To appaer in Proc. 8th International Conference on Similarity Search and Applications (SISAP 2015),
Lecture Notes in Computer Science, Springer, 2015 (12-14 October 2015, Glasgow, Scotland, UK)

Efficient Approximate 3-Dimensional Point Set Matching Using Root-Mean-Square Deviation Score

Yoichi Sasaki[1], Tetsuo Shibuya[2], Kimihito Ito[3], and Hiroki Arimura[1]

[1] IST, Hokkaido University, Sapporo, Japan, {ysasaki,arim}@ist.hokudai.ac.jp
[2] University of Tokyo, Tokyo, Japan, tshibuya@hgc.jp
[3] CZC, Hokkaido University, Sapporo, Japan, itok@czc.hokudai.ac.jp

Abstract. In this paper, we study approximate point subset match (APSM) problem with minimum RMSD score under translation, rotation, and one-to-one correspondence in d-dimension. Since this problem seems computationally much harder than the previously studied APSM problems with translation only or distance evaluation only, we focus on speed-up of exhaustive search algorithms that can find all approximate matches. First, we present an efficient branch-and-bound algorithm using a novel lower bound function of the minimum RMSD score. Next, we present another algorithm that runs fast with high probability when a set of parameters are fixed. Experimental results on real 3-D molecular data sets showed that our branch-and-bound algorithm achieved significant speed-up over the naive algorithm still keeping the advantage of generating all answers.

Keywords: 3D point set matching, RMSD, geometric transformation, one-to-one correspondence, branch and bound, probabilistic analysis

1 Introduction

1.1 Background. The approximate point set matching (APSM) is one of the fundamental problems in computer science, while it plays important roles in many application areas including molecular biology, image retrieval, pattern recognition, music information retrieval, and geographic information systems [12]. For every $d \geq 1$, the d-dimensional approximate point set matching problem considered in this paper can be described as follows. An input consists of a data set T and a pattern set P of n and k points in $\mathbb{R}^d$, respectively, and a positive integer $r > 0$, called distance threshold. The task is finding some point subset $Q \subseteq T$ of k data points that are similar to $P \subseteq \mathbb{R}^d$ w.r.t. a given distance measure under some transformation f, such as translation and rotation, and under some correspondence π between the elements of two sets.

In molecular biology, for instance, such an algorithm for solving 3-D APSM can be used to predict the unknown function of a given target protein with known structure. To do this, we search a database of proteins with known functions for structurally similar proteins that may indicate the unknown function.

In real molecular databases, individual data entry of molecules in a database may contain measurement errors, and may have different origin, coordinate, and numbering of data points from data to data. Therefore, a point set matching algorithm should have capability of finding approximation matches, and moreover should be tolerant under translation, rotation, and also 1-1 correspondence between points. For this reason, we focus on point set matching of this kind.

1.2 Related work. APSM and its variants have been extensively studied for many years (See survey [12]). Our version of APSM problem in this paper is equipped with full of invariance requirements, that is, all of invariance under rotation, translation, and 1-1 correspondence. However, this invariance makes the problem computationally much harder than the previously studied APSM problems. Most of previous theoretical results on efficient APSM problems under rigid motion seem to fall into two categories: one is an point subset match problem [2], and another is a distance evaluation problem [3] that detects the congruence between two point sets *of same size*.

For example, in the first category, de Rezende and Lee [8] presented $O(kn^d)$ time exact point subset matching algorithm for $d \geq 2$. However, it seems difficult to extend their algorithm for approximate matching. For approximate point subset matching, Goodrich *et al.* [10], and later Cho and Mount [6], presented simple constant approximation algorithms for $d = 2, 3$ under directed Hausdorff distance based on aligning pairs of points. Although this algorithm has quadratic time complexity in n, it only has constant approximation ratio larger than three [6]. In the second category, Alt *et al.* [3] presented an approximate distance evaluation algorithm under rigid motion that runs in polynomial time in n for $d = 2, 3$. We note that all of above results were obtained for Hausdorff distance which allows many-to-one correspondence between points.

