PROTEIN STRUCTURE-STRUCTURE ALIGNMENT WITH DISCRETE FRÉCHET DISTANCE

MINGHUI JIANG[*]

Department of Computer Science, Utah State University,
Logan, UT 84322-4205, USA
Email: mjiang@cc.usu.edu

YING XU[†]

Department of Biochemistry and Molecular Biology, University of Georgia,
Athens, GA 30602-7229, USA
Email: xyn@bmb.uga.edu

BINHAI ZHU

Department of Computer Science, Montana State University,
Bozeman, MT 59717-3880, USA
Email: bhz@cs.montana.edu

Matching two geometric objects in 2D and 3D spaces is a central problem in computer vision, pattern recognition and protein structure prediction. In particular, the problem of aligning two polygonal chains under translation and rotation to minimize their distance has been studied using various distance measures. It is well known that the Hausdorff distance is useful for matching two point sets, and that the Fréchet distance is a superior measure for matching two polygonal chains. The discrete Fréchet distance closely approximates the (continuous) Fréchet distance, and is a natural measure for the geometric similarity of the folded 3D structures of bio-molecules such as proteins. In this paper, we present new algorithms for matching two polygonal chains in 2D to minimize their discrete Fréchet distance under translation and rotation, and an effective heuristic for matching two polygonal chains in 3D. We also describe our empirical results on the application of the discrete Fréchet distance to the protein structure-structure alignment.

1. Introduction

Matching two geometric objects in 2D and 3D spaces is a central problem in computer vision, pattern recognition and protein structure prediction. A lot of research has been done in this aspect using various distance measures. One of the most popular distance measures is the Hausdorff distance $d_{\mathcal{H}}$. For arbitrary bounded sets $A, B \subseteq \mathbb{R}^2$, it is defined

[*] Supported by USU research funds A13501 and A14766.

[†] Supported in part by National Science Foundation (NSF/DBI-0354771, NSF/ITR-IIS-0407204) and by a Distinguished Cancer Scholar grant from the Georgia Cancer Coalition.

as follows:

$$d_{\mathcal{H}}(A,B) = \max\left(\sup_{a\in A}\inf_{b\in B} dist(a,b), \sup_{b\in B}\inf_{a\in A} dist(a,b)\right).$$

where $dist$ is the underlying metric in the plane, for example the Euclidean metric. Given two point sets with m and n points respectively in the plane, their minimum Hausdorff distance under translation can be computed in $O(mn(m+n)\alpha(mn)\log(mn))$ time [12] and, when both translation and rotation are allowed, in $O((m+n)^6\log(mn))$ time [11]. Given two polygonal chains with m and n vertices respectively in the plane, their minimum Hausdorff distance under translation can be computed in $O((mn)^2\log^3(mn))$ time [6] and, when both translation and rotation are allowed, in $O((mn)^4(m+n)\log(m+n))$ time [2].

The Hausdorff distance is a good measure for the similarity of point sets, but it is inadequate for the similarity of polygonal chains; one can easily come up with examples of two polygonal chains with a small Hausdorff distance but drastically different geometric shapes. Alt and Godau [3] proposed to use the Fréchet distance to measure the similarity of two polygonal chains. The Fréchet distance $\delta_{\mathcal{F}}$ between two parametric curves $f:[0,1]\to\mathbb{R}^2$ and $g:[0,1]\to\mathbb{R}^2$ is defined as follows:

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

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. Given two polygonal chains with m and n vertices respectively in the plane, their Fréchet distance at fixed positions can be computed in $O(mn\log(m+n))$ time [4]; their minimum Fréchet distance under translation can be computed in $O((mn)^3(m+n)^2\log(m+n))$ time [5] and, when both translation and rotation are allowed, in $O((m+n)^{11}\log(m+n))$ time [17].

The Fréchet distance is a superior measure for the similarity of polygonal curves, but it is very difficult to handle. Eiter and Mannila [9] introduced the discrete Fréchet distance as a close approximation of the (continuous) Fréchet distance. We now review their definition of the discrete Fréchet distance using our notations (but with exactly the same idea).

Definition 1.1. Given a polygonal chain $P = \langle p_1,\ldots,p_n\rangle$ of n vertices, a k**-walk** along P partitions the vertices of P into k disjoint non-empty subsets $\{P_i\}_{i=1..k}$ such that $P_i = \langle p_{n_{i-1}+1},\ldots,p_{n_i}\rangle$ and $0 = n_0 < n_1 < \cdots < n_k = n$.

