i in ϕ, we construct two sequences S_i and S_i^*. Let $F_{i_1}, \cdots, F_{i_u}$ be the clauses in ϕ that contain x_i, and let $F_{j_1}, \cdots, F_{j_v}$ be the clauses of ϕ that contain $\neg x_i$. Let $S_i = F_{i_1} \cdots F_{i_u} F_{j_1} \cdots F_{j_v}$ and $S_i^* = F_{j_1} \cdots F_{j_v} F_{i_1} \cdots F_{i_u}$. We next convert S_i (S_i^*) into a sequence of 3D points T_i (T_i^*), where each occurrence of $F_k (1 \leq k \leq m)$ in S_i or S_i^* corresponds to a unique point in $\{p_{kj} | j \leq 3\}$. Note that since F_k contains 3 literals, it appears in all S_i and S_i^* exactly three times. So from now on we assume that the three occurrences of p_{kj}'s are always in the order p_{k1}, p_{k2} and p_{k3} and with this in mind we will use p_k to simplify the presentation.

Let $A = \langle T_1, q_1, T_2, q_2, \cdots, q_{n-1}, T_n \rangle$ and $B = \langle T_1^*, q_1, T_2^*, q_2, \cdots, q_{n-1}, T_n^* \rangle$.

Assume that $x_1 = b_1, \cdots, x_n = b_n$ are assignments that make ϕ true. If $b_i = 1$, simplify both T_i and T_i^* to $T_i' = p_{i_1}, \cdots, p_{i_u}$ and $T_i^{*'} = p_{i_1}, \cdots, p_{i_u}$, respectively. If $b_i = 0$, simplify both T_i and T_i^* to $T_i' = p_{j_1}, \cdots, p_{j_v}$ and $T_i^{*'} = p_{j_1}, \cdots, p_{j_v}$, respectively. It is easy to see that $A' = \langle T_1', q_1, T_2', \cdots, T_{n-1}', q_{n-1}, T_n' \rangle$ is the same as $B' = \langle T_1^{*'}, q_1, T_2^{*'}, \cdots, T_{n-1}^{*'}, q_{n-1}, T_n^{*'} \rangle$ except that some p_j in T_i' are at most 2ϵ distance away. It is easy to see that $d_H(A, A') \leq 2\epsilon, d_H(B, B') \leq 2\epsilon$ and $d_F(A', B') = 0$; moreover, $K = 2n - 1$.

Assume that A is simplified into A'' and B is simplified into B'' via removing some points in $\{p_{ij} | 1 \leq i \leq n, 1 \leq j \leq 3\}$ such that $d_H(A, A'') \leq 2\epsilon$, $d_H(B, B'') \leq 2\epsilon$, and $d_F(A'', B'') = 0$. Notice that the distance between p_{ij} and p_{kl} and the distance between q_i and q_k are at least one, as long as $i \neq k$. The condition that $d_H(A, A'') \leq 2\epsilon$, $d_H(B, B'') \leq 2\epsilon$ implies that we can only remove points in $\{p_{ij} | 1 \leq i \leq n, 1 \leq j \leq 3\}$ and we must leave at least one point in A'', B'' for each $p_{ij}, 1 \leq j \leq 3$. As F_i cannot contain both x_k and $\neg x_k$, on the subchain between q_r and q_{r+1} on A or B there is exactly one point p_z, for some z. Finally, as $K = 2n - 1$, to make $d_F(A'', B'') \leq 2\epsilon$, we must leave all q_s's and leave exactly one point in A'', B'' for each $p_{ij}, 1 \leq j \leq 3$.

Let $T_i^{''}$ and $T_i^{*''}$ be the subchains in A'' and B'' which are obtained from simplifying T_i and T_i^* in A and B respectively. If $T_i^{''}$ is empty then we can assign a value to x_i arbitrarily. Now we focus on the case when $T_i^{''}$ is not empty, which implies that $T_i^{''}$ and $T_i^{*''}$ have the same size and $d_F(T_i^{''}, T_i^{*''}) \leq 2\epsilon$. If $T_i^{''}$ is not empty and it is a subsequence of $p_{i_1}, \cdots, p_{i_u}$ then we assign $x_i = 1$. If $T_i^{''}$ is not empty and it is a subsequence of $p_{j_1}, \cdots, p_{j_v}$ then we assign $x_i = 0$. It is easy to see that ϕ is true by the assignments to those variables $x_1, \cdots, x_n$.

To conclude the proof of this theorem, notice that the reduction takes linear (in the length of ϕ) time. □

We comment that for several *optimization* versions of the problem the proof still holds. For instance, when all the other conditions hold and we try to minimize K, then the problem is still NP-complete. Moreover, as in the above proof deciding whether $d_F(A', B') = 0$ is NP-complete, when all other conditions hold, there is no polynomial time algorithm for approximating $d_F(A', B')$ unless P=NP. The above theorem certainly implies that it is better to add Steiner points when we simplify a pair of (adjacent) ridges in ridge-based geometric model simplification.