The selection of distance score gives another dimension to the APSM problem. In this paper, we consider the *minimum root-mean-square deviation score* (*minimum RMSD score*) between point sets, which is widely used similarity score in molecular biology [5]. This score requires that there is a 1-1 correspondence π between a transformed set $f(P)$ and Q such that the sum of the squares of Euclidean distances between each point p in $f(P)$ and its counterpart q in Q is within a given threshold r, that is,

$$r \geq \min_{\pi} \min_{f} \sqrt{\frac{1}{n} \sum_{i=1}^{n} ||f(p_i) - q_{\pi(i)}||^2} \tag{1}$$

for some candidate subset $Q \subseteq T$. This minimum RMSD score resembles the *directed approximate congruence* [3] between two k-point sets except that the latter requires that there is a 1-1 correspondence that maps each point in $f(P)$ to its counterpart q in Q within distance ε. In other words, the distance between $f(P)$ and Q is measured in L_2-norm in our problem, while it is measured in L_∞-norm in [3]. Overall, from the above discussions, most of the previous results on APSM do not apply to our problem.

1.3 Research goal. In this paper, we consider the approximate 3-D point set matching problem with respect to the minimum RMSD score for sets under both of translation and rotation and under 1-1 correspondence (APSM(trans, rot, 1-1; RMSD), for short). In addition,we are also interested in finding *all* approximate matches rather than *just one* or *some* matches. A straightforward approach to solve this problem is to use exhaustive search, which enumerates all candidate point subsets of T and all 1-1 correspondences between a pattern and each of them. Then, for each enumerated candidate, we test the minimum RMSD score under translation and rotation by some matrix computation such as [15]. However, in practice, one serious problem with this exhaustive search method is its exact exponential time complexity. Since the method must exactly enumerate all of $\binom{n}{k} = n^{\Theta(k)}$ combination of k data points in T, it always visits the leaves of the search tree regardless of the content of input sets.

1.4 Main results of this paper. To overcome this difficulty, we study a branch-and-bound algorithm for APSM(trans, rot, 1-1; RMSD) problem. This algorithm finds all approximate matches within threshold r by systematically enumerating all 1-1 correspondences between candidate subsets of k data points from smaller to larger prefixes based on recursive computation. At each iteration of the search, it tests if the current candidate prefix of size $i < k$ satisfies a given lower bound function, and then prune the search if the test failed. As a key of the algorithm, we show that the proposed lower bound function is sound in the sense that it cannot prune any successful search branches. Although the time complexity of the obtained algorithm is still $n^{O(k)}$ in n and k in the worst case, the algorithm can make early pruning depending on the content of input data.

We also present a *fixed-parameter-tractable* style algorithm [9] that runs particularly fast in terms of the data set size n when the parameters, namely, the size k and radius ℓ of a pattern set, and the distance threshold r are fixed. For each data point, the algorithm forms a small sample set consisting of all data points within some fixed distance, and then applies any APSM algorithm to the sample set. Assuming the spatial Poisson process [14] in $\mathbb{R}^d$, we show that the algorithm runs in linear time in expected case, and runs in $O(n \log^k n)$ time and $O(\log n)$ space with high probability for fixed parameters.

Finally, the experimental results on real data sets of 3-D molecules showed that the proposed branch-and-bound algorithm was one to two order of magnitude faster than the straightforward exhaustive search algorithm. Hence, our lower bound function and pruning technique achieve significant speed-up of approximate 3-D point subset matching.

1.5 Organization of this paper. In Sec. 2, we introduce the basic definitions and notations related to the approximate point subset matching problem with the minimum RMSD under translation, rotation, and 1-1 correspondence. First in Sec. 3, we present the branch-and-bound algorithm using lower bound function over candidate prefixes. In Sec. 4, we present the fixed parameter algorithm and gives analysis of its complexity. Finally, Sec. 6 concludes the paper.

2 Preliminaries

We give brief review of basic concepts and notations in geometry [7]. Then, we introduce our point subset matching problem.