Given two polygonal chains $A = \langle a_1,\ldots,a_m\rangle$ and $B = \langle b_1,\ldots,b_n\rangle$, a **paired walk** along A and B is a k-walk $\{A_i\}_{i=1..k}$ along A and a k-walk $\{B_i\}_{i=1..k}$ along B for some k, such that, for $1\le i\le k$, either $|A_i|=1$ or $|B_i|=1$ (that is, either A_i or B_i contains exactly one vertex). The **cost** of a paired walk $W = \{(A_i,B_i)\}$ along two chains A and B is

$$d_{\mathcal{F}}^{W}(A,B) = \max_i \max_{(a,b)\in A_i\times B_i} dist(a,b).$$

The **discrete Fréchet distance** between two polygonal chains A and B is

$$d_{\mathcal{F}}(A, B) = \min_{W} d_{\mathcal{F}}^{W}(A, B).$$

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

Let's consider again 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) $|B_i| > |A_i| = 1$: the person stays and the dog moves forward;
(2) $|A_i| > |B_i| = 1$: the person moves forward and the dog stays;
(3) $|A_i| = |B_i| = 1$: both the person and the dog move forward.

The following figure shows the relationship between discrete and continuous Fréchet distances. In Figure 1 (I), we have two polygonal chains $\langle a, b\rangle$ and $\langle c, d, e\rangle$; their continuous Fréchet distance is the distance from d to the segment $\overline{ab}$, that is, $dist(d, o)$. The discrete Fréchet distance is $dist(d, b)$. As we can see from the figure, the discrete Fréchet distance could be arbitrarily larger than the continuous distance. On the other hand, if we put enough sample points on the two polygonal chains, then the resulting discrete Fréchet distance, that is, $dist(d, f)$ in Figure 1 (II), closely approximates $dist(d, o)$.

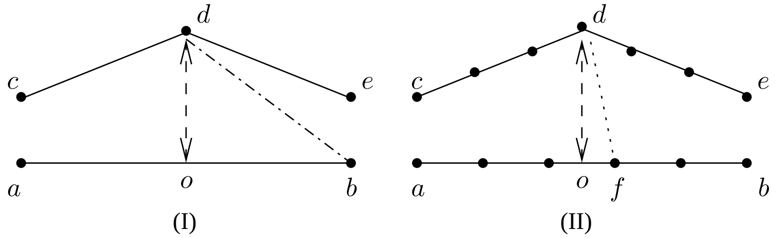


Figure 1. The relationship between discrete and continuous Fréchet distances.

Given two polygonal chains of m and n vertices respectively, their discrete Fréchet distance can be computed in $O(mn)$ time by a dynamic programming algorithm [9]. We now describe our algorithm based on the same idea.

Given two polygonal chains $A = \langle a_1, \ldots, a_m\rangle$ and $B = \langle b_1, \ldots, b_n\rangle$, and their two subchains $A[1..i] = \langle a_1, \ldots, a_i\rangle$ and $B[1..j] = \langle b_1, \ldots, b_j\rangle$, let $d_{<}(i, j)$ (respectively, $d_{>}(i, j)$) denote the discrete Fréchet distance between $A[1..i]$ and $B[1..j]$ such that a_i (respectively, b_j) belongs to a single-vertex subset in the paired walk, and define $d(i, j) = \min\{d_{<}(i, j), d_{>}(i, j)\}$. The discrete Fréchet distance $d_{\mathcal{F}}(A, B) = \min\{d_{<}(m, n), d_{>}(m, n)\}$ can be computed in $O(mn)$ time with the base conditions

$$d_{<}(i, 0) = d_{<}(0, j) = 0 \quad \text{and} \quad d_{>}(i, 0) = d_{>}(0, j) = 0 \quad \text{and} \quad d(i, 0) = d(0, j) = 0,$$

and the recurrences

$$
\begin{aligned}
d_<(i,j) &= \max\begin{cases} dist(a_i, b_j) \\ \min\{d(i-1,j-1), d(i,j-1)\} \end{cases} \\
d_>(i,j) &= \max\begin{cases} dist(a_i, b_j) \\ \min\{d(i-1,j-1), d(i-1,j)\} \end{cases} \\
d(i,j) &= \min\{d_<(i,j), d_>(i,j)\}.
\end{aligned}
$$

In this paper, we present new algorithms that compute the minimum discrete Fréchet distance of two polygonal chains in the plane under translation in $O((mn)^3 \log(m+n))$ time and, when both rotation and translation are allowed, in $O((mn)^4 \log(m+n))$ time. These bounds are two or three orders of magnitude smaller than the corresponding best bounds [5,17] using the continuous Fréchet distance measure.

