Journal of Bioinformatics and Computational Biology
Vol. 6, No. 1 (2008) 51–64

Imperial College Press
www.icpress.co.uk

PROTEIN STRUCTURE–STRUCTURE ALIGNMENT WITH DISCRETE FRÉCHET DISTANCE

MINGHUI JIANG

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

YING XU

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

BINHAI ZHU

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

Received 15 April 2007
Accepted 25 October 2007

Matching two geometric objects in two-dimensional (2D) and three-dimensional (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 biomolecules such as proteins. In this paper, we present new algorithms for matching two polygonal chains in two dimensions to minimize their discrete Fréchet distance under translation and rotation, and an effective heuristic for matching two polygonal chains in three dimensions. We also describe our empirical results on the application of the discrete Fréchet distance to protein structure–structure alignment.

Keywords: Protein structure–structure alignment; discrete Fréchet distance; geometric pattern matching.

1. Introduction

Matching two geometric objects in two-dimensional (2D) and three-dimensional (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 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, e.g. 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[1] or, when both translation and rotation are allowed, in $O((m+n)^6\log(mn))$ time.[2] 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[3] or, when both translation and rotation are allowed, in $O((mn)^4(m+n)\log(m+n))$ time.[4]

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[5] 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 nondecreasing 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[6]; and their minimum Fréchet distance under translation can be computed in $O((mn)^3(m+n)^2\log(m+n))$ time[7] or, when both translation and rotation are allowed, in $O((m+n)^{11}\log(m+n))$ time.[8]

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[a]).

Definition 1. Given a polygonal chain $P = \langle p_1, p_2, \ldots, p_n\rangle$ of n vertices, a **k-walk** along P partitions the vertices of P into k disjoint nonempty subsets $\{P_i\}_{i=1,2,\ldots,k}$ such that $P_i = \langle p_{n_{i-1}+1}, \ldots, p_{n_i}\rangle$ and $0 = n_0 < n_1 < \cdots < n_k = n$.

[a]Unaware of the previous work by Eiter and Mannila,[9] the authors of this paper had come up with this idea independently. Indeed, the discrete Fréchet distance is such a natural concept that it has been rediscovered many times. The recent work by Mosig and Clausen[10] is another example.

Given two polygonal chains $A = \langle a_1, a_2, \ldots, a_m \rangle$ and $B = \langle b_1, b_2, \ldots, b_n \rangle$, a **paired walk** along A and B is a k-walk $\{A_i\}_{i=1,2,\ldots,k}$ along A and a k-walk $\{B_i\}_{i=1,2,\ldots,k}$ along B for some k such that, for $1 \leq i \leq k$, 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 us 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.

Figure 1 shows the relationship between discrete and continuous Fréchet distances. In Fig. 1(a), 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 Fig. 1(b) — closely approximates $dist(d, o)$.

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.[9,10] We now describe our algorithm based on the same idea.

Given two polygonal chains $A = \langle a_1, a_2, \ldots, a_m \rangle$ and $B = \langle b_1, b_2, \ldots, b_n \rangle$, and their two subchains $A[1, 2, \ldots, i] = \langle a_1, a_2, \ldots, a_i \rangle$ and $B[1, 2, \ldots, j] = \langle b_1, b_2, \ldots, b_j \rangle$, let $d_<(i, j)$ — respectively, $d_>(i, j)$ — denote the discrete Fréchet distance between $A[1, 2, \ldots, i]$ and $B[1, 2, \ldots, j]$ such that a_i — respectively, b_j — belongs to

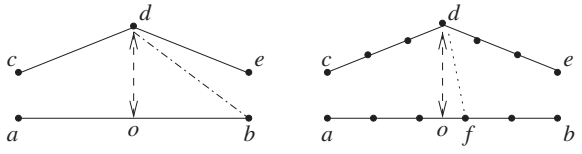


Fig. 1. The relationship between discrete and continuous Fréchet distances.

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

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

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

$$d(i,j) = \min\{d_<(i,j), d_>(i,j)\}.$$

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[7,8] using the continuous Fréchet distance measure. Our technique is not original: similar to the previous algorithms[1–4,7,8,11,12] on the Hausdorff and Fréchet distance measures, our algorithms essentially enumerate all possible critical transformations determined by some carefully chosen reference pairs of geometric objects. A major characteristic of this approach is that the time complexity of the algorithm crucially depends on the combinatorial complexity of the critical transformations, which in turn depends on the intrinsic complexity of the underlying distance measure. Since the discrete Fréchet distance measure does not consider the distances involving points in the interior of the edges of the polygonal chains, the number of critical transformations is drastically reduced, which, not surprisingly, leads to the reduced time complexities of our algorithms.