2.1 Basic definitions. We denote by $\mathbb{R}$ and $\mathbb{N}$ the sets of all real numbers and integers, respectively. For a real-valued k-vector $Q = (q_1, \ldots, q_k) \in \mathbb{R}^k$, we denote its L_2- and L_∞-norms by $L_2(Q) = \left\{\frac{1}{k}\sum_i |q_i|^2\right\}^{1/2}$ and $L_\infty(Q) = \max_i |q_i|$. For any k-vector Q, we have the inequality $\frac{1}{k^{1/2}} L_\infty(Q) \leq L_2(Q) \leq L_\infty(Q)$. For a matrix or a vector A, $A^\top$ denotes the *transpose* of A.

Let $d \geq 1$ be the dimension of the space. In this paper, we consider the 3-D space $\mathbb{R}^3$, but all the results also apply to the d-dimensional space for every fixed $d \geq 1$. An element $P = (p_1, \ldots, p_d)^\top \in \mathbb{R}^d$ of the space $\mathbb{R}^d$ is called a *point*. The *Euclidean distance* (or *distance*, for short) between points p and q is given by the L_2-norm $||p - q|| = L_2(p - q)$.

For a *point sequence* $S = (s_1, \ldots, s_k) \in (\mathbb{R}^d)^k$ of length k in $\mathbb{R}^d$ and any $0 \leq i \leq k$, we define the *i-prefix* of S as the subsequence $S[1..i]$ consisting of the first i points of S. A *point set* is an unordered collection $T = \{t_1, \ldots, t_n\}$ of points in the space. We define the *size* of T by the number of its elements $|T| = n$. In what follows, we represent a set as a point sequence by assuming some fixed ordering of the elements. For any $k \geq 0$, a point set P is a *k-point set* (or *k-set*, for short) if $|P| = k$.

2.2 The minimum RMSD score for k-point sets. From now on, we introduce the distance score *MinRMSD* between two k-point sets under translation, rotation and 1-1 correspondence.

Let $P = \{p_1, \ldots, p_k\}$ and $Q = \{q_1, \ldots, q_k\}$ be two k-point sets in $\mathbb{R}^3$. In what follows, we assume an arbitrary fixed ordering over the indices since the following discussion does not depend on this choice of ordering. Based on this, we often regard P and Q as point sequences by assuming underlying ordering. We first try to *align* the points in P with the points in $\pi(Q) = \{Q_{\pi(1)}, \ldots, Q_{\pi(k)}\}$ permuted by a *1-1 correspondence* π between P and Q, that is, any 1-1 mapping π over indices so that p_i and $q_{\pi(i)}$ correspond each other. We denote by $\mathcal{O}$ the class of all 1-1 correspondences over $\{1, \ldots, k\}$.

A geometric transformation is any 1-1 mapping $f : \mathbb{R}^d \to \mathbb{R}^d$ over d-dim space [3, 7]. A *rigid motion* is a transformation generated by any combination of translation, rotation, and reflection. A rigid motion is a ransformation that does not change the distance. It is known that matching for P under rigid motion can be reduced to matching for P and its reflection under translation and rotation [3]. We denote by $\mathcal{RT}$ the class of all compositions of rotations and translations. It is well known that any transformation f in $\mathcal{RT}$ is obtained by application of one d-dim rotation matrix $R \in \mathbb{R}^{d\times d}$ and one d-dim translation vector $v \in \mathbb{R}^d$ such that $f(p) = Rp + v$ for any point $p \in \mathbb{R}^d$ (See [3]). For a k-set $P = \{p_1, \ldots, p_k\}$, we extend f by $f(P) = \{Rp_1, \ldots, Rp_k\} = RP$ assuming the correspondence. In what follows, we identify a transformation f in $\mathcal{RT}$ and the associated pair (R, v) if it is clear from context.