6 Concluding Remarks

In this paper, for the first time, we study the problem of simplifying/approximating polylines in 3D under the discrete Fréchet distance. There are many open questions. (1) Our algorithms also work for any fixed dimension $d > 3$. However, when applied on 2D chains our algorithms might return self-intersecting approximating chains. This is also a problem for previous chain simplification algorithms using (continuous) Fréchet distance. In fact, this was listed as an open problem in [1]. How can we handle this problem? (2) In Theorem 4, the running time of the algorithm is dominated by the computation of the smallest enclosing balls in table $R[-,-,-]$. Is there a way to improve the $O(n^3)$ bound? Also, regardless of the running time it might be possible to reduce the space complexity in Theorem 4 (and Theorem 2). (3) The proof of the NP-completeness of CPS-2H uses general 3D polylines, not really ridges. Can we use ridges to finish the proof? (4) What is the complexity of the CPS-3F problem? We conjecture that it is also NP-complete.

References

1. Agarwal, P., Har-Peled, S., Mustafa, N., Wang, Y.: Near-linear time approximation algorithms for curve simplification. Algorithmica 42, 203–219 (2005)
2. Alt, H., Godau, M.: Measuring the resemblance of polygonal curves. In: Proceedings of the 8th Annual Symposium on Computational Geometry (SoCG 1992), pp. 102–109 (1992)
3. Alt, H., Godau, M.: Computing the Fréchet distance between two polygonal curves. Intl. J. Computational Geometry and Applications 5, 75–91 (1995)
4. Barequet, G., Chen, D.Z., Daescu, O., Goodrich, M., Snoeyink, J.: Efficiently approximating polygonal paths in three and higher dimensions. Algorithmica 33, 150–167 (2002)
5. Chan, T.: Optimal output-sensitive convex hull algorithms in two and three dimensions. Discrete and Computational Geometry 16, 361–368 (1996)

6. Chan, S., Chin, F.: Approximation of polygonal curves with minimum number of line segments or minimum error. Intl. J. Computational Geometry and Applications 6, 59–77 (1996)
7. Chen, D.Z., Daescu, O.: Space-efficient algorithms for approximating polygonal curves in two-dimensional space. Intl. J. Computational Geometry and Applications 13, 95–111 (2003)
8. Chen, Z., Fu, B., Zhu, B.: The approximability of the exemplar breakpoint distance problem. In: Cheng, S.-W., Poon, C.K. (eds.) AAIM 2006. LNCS, vol. 4041, pp. 291–302. Springer, Heidelberg (2006)
9. Eiter, T., Mannila, H.: Computing discrete Fréchet distance. Tech. Report CD-TR 94/64, Information Systems Department, Technical University of Vienna (1994)
10. Eu, D., Toussaint, G.: On approximating polygonal curves in two and three dimensions. CVGIP: Graphical Models and Image Processing 56, 231–246 (1994)
11. Fréchet, M.: Sur quelques points du calcul fonctionnel. Rendiconti del Circolo Mathematico di Palermo 22, 1–74 (1906)
12. Godau, M.: A natural metric for curves — computing the distance for polygonal chains and approximation algorithms. In: Jantzen, M., Choffrut, C. (eds.) STACS 1991. LNCS, vol. 480, pp. 127–136. Springer, Heidelberg (1991)
13. Guibas, L., Hershberger, J., Mitchell, J., Snoeyink, J.: Approximating polygons and subdivisions with minimum-link paths. Intl. J. Computational Geometry and Applications 3, 383–415 (1993)
14. Imai, H., Iri, M.: Computational-geometric methods for polygonal approximation. CVGIP 36, 31–41 (1986)
15. Imai, H., Iri, M.: An optimal algorithm for approximating a piecewise linear function. J. of Information Processing 9, 159–162 (1986)
16. Imai, H., Iri, M.: Polygonal approximation of a curve — formulations and algorithms. In: Toussaint, G. (ed.) Computational Morphology, pp. 71–86 (1988)
17. Jiang, M., Xu, Y., Zhu, B.: Protein structure-structure alignment with discrete Fréchet distance. In: Proceedings of the 5th Asia-Pacific Bioinformatics Conf (APBC'07), pp. 131–141 (2007)
18. Kenyon-Mathieu, C., King, V.: Verifying partial orders. In: Proceedings of the 21st Annual Symposium on Theory of Computing (STOC'89), pp. 367–374 (1989)
19. Megiddo, N.: Linear programming in linear time when the dimension is fixed. J. ACM 31(1), 114–127 (1984)
20. Melkman, A., O'Rourke, I.: On polygonal chain approximation. In: Toussaint, G. (ed.) Computational Morphology, pp. 87–95 (1988)
21. McAllister, M., Snoeyink, J.: Medial axis generalisation of hydrology networks. In: AutoCarto 13: ACSM/ASPRS Ann. Convention Technical Papers, Seattle, WA, pp. 164–173. (1997)
22. Varadarajan, K.: Approximating monotone polygonal curves using the uniform metric. In: Proceedings of the 12th Annual Symposium on Computational Geometry (SoCG 1996), pp. 311–318 (1996)
23. Wenk, C.: Shape Matching in Higher Dimensions. PhD thesis, Freie Universitaet Berlin (2002)
24. Zhu, B.: Protein local structure alignment under the discrete Fréchet distance. J. Computational Biology 14(10), 1343–1351 (2007)