Admittedly, our algorithms only solve a special case of the more difficult problem for the continuous Fréchet distance measure using essentially the same standard technique; the more general solution for the continuous Fréchet distance[7,8] may be simplified to a solution for the discrete Fréchet distance. However, we recognize (as Eiter and Mannila[9] also did) that the discrete Fréchet distance is a very important special case of the (continuous) Fréchet distance. We believe, especially in light of the biological applications of the discrete Fréchet distance, that it deserves special treatment.

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 three dimensions. We design a heuristic method for aligning two polygonal chains in three dimensions 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 Sec. 2, we present our algorithms for matching two polygonal chains in two dimensions under translation and rotation. In Sec. 3, we describe our heuristic method for matching two polygonal chains in three dimensions under translation and rotation, and present our empirical results on protein structure–structure alignment with the discrete Fréchet distance. In Sec. 4, we conclude the paper.

2. Matching 2D Polygonal Chains Under Translation and Rotation

Definition 2. (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 3. (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 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 two vertices $a \in A$ and $b \in B$ such that $dist(a, \tau(b)) = \epsilon$.

2.1. *Matching under translation*

We first consider the transformation class T_t of all translations.

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

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 1. Let $W = \{(A_i, B_i)\}$ be the Fréchet alignment of A and $\tau(B)$ such that $d_{\mathcal{F}}^{W}(A, \tau(B)) = \epsilon$. 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 $d_{\mathcal{F}}^{W}(A, \tau'(B)) = \max_i \max_{(p,q) \in A_i \times B_i} dist(p, \tau'(q))$ as τ' changes continuously.

As τ' changes continuously, $\tau'(b)$ rotates around a in a circle of radius ϵ. If $d_{\mathcal{F}}^{W}(A, \tau'(B))$ 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 $a \neq c$ and $b = d$, we have case 2; if $a = c$ and $b \neq d$, we have case 3; and if $a \neq c$ and $b \neq d$, 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 $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$.
(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 $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$.
(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 $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$.
(4) For every four vertices $a, c \in A$ and $b, d \in B$ such that $\overrightarrow{ac} \neq \overrightarrow{bd}$, compute all possible translations τ' such that $dist(a, \tau'(b)) = dist(c, \tau'(d)) = \epsilon$, and check whether $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon$.

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

Lemma 2. *If there is a translation τ' such that $d_{\mathcal{F}}(A, \tau'(B)) = \epsilon'$, then, for any distance $\epsilon \geq \epsilon'$, 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.

Alternatively, we can use parametric search with Cole's sorting trick[7,13] to obtain an $O(m^3n^3\log(m+n))$ time exact algorithm. This can be roughly done as follows. The optimization problem can be solved by at most $O(m^2n^2)$ independent binary searches, each on a set of at most mn items. The comparison in the binary searches is exactly the decision procedure which we have just described, which takes $O(m^3n^3)$ time. When a group of comparisons is performed, Cole's "batching rule" obviously holds in this case. It then follows from Lemma 4 in Cole's paper[13] that the parametric search procedure can be done in $O(\log m^2n^2 + \log mn) = O(\log(m+n))$ steps. The overall running time for solving the optimization problem is therefore $O(m^3n^3\log(m+n))$. We have the following theorem.

Theorem 1. *For minimizing the discrete Fréchet distance between two 2D polygonal chains under translation, we have an $O(m^3n^3\log(1/\epsilon))$ time $1+\epsilon$ approximation algorithm and an $O(m^3n^3\log(m+n))$ 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 3. *Given two 2D polygonal chains A and B, if there is a transformation $\tau \in T_{\mathrm{tr}}$ such that $d_{\mathcal{F}}(A, \tau(B)) = \epsilon > 0$, then one of the following seven cases is true*:

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

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 1. Let $W = \{(A_i, B_i)\}$ be the Fréchet alignment of A and $\tau(B)$ such that $d_{\mathcal{F}}^{W}(A, \tau(B)) = \epsilon$. 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 $d_{\mathcal{F}}^{W}(A, \tau'(B))$ 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 $d_{\mathcal{F}}^{W}(A, \tau'(B))$ 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 but 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