Assuming a 1-1 correspondence π between k-sets P and Q, the *root-mean-square deviation* (*RMSD*) between P and $\pi(Q)$ is defined as the average of the squared Euclidean distances between the corresponding pairs of points $||p_i - q_{\pi(i)}||$ for $i = 1, \ldots, k$. Then, the *minimum root-mean-square deviation* under the class $\mathcal{RT}$ of rotations and translations, denoted by $MinRMSD(P, Q)$, is defined as the minimum value of the RMSD score between $f(P)$ and $\pi(Q)$ over all transformations in RT given by

$$MinRMSD(P, Q) \;=\; \min_{\pi} \min_{f} \sqrt{\frac{1}{k} \sum_{i=1}^{k} ||f(p_i) - q_{\pi(i)}||^2}, \tag{2}$$

where π ranges over all 1-1 correspondences over $\{1, \ldots, k\}$, and f ranges over all transformations $f_{R,v}$ in $\mathcal{RT}$ specified by a rotation R and a translation v in d-dim space. If π is already specified, it is known that the optimal transformation f in $\mathcal{RT}$ minimizing $RMSD(P, \pi(Q))$ can be computed in linear time in fixed $d \geq 1$ [11, 16] by using singular value decomposition (SVD) after aligning the centroid of P and $\pi(Q)$. Such a linear time procedure does not seem to be known for Hausdorff distance.

2.3 Approximate point subset matching problem. Let $1 \leq k \leq n$ be any positive integers. A *data set* and a *pattern set* are sets $T = \{t_1, \ldots, t_n\}$ and $P = \{p_1, \ldots, p_k\}$ of points in $\mathbb{R}^d$, respectively. Each member of P (Q, resp.) is called a *data point* (a *pattern point*, resp.). A *distance threshold* is any positive real number $r > 0$. A *match* for pattern set P w.r.t. r is any k-subset Q of T such that $MinRMSD(P, Q) \leq r$ holds.

Now, we state our approximate pattern matching problem as follows. A *candidate* k-subset in T is any k-subset of T.

Definition 1 (APSM(trans, rot, 1-1; RMSD) problem). The *approximate point set matching problem with MinRMSD score under translation, rotation, and 1-1 correspondence*, abbreviated as APSM(trans, rot, 1-1; RMSD), is defined as follows: Given a data set $T \subseteq \mathbb{R}^d$ of n points, a pattern set $P \subseteq \mathbb{R}^d$ of k points, and a positive real number $r > 0$, find all match $Q \subseteq T$ of k data points that satisfy the condition $RMSD(f(P), \pi(Q)) \leq r$,

The above definition is the *enumeration version* of APSM problem. In the *decision version* of the APSM problem, given T, P, and r, an algorithm must decide if $MinRMSD(P, Q) \leq r$. In the *optimization version*, given T and P, an algorithm must find some 1-1 correspondence π and transformation f in $\mathcal{RT}$ that minimizes $MinRMSD(f(P), \pi(Q))$. In what follows, we present algorithms for the enumeration version of APSM. It is not hard to convert these algorithms to solve the decision and optimization versions in the same time and space complexity though the converse is not true in general.

2.4 A naive algorithm for approximate point subset matching. As the basis of our discussion, in Algorithm 1, we show the naive exhaustive search algorithm for 3-D APSM(trans, rot, 1-1; RMSD) problem in $kn^{\Theta(k)}$ time. In this

Algorithm 1 A naive algorithm for solving the enumeration version of 3-D APSM(trans, rot, 1-1; RMSD) problem using exhaustive search

```
1: procedure MatchNaive(P, T, r)
   Input: A text point set T[1..n], a pattern point set P[1..k], a real number r > 0.
   Output: All matchings of P in T with minimum RMSD score no larger than r.
2:     FindNaive((), 0, |P|, |T|, P, T);

3: procedure FindNaive(Q = (q_1, ..., q_i), i, k, n, P, T)
4:     if i = k then                                   ▷ Q becomes a k-subset
5:         if MinRMSD(P, Q) ≤ r then
6:             Report Q as a match;
7:             return;
8:     for j = 1, ..., n do
9:         if T[j] ∉ Q then
10:            FindNaive((q_1, ..., q_i, T[j]), i + 1, k, n, P, T);   ▷ Recursive call
```