Our interest in matching two polygonal chains in 2D and 3D spaces is motivated by the application of protein structure-structure alignment. The discrete Fréchet distance is a very natural measure in this application because a protein can be viewed essentially as a chain of discrete amino acids in 3D. We design a heuristic method for aligning two polygonal chains in 3D based on the intuition behind our theoretical results for the 2D case, and use it to measure the geometric similarity of protein tertiary structures with real protein data drawn from the Protein Data Bank (PDB) hosted at `http://www.rcsb.org/pdb/`.

The paper is organized as follows. In Section 2, we present our algorithms for matching two polygonal chains in 2D under translation and rotation. In Section 3, we describe our heuristic method for matching two polygonal chains in 3D under translation and rotation, and present our empirical results on protein structure-structure alignment with the discrete Fréchet distance. In Section 4, we conclude the paper.

2. Matching 2D Polygonal Chains Under Translation and Rotation

Definition 2.1. *(Optimization Problem)* Given two polygonal chains A and B, a transformation class T, and a distance measure d, find a transformation $\tau \in T$ such that $d(A, \tau(B))$ is minimized.

Definition 2.2. *(Decision Problem)* Given two polygonal chains A and B, a transformation class T, a distance measure d, and a real number $\epsilon \geq 0$, decide whether there is a transformation $\tau \in T$ such that $d(A, \tau(B)) \leq \epsilon$.

Observation 2.1. Given two polygonal chains A and B, if there is a transformation τ such that $d_{\mathcal{F}}(A, \tau(B)) = \epsilon$, then there are a vertex $a \in A$ and a vertex $b \in B$ such that $dist(a, \tau(b)) = \epsilon$.

2.1. *Matching Under Translation*

We first consider the transformation class T_t of all translations.

Lemma 2.1. *Given two 2D polygonal chains A and B, if there is a translation $\tau \in T_t$ such that $d_{\mathcal{F}}(A, \tau(B)) = \epsilon > 0$, then one of the following four cases is true:*

(1) *there are a vertex $a \in A$ and a vertex $b \in B$ such that, for any translation $\tau' \in T_t$, $dist(a, \tau'(b)) = \epsilon \implies d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$;*
(2) *there are two vertices $a, c \in A$, a vertex $b \in B$, and a translation $\tau' \in T_t$ such that $dist(a, \tau'(b)) = dist(c, \tau'(b)) = \epsilon$ and $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$;*
(3) *there are a vertex $a \in A$, two vertices $b, d \in B$, and a translation $\tau' \in T_t$ such that $dist(a, \tau'(b)) = dist(a, \tau'(d)) = \epsilon$ and $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$;*
(4) *there are two vertices $a, c \in A$, two vertices $b, d \in B$, and a translation $\tau' \in T_t$ such that $\overrightarrow{ac} \ne \overrightarrow{bd}$ (that is, either $|ac| \ne |bd|$, or $\overrightarrow{ac}$ and $\overrightarrow{bd}$ have different directions), $dist(a, \tau'(b)) = dist(c, \tau'(d)) = \epsilon$, and $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$.*

Proof. Let $a \in A$ and $b \in B$ be the two vertices such that $dist(a, \tau(b)) = \epsilon$, the existence of which is guaranteed by Observation 2.1. Let $W = \{(A_i, B_i)\}$ be the Fréchet alignment of A and $\tau(B)$ such that $d^W_{\mathcal{F}}(A, \tau(B)) = \epsilon$. We translate B with τ' (starting at τ) such that the distance between the two vertices a and b remains at exactly ϵ, that is, $dist(a, \tau'(b)) = \epsilon$. We consider the distance $d^W_{\mathcal{F}}(A, \tau'(B)) = \max_i \max_{(p,q) \in A_i \times B_i} dist(p, \tau'(q))$ as τ' changes continuously.