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

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. *For minimizing the discrete Fréchet distance between two 2D polygonal chains under translation and rotation, we have an $O(m^4n^4 \log(1/\epsilon))$ time $1+\epsilon$ approximation algorithm and an $O(m^4n^4 \log(m+n))$ 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 form 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 three dimensions in roughly $O((mn)^7)$ time (ignoring the log factors) under both translation and rotation. (Instead of using only three pairs of reference vertices as in the 2D case, six pairs of reference vertices are necessary for the six degrees of freedom in the 3D case.) Although this $O((mn)^7)$ running time for the discrete Fréchet distance is far less than the current best $O((m+n)^{20} \log(m+n))$ running time for the continuous Fréchet distance,[8] it is still 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 $A = \langle a_1, a_2, \ldots, a_m \rangle$ and $B = \langle b_1, b_2, \ldots, b_n \rangle$, if $d_{\mathcal{F}}(A, B) = \epsilon$, then we must have both $dist(a_1, b_1) \leq \epsilon$ and $dist(a_m, b_n) \leq \epsilon$. 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 $1 \leq i \leq n$, $dist(a_i, b_i) \leq \epsilon$. It follows that $dist(a, b) \leq \epsilon$, 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 for each chain. We note that the use of reference points in geometric pattern matching has been studied by Aichholzer *et al.*[11]; there are also related works on reference points for matching by using the Fréchet distance measure.[7,12] 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 $\triangle a a_1 a_m$ and $\triangle b b_1 b_n$ are coplanar, 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[b] 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.

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 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 1 second to align two chains of lengths 325 on our test machine. Figure 2 shows two screenshots of our program, before (left) and after (right) aligning the two protein chains 1o7j.a and 1hfj.c.

We compare our heuristic with ProteinDBS,[14] 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 2D 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.[14] According to the

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

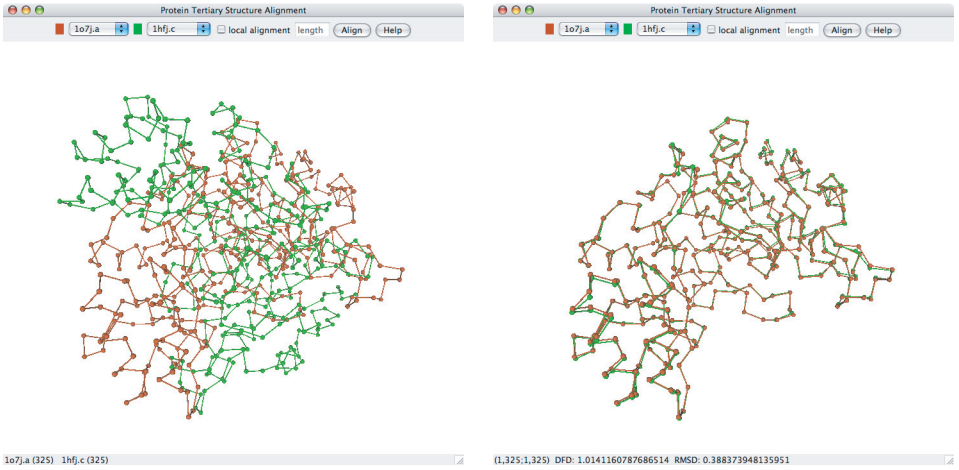


Fig. 2. The alignment of 1o7j.a and 1hfj.c by our heuristic.

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

Protein chain	Alignment length	RMSD[a] (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

[a]RMSD: root mean square deviation.

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.

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.

Instead of using the unweighted centers as the reference points in our heuristic, a possible alternative (as pointed out by the anonymous referees of an earlier version of this paper) is to use the Steiner points[7,11,12] (which can be considered as the

weighted centers of the protein chains) to achieve protein structure–structure alignments with certain provable qualities. We defer the exploration of this interesting idea to our future work.

4. Conclusion

In this paper, we presented the first algorithms for matching two polygonal chains in two dimensions 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 biomolecules 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 DALI[15] and CE.[16] We see great potential for using the discrete Fréchet distance in the local alignment,[17] the feature identification, and the consensus shape construction[18] of multiple proteins.

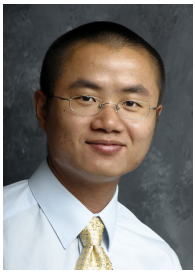
Acknowledgments

M. Jiang was supported by the USU research fund A13501. Y. Xu was supported partly by the National Science Foundation (NSF/DBI-0354771, NSF/IIS-0407204, NSF/DBI-0542119, NSF/CCF-0621700) and partly by a Distinguished Cancer Scholar grant from the Georgia Cancer Coalition.

References

1. Huttenlocher DP, Kedem K, Sharir M, The upper envelope of Voronoi surfaces and its applications, in *Proceedings of the 7th Annual Symposium on Computational Geometry (SoCG'91)*, pp. 194–203, 1991.
2. Huttenlocher DP, Kedem K, Kleinberg JM, 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)*, pp. 110–119, 1992.
3. Agarwal PK, Sharir M, Toledo S, Applications of parametric search in geometric optimization, in *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'92)*, pp. 72–82, 1992.
4. Alt H, Behrends B, Blömer J, Approximate matching of polygonal shapes (extended abstract), in *Proceedings of the 7th Annual Symposium on Computational Geometry (SoCG'91)*, pp. 186–193, 1991.
5. Alt H, Godau M, Measuring the resemblance of polygonal curves, in *Proceedings of the 8th Annual Symposium on Computational Geometry (SoCG'92)*, pp. 102–109, 1992.
6. Alt H, Godau M, Computing the Fréchet distance between two polygonal curves, *Int J Comput Geometry Appl* **5**:75–91, 1995.