As τ' changes continuously, $\tau'(b)$ rotates around a in a circle of radius ϵ. If $d^W_{\mathcal{F}}(A, \tau'(B))$ always remains at ϵ, we have case 1; otherwise, there are two vertices $c \in A_i$ and $d \in B_i$ for some i such that the distance $dist(c, \tau'(d))$ crosses the threshold ϵ. We cannot have both $a = c$ and $b = d$ because the distance $dist(a, \tau'(b))$ always remains at ϵ; for the same reason, we cannot have $\overrightarrow{ac} = \overrightarrow{bd}$. There are three possible cases: if $a \ne c$ and $b = d$, we have case 2; if $a = c$ and $b \ne d$, we have case 3; if $a \ne c$ and $b \ne d$, we have case 4. □

The previous lemma implies the following algorithm that checks the four cases:

(1) For every two vertices $a \in A$ and $b \in B$, compute an arbitrary translation τ' such that $dist(a, \tau'(b)) = \epsilon$, and check whether $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$.
(2) For every three vertices $a, c \in A$ and $b \in B$, compute all possible translations τ' such that $dist(a, \tau'(b)) = dist(c, \tau'(b)) = \epsilon$, and check whether $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$.
(3) For every three vertices $a \in A$ and $b, d \in B$, compute all possible translations τ' such that $dist(a, \tau'(b)) = dist(a, \tau'(d)) = \epsilon$, and check whether $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$.
(4) For every four vertices $a, c \in A$ and $b, d \in B$ such that $\overrightarrow{ac} \ne \overrightarrow{bd}$, compute all possible translations τ' such that $dist(a, \tau'(b)) = dist(c, \tau'(d)) = \epsilon$, and check whether $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$.

The algorithm answers yes if it finds at least one translation τ' such that $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$; otherwise, it answers no. As we can see from the following lemma, this algorithm solves the decision problem.

Lemma 2.2. *If there is translation τ' such that $d_{\mathcal{F}}(A, \tau'(B)) = \epsilon'$, then, for any distance $\epsilon \ge \epsilon'$, there exists a translation τ such that $d_{\mathcal{F}}(A, \tau(B)) = \epsilon$.*

Proof. As we translate B from $\tau'(B)$ to infinity, the discrete Fréchet distance between A and the translated B changes continuously (since it is a composite function based on the continuous Euclidean distance functions) from $d_{\mathcal{F}}(A, \tau'(B)) = \epsilon'$ to infinity. The continuity implies that, for any $\epsilon \geq \epsilon'$, there exists a translation τ such that $d_{\mathcal{F}}(A, \tau(B)) = \epsilon$. □

We now analyze the algorithm. In cases 2 and 3, given two points p and q such that $p \neq q$, the two equations $dist(x, p) = \epsilon$ and $dist(x, q) = \epsilon$ together determine x (there are at most two solutions for x) since the 2D point x has two variable components. In case 4, given two points p and q, and a vector $\vec{v} \neq \overrightarrow{pq}$, the two equations $dist(x, p) = \epsilon$ and $dist(x + \vec{v}, q) = \epsilon$ are independent and determine x (there are at most a constant number of solutions for x). Given a translation τ', to check whether $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$ takes $O(mn)$ time. The overall time complexity is $O(mn \cdot m^2 n^2) = O(m^3 n^3)$.

With binary search, our algorithm for the decision problem implies an $O(m^3 n^3 \log(1/\epsilon))$ time $1 + \epsilon$ approximation for the optimization problem; with parametric search (applying Cole's sorting trick [5,8]), it implies an $O(m^3 n^3 \log(m + n))$ time exact algorithm. We have the following theorem.

Theorem 2.1. *For minimizing the discrete Fréchet distance between two 2D polygonal chains under translation, we have an $O(m^3 n^3 \log(1/\epsilon))$ time $1 + \epsilon$ approximation algorithm and an $O(m^3 n^3 \log(m + n))$ time exact algorithm.*

2.2. *Matching Under Translation and Rotation*

We next consider the transformation class T_{tr} that includes both translations and rotations.

Lemma 2.3. *Given two 2D polygonal chains A and B, if there is a transformation $\tau \in T_{tr}$ such that $d_{\mathcal{F}}(A, \tau(B)) = \epsilon > 0$, then one of the following seven cases is true:*

1. *there are a vertex $a \in A$ and a vertex $b \in B$ such that, for any transformation $\tau' \in T_{tr}$, $dist(a, \tau'(b)) = \epsilon \implies d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$;*
2. *there are two vertices $a, c \in A$ and two vertices $b, d \in B$ such that, for any transformation $\tau' \in T_{tr}$, $dist(a, \tau'(b)) = dist(c, \tau'(d)) = \epsilon \implies d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$;*
3. *there are two vertices $a, c \in A$, three vertices $b, d, f \in B$, and a transformation $\tau' \in T_{tr}$ such that $dist(a, \tau'(b)) = dist(c, \tau'(d)) = dist(c, \tau'(f)) = \epsilon$ and $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$.*
4. *there are three vertices $a, c, e \in A$, two vertices $b, d \in B$, and a transformation $\tau' \in T_{tr}$ such that $dist(a, \tau'(b)) = dist(c, \tau'(d)) = dist(e, \tau'(d)) = \epsilon$ and $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$.*
5. *there are three vertices $a, c, e \in A$ and three vertices $b, d, f \in B$ ($\triangle ace$ and $\triangle bdf$ are not congruent), and a transformation $\tau' \in T_{tr}$ such that $dist(a, \tau'(b)) = dist(c, \tau'(d)) = dist(e, \tau'(f)) = \epsilon$, and $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$.*
6. *there are three vertices $a, c, e \in A$ and three vertices $b, d, f \in B$ ($\triangle ace$ and $\triangle bdf$ are congruent), and a transformation $\tau' \in T_{tr}$ such that the two triangles*

$\triangle ace$ and $\tau'(\triangle bdf)$ are not parallel (their corresponding edges are not parallel), $dist(a, \tau'(b)) = dist(c, \tau'(d)) = dist(e, \tau'(f)) = \epsilon$, and $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$.

(7) there are three vertices $a, c, e \in A$ and three vertices $b, d, f \in B$ ($\triangle ace$ and $\triangle bdf$ are congruent) such that, for any transformation $\tau' \in T_{tr}$, if $\triangle ace$ and $\tau'(\triangle bdf)$ are parallel, and if $dist(a, \tau'(b)) = dist(c, \tau'(d)) = dist(e, \tau'(f)) = \epsilon$, then $d_{\mathcal{F}}(A, \tau'(B)) \le \epsilon$.

Proof. Let $a \in A$ and $b \in B$ be the two vertices such that $dist(a, \tau(b)) = \epsilon$, the existence of which is guaranteed by Observation 2.1. Let $W = \{(A_i, B_i)\}$ be the Fréchet alignment of A and $\tau(B)$ such that $d^W_{\mathcal{F}}(A, \tau(B)) = \epsilon$. Without loss of generality, we assume that $a \in A_i$, $b \in B_i$, and b is the only vertex in B_i.

Starting with $\tau' = \tau$, we rotate B around the vertex b. During the rotation, the distance between the two vertices a and b remains at exactly ϵ, that is, $dist(a, \tau'(b)) = \epsilon$. If $d^W_{\mathcal{F}}(A, \tau'(B))$ always remains at ϵ, we have case 1.

Otherwise, there are two vertices $c \in A_j$ and $d \in B_j$ for some j such that the distance $dist(c, \tau'(d))$ crosses the threshold ϵ. We must have $i \neq j$ because b is the only vertex in B_i and the positions of the vertices in A_i are fixed as we rotate B around b. It follows that $a \neq c$ and $b \neq d$. Now, we continue to transform B while keeping the two constraints $dist(a, \tau'(b)) = \epsilon$ and $dist(c, \tau'(d)) = \epsilon$ satisfied. If $d^W_{\mathcal{F}}(A, \tau'(B))$ always remains at ϵ, we have case 2.

Otherwise, there are two vertices $e \in A_k$ and $f \in B_k$ for some k such that the distance $dist(e, \tau'(f))$ crosses the threshold ϵ. We must have $k \neq i$ for the same reason that $j \neq i$. We consider two possibilities: either $k = j$ or $k \neq j$.

If $k = j$, then we must have either $e = c$ or $f = d$ because either A_j or B_j contains a single vertex. We cannot have both $e = c$ and $f = d$ because we keep the constraint $dist(c, \tau'(d)) = \epsilon$ satisfied during the transformation. If $e = c$, we have case 3; if $f = d$, we have case 4.

If $k \neq j$, then we consider the two triangles $\triangle ace$ and $\tau'(\triangle bdf)$.