7. Alt H, Knauer C, Wenk C, Matching polygonal curves with respect to the Fréchet distance, in *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS'01)*, pp. 63–74, 2001.
8. Wenk C, Shape matching in higher dimensions, Ph.D. thesis, Freie Universitä Berlin, Berlin, Germany, 2002.
9. Eiter T, Mannila H, Computing discrete Fréchet distance, Technical Report CD-TR 94/64, Information Systems Department, Technical University of Vienna, Vienna, Austria, 1994.
10. Mosig A, Clausen M, Approximately matching polygonal curves with respect to the Fréchet distance, *Comput Geometry Theory Appl* **30**(2):113–127, 2005.
11. Aichholzer O, Alt H, Rote G, Matching shapes with a reference point, *Int J Comput Geometry Appl* **7**(4):349–363, 1997.
12. Knauer C, Algorithms for comparing geometric patterns, Ph.D. thesis, Freie Universitä Berlin, Berlin, Germany, 2002.
13. Cole R, Slowing down sorting networks to obtain faster sorting algorithms, *J ACM* **34**:200–208, 1987.
14. Shyu C-R, Chi P-H, Scott G, Xu D, ProteinDBS: A real-time retrieval system for protein structure comparison, *Nucleic Acids Res* **32**:W572–W575, 2004.
15. Holm L, Sander C, Mapping the protein universe, *Science* **273**(5275):595–602, 1996.
16. Shindyalov IN, Bourne PE, Protein structure alignment by incremental combinatorial extension (CE) of the optimal path, *Protein Eng* **11**(9):739–747, 1998.
17. Gusfield D, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, Cambridge, UK, 1997.
18. Chew LP, Kedem K, Finding the consensus shape of a protein family, in *Proceedings of the 18th Annual Symposium on Computational Geometry (SoCG'02)*, pp. 64–73, 2002.



Minghui Jiang is an Assistant Professor of Computer Science at Utah State University, USA. He received a B.S. in Physics from Peking University, China, in 1997; two M.S. degrees in Computer Science and Physics from Purdue University, USA, in 1999; and a Ph.D. in Computer Science from Montana State University, USA, in 2005. His research interests are in the design and analysis of algorithms, discrete and computational geometry, bioinformatics and computational biology, and combinatorial optimization.

Ying Xu is an Endowed Professor under the title "Regents–Georgia Research Alliance Eminent Scholar" in the Biochemistry and Molecular Biology Department, as well as Director of the Institute of Bioinformatics, at the University of Georgia (UGA), USA. Before joining UGA in September 2003, he was a senior staff scientist and group leader at Oak Ridge National Laboratory (ORNL), USA, where he still holds a joint position. He received his Ph.D. in Theoretical Computer Science from the

University of Colorado at Boulder, USA, in 1991. His Ph.D. thesis was on the development of efficient algorithms for matroid intersection problems (supervised by Hal Gabow). Between 1991 and 1993, he was a Visiting Assistant Professor at Colorado School of Mines. He started his bioinformatics career in 1993, when he joined Ed Uberbacher's group at ORNL to work on the GRAIL project. His current research interests include computational inference and modeling of biological pathways and networks, particularly for microbial organisms; comparative genome analyses; protein structure prediction and modeling; and cancer bioinformatics. He is interested in both bioinformatic tool development and study of biological problems using *in silico* approaches. Having over 200 publications (including 4 books), Prof. Xu has also given over 150 invited/contributed talks at conferences, research organizations, and universities. He currently serves on the editorial boards of four international journals, and is Co-Editor-in-Chief of the "Bioinformatics and Computational Biology" book series by World Scientific Publishing.

Binhai Zhu is a Professor in Computer Science at Montana State University, USA. He obtained his Ph.D. in Computer Science from McGill University, Canada, in 1994; and was a postdoctoral research associate at Los Alamos National Laboratory, USA, from 1994 to 1996. From 1996 to 2000, he was an Assistant Professor at City University of Hong Kong, Hong Kong. He has been at Montana State University since 2000 (Associate Professor until 2006, Professor since 2006). Professor Zhu's research interests are geometric computing, biological/geometric modeling, bioinformatics, and combinatorial optimization. He has published over 90 papers in these areas.