(1) If they are not congruent, we have case 5.
(2) If they are congruent by not parallel, we have case 6.
(3) If they are both congruent and parallel, then we translate B continuously while keeping the three constraints $dist(a, \tau'(b)) = dist(c, \tau'(d)) = dist(e, \tau'(f)) = \epsilon$ satisfied. During the translation, we either encounter another pair of vertices e' and f' whose distance crosses the threshold ϵ or not. If we encounter e' and f', then the two triangles $\triangle ace'$ and $\triangle bdf'$ must not be congruent, and we have case 5; otherwise, we have case 7. □

As before, the previous lemma implies an algorithm for the decision problem. We now analyze the running time. In cases 1, 2, and 7, we only need to find one transformation τ'. In cases 3 and 4, there are at most four transformations for τ'. In case 5, the transformation for τ' can be specified by six variables: the x and y coordinates of the three vertices b, d, and f; we also have six constraints for the lengths of the six segments ab, cd, ef, bd, df, and bf. Each constraint is specified by a quadratic equation. There are at most a constant number of solutions for these equations.

In case 6, we have two congruent triangles $\triangle ace$ and $\triangle b'd'f'$ ($\triangle b'd'f' = \tau'(\triangle bdf)$). If the two triangles have the same enclosing circle, then there are at most two transformations such that $|ab'| = |cd'| = |ef'| = \epsilon$. If the two triangles do not have the same enclosing circle, then we can always translate $\triangle ace$ to $\triangle a'c'e'$ such that $\triangle a'c'e'$ and $\triangle b'd'f'$ have the same enclosing circle, then rotate $\triangle a'c'e'$ to $\triangle b'd'f'$. We have $|ab'| = |cd'| = |ef'| = \epsilon > 0$, $|a'b'| = |c'd'| = |e'f'| = x > 0$ (since they are not parallel), and

$$\overrightarrow{ab'} = \overrightarrow{a'b'} + \vec{v}, \qquad \overrightarrow{cd'} = \overrightarrow{c'd'} + \vec{v}, \qquad \overrightarrow{ef'} = \overrightarrow{e'f'} + \vec{v}.$$

Given a fixed vector $\vec{v}$, the equation $\vec{w} = \vec{u} + \vec{v}$, subject to the two constraints $|\vec{w}| = \epsilon > 0$ and $|\vec{u}| = x > 0$, has at most two solutions for $\vec{w}$ and $\vec{u}$. On the other hand, the three vectors $\overrightarrow{a'b'}$, $\overrightarrow{c'd'}$, and $\overrightarrow{e'f'}$ are distinct, which is a contradiction. Therefore, the two triangles $\triangle ace$ and $\triangle b'd'f'$ must have the same enclosing circle.

Theorem 2.2. *For minimizing the discrete Fréchet distance between two 2D polygonal chains under translation and rotation, we have an* $O(m^4 n^4 \log(1/\epsilon))$ *time* $1 + \epsilon$ *approximation algorithm and an* $O(m^4 n^4 \log(m + n))$ *time exact algorithm.*

3. Protein Structure-Structure Alignment

The discrete Fréchet distance between two polygonal chains is a natural measure for comparing the geometric similarity of protein tertiary structures because the alpha-carbon atoms along the backbone of a protein essentially forms a 3D polygonal chain.

Generalizing the theoretical results in the previous section, it is possible to match two polygonal chains with m and n vertices in 3D in roughly $O((mn)^5)$ time under both translation and rotation. However, this would be too slow for our target application of protein structure-structure alignment, where a typical protein corresponds to a 3D polygonal chain with 300–500 amino acids. Instead of an exact algorithm, we propose an intuitive heuristic and present our empirical results showing its effectiveness in matching two similar polygonal chains.

3.1. *A Heuristic for Matching 3D Polygonal Chains Under Translation and Rotation*

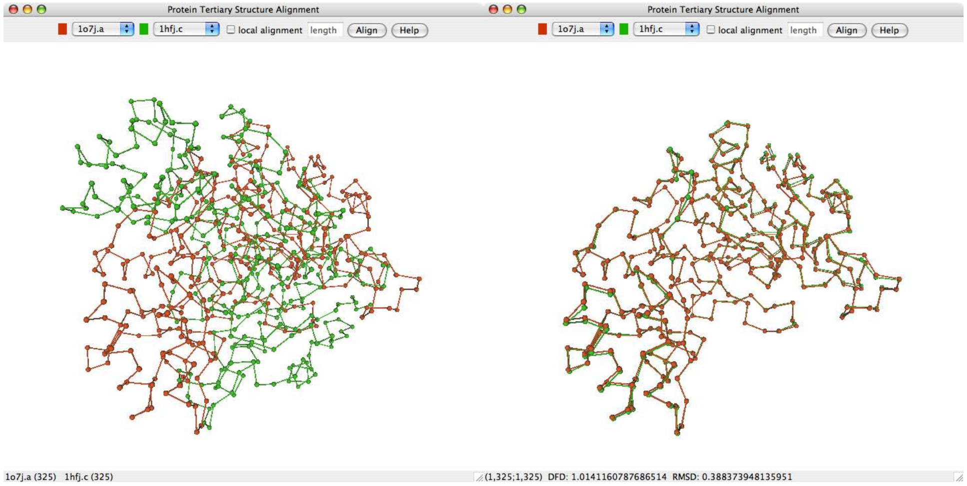
Given a 3D chain C of n vertices, the coordinates of each vertex c_i of C can be represented by a 3D vector $\vec{c_i}$. The *center* c of the chain C corresponds to the vector $\vec{c} = \frac{\sum_i \vec{c_i}}{n}$. We observe that, given two polygonal chains $A = \langle a_1, \ldots, a_m \rangle$ and $B = \langle b_1, \ldots, b_n \rangle$, if $d_{\mathcal{F}}(A, B) = \epsilon$, then we must have both $dist(a_1, b_1) \le \epsilon$ and $dist(a_m, b_n) \le \epsilon$. If ϵ is smaller than half the minimum distance between two consecutive vertices in either A or B, then the Fréchet alignment of A and B must contain only one-to-one matches between vertices of A and B. That is, we must have $m = n$ and, for $1 \le i \le n$, $dist(a_i, b_i) \le \epsilon$. It follows that $dist(a, b) \le \epsilon$, where a and b are the centers of A and B, respectively.

The observation above suggests that we can use the three points, the two end-vertices and the center, as the reference points [1] for each chain. For two polygonal chains with a small discrete Fréchet distance, their corresponding reference points must be close. In general, the position and orientation of each polygonal chain is determined by the positions of its three reference points. We have the following heuristic for matching A and B under translation and rotation:

(1) Translate B such that the center a of A and the center b of B coincide.
(2) Rotate B around b such that the two triangles $\triangle aa_1a_m$ and $\triangle bb_1b_n$ are co-planar and such that the two vectors $\frac{\vec{a_1}+\vec{a_m}}{2} - \vec{a}$ and $\frac{\vec{b_1}+\vec{b_n}}{2} - \vec{b}$ have the same direction.
(3) Rotate B for a small angle around the axis through its two randomly chosen vertices. If this does not decrease the discrete Fréchet distance between A and B, rotate back.
(4) Repeat the previous tuning step for a number of times.

3.2. *The Experiment*

We implemented our protein structure-structure alignment heuristic and a protein visualization software [a] in Java. The experiment was conducted on an Apple iMac with a 2GHz PowerPC G5 processor and 2GB DDR SDRAM memory running Mac OS 10.4.3 and Java 1.4.2.



1o7j.a (325) 1hfj.c (325)

(1,325;1,325) DFD: 1.0141160787686514 RMSD: 0.388373948135951

Figure 2. The alignment of 1o7j.a and 1hfj.c by our heuristic.

In the experiment, we align the protein chain 1o7j.a (PDB ID 1o7j; chain A) with seven other protein chains 1hfj.c, 1qd1.b, 1toh, 4eca.c, 1d9q.d, 4eca.b, and 4eca.d. Each of these

[a] The program is hosted on the web at `http://www.cs.usu.edu/∼mjiang/frechet.html`.

eight chains contains exactly 325 vertices, where each vertex represents an alpha-carbon atom on the protein backbone. When the number of tuning steps is set to 20, our program takes less than one second to align two chains of lengths 325 on our test machine. Figure 2 shows two screenshots of our program, before and after aligning the two protein chains 1o7j.a and 1hfj.c.

We compare our heuristic with ProteinDBS [16], an online protein database search engine hosted at `http://proteindbs.rnet.missouri.edu/` that supports protein structure-structure alignment. ProteinDBS uses computer vision techniques to align two protein chains based on the two-dimensional distance matrix generated from the 3D coordinates of the alpha-carbon atoms on the protein backbones. The two chains 1o7j.a and 1hfj.c are examples given in the ProteinDBS paper [16]. According to the query result from the ProteinDBS website, the seven chains (1hfj.c, 1qd1.b, 1toh, 4eca.c, 1d9q.d, 4eca.b, 4eca.d) have global tertiary structures most similar to 1o7j.a.

Table 1. The characteristics of the seven chains with the highest similarity ranking by ProteinDBS.

Protein Chain	Alignment Length	RMSD (in angstrom)	Discrete Fréchet Distance (in angstrom)
1hfj.c	325	0.27	1.01
1qd1.b	85	2.81	22.90
1toh	55	2.91	35.09
4eca.c	317	1.10	6.01
1d9q.d	81	2.88	22.18
4eca.b	317	1.09	5.76
4eca.d	318	1.45	5.92

By comparing the image patterns in the distance matrices instead of aligning the tertiary structures geometrically, ProteinDBS is very efficient but not so accurate. We refer to Table 1, which lists the characteristics of the alignments generated by ProteinDBS. The three protein chains, 1qd1.b, 1toh, and 1d9q.d, have global tertiary structures dissimilar to that of the chain 1o7j.a, but they are incorrectly ranked among the top by ProteinDBS. The discrete Fréchet distances of these chains and the query chain computed by our heuristic correctly identify the three dissimilar protein chains.

4. Conclusion

In this paper, we present the first algorithms for matching two polygonal chains in 2D to minimize their discrete Fréchet distance under translation and rotation. Our algorithms are two or three orders of magnitude faster than the fastest algorithms using the continuous Fréchet distance, and can be readily generalized to higher dimensions.

The discrete Fréchet distance is a natural measure for comparing the folded 3D structures of bio-molecules such as proteins. Our experiment shows that our heuristic for aligning protein tertiary structures using the discrete Fréchet distance is more accurate than ProteinDBS's structure aligning algorithm, which is based on computer vision techniques. We are currently conducting more empirical studies and refining our protein structure-structure alignment algorithm with additional ideas from some other popular algorithms such as the

Combinatorial Extension (CE) Method [15] hosted at `http://cl.sdsc.edu/`. We see great potential for using the discrete Fréchet distance in the local alignment [10], the feature identification, and the consensus shape construction [7] of multiple proteins.

References

1. O. Aichholzer, H. Alt, and G. Rote. Matching shapes with a reference point. *International Journal of Computational Geometry & Applications*, 7(4):349–363, 1997.
2. H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes (extended abstract). In *Proceedings of the 7th Annual Symposium on Computational Geometry (SoCG'91)*, pages 186–193, 1991.
3. H. Alt and M. Godau. Measuring the resemblance of polygonal curves. In *Proceedings of the 8th Annual Symposium on Computational Geometry (SoCG'92)*, pages 102–109, 1992.
4. H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995.
5. H. Alt, C. Knauer, and C. Wenk. Matching polygonal curves with respect to the Fréchet distance. In *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS'01)*, pages 63–74, 2001.
6. P.K. Agarwal, M. Sharir, and S. Toledo. Applications of parametric search in geometric optimization. In *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'92)*, pages 72–82, 1992.
7. L.P. Chew and K. Kedem. Finding the consensus shape of a protein family. In *Proceedings of the 18th Annual Symposium on Computational Geometry (SoCG'02)*, pages 64–73, 2002.
8. R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *Journal of the ACM*, 34:200–208, 1987.
9. T. Eiter and H. Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Information Systems Department, Technical University of Vienna, 1994.
10. D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
11. D.P. Huttenlocher, K. Kedem, and J.M. Kleinberg. On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under Euclidean motion in the plane. In *Proceedings of the 8th Annual Symposium on Computational Geometry (SoCG'92)*, pages 110–119, 1992.
12. D.P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of Voronoi surfaces and its applications. In *Proceedings of the 7th Annual Symposium on Computational Geometry (SoCG'91)*, pages 194–203, 1991.
13. P. Indyk. Approximate nearest neighbor algorithms for Fréchet distance via product metrics. In *Proceedings of the 18th Annual Symposium on Computational Geometry (SoCG'02)*, pages 102–106, 2002.
14. A. Mosig and M. Clausen. Approximately matching polygonal curves with respect to the Fréchet distance. *Computational Geometry: Theory and Applications*, 30(2):113–127, 2005.
15. I.N. Shindyalov and P.E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, 11(9):739–747, 1998.
16. C.-R. Shyu, P.-H. Chi, G. Scott, and D. Xu. ProteinDBS: a real-time retrieval system for protein structure comparison. *Nucleic Acids Research*, 32:W572–575, 2004.
17. C. Wenk. *Shape Matching in Higher Dimensions*. PhD thesis, Freie Universitaet Berlin, 2002.