algorithm, the subprocedure FindNaive starts with the *empty prefix* $Q = ()$ and $i = 0$. Then, it recursively traverses the search space of *i-candidate prefixes* $Q = Q[1..i]$ from smaller to larger for all $i = 0, \ldots, k$, where each $Q[1..i]$ represents an ordered set of i data points $\{Q_{\pi(1)}, \ldots, Q_{\pi(i)}\}$ with correspondence π. At each iteration, it grows the current candidate prefix by appending a new data point from $T \setminus Q$. Whenever the condition $|Q| = k$ holds, it tests if the condition $\mathit{MinRMSD}(P[1..i], Q[1..i]) = \min_f \mathit{MinRMSD}(f(P[1..i]), \pi(Q)) \leq r$ in linear time in k assuming π as mentioned in Sec. 2.2. Since each iteration takes $O(k)$ time, the total running time is $O(ks) = kn^{O(n)}$ time, where $s = n^{O(k)}$ is the number of all i-prefixes with $i \leq k$. One problem with this algorithm is that it cannot make early termination, and thus always takes $kn^{\Theta(n)}$ time regardless of the data content.

3 A faster point set matching algorithm with pruning

In this section, we discuss speed-up of the naive algorithm in the previous section. We present an efficient point set matching algorithm MatchFast based on branch-and-bound search with early pruning.

The basic idea of our algorithm is using a lower bound function LB of *MinRMSD* score to make early pruning of unsuccessful branches. We design the lower bound function LB such that for any candidate i-prefix $Q = Q[1..i]$ consisting of $i \leq k$ data points, $LB(P[1..i], Q[1..i]) > r$ implies $\mathit{MinRMSD}(P, R)$ for any k-subset $R = R[1..k]$ of T that is an extension of Q such that $R[1..i] = Q[1..i]$. Based on this idea, we present a simple lower bound function for *MinRMSD* that our branch-and-bound algorithm uses.

Theorem 1 (sound lower bound function). *Let P and Q be k-point sets as sequences. For any integer $1 \leq i \leq k$, it holds that*

$$\mathit{MinRMSD}(P, Q) \geq \left(\frac{i}{k}\right)^{1/2} \mathit{MinRMSD}(P[1..i], Q[1..i]) \tag{3}$$

Algorithm 2 A faster branch-and-bound algorithm for solving the enumeration version of 3-D APSM(trans, rot, 1-1; RMSD) problem using exhaustive search

```
1: procedure MatchFast(P, T, r)
2:     FindFast((), 0, |P|, |T|, P, T);
3: procedure FindFast(Q = (q_1, ..., q_i), i, k, n, P, T)        ▷ candidate prefix Q
4:     if i = k then                                              ▷ Q becomes a k-subset
5:         if MinRMSD(P, Q) ≤ r then
6:             Report Q as a match;
7:     for j = 1, ..., n do
8:         if T[j] ∉ Q then then continue;
9:             R := (q_1, ..., q_i, T[j]);                        ▷ Append a new data point
10:            if (i/k)^{1/2} MinRMSD(P[1..i+1], R) > r then continue;   ▷ Pruning
11:            FindFast(R, i + 1, k, n, P, T);                    ▷ Call itself recursively
```

Proof. Consider the sum of squared distances $SSD(P,Q) = \sum_{i=1}^{k} ||f_{R,v}(P[i]) - Q[i]||^2$. Suppose that we append a pair of new points $P[i]$ and $Q[i]$ to P and Q. Then, we see that the minimum of $SSD(f(P[i]), Q[i])$ over all transformations f is larger than or equal to the sum of the squared distance $||P[i] - Q[i]||^2$ and the minimum of $SSD(f'(P[i-1]), Q[i-1])$ over all transformations f'. Since the minimum of $SSD(f(P[i]), Q[i])$ over all f equals $k \cdot (MinRMSD(P,Q))^2$, we have the equality $k \cdot MinRMSD(P[1..k], Q[1..k])^2 \geq i \cdot MinRMSD(P[1..i], Q[1..i])^2$ $(*)$. By taking the square root of the both side of (*), the theorem follows. □

From the above Theorem, we propose $(i/k)^{1/2}$ $MinRMSD(P[1..i], Q[1..i])$ as the lower bound function for $MinRMSD(P,Q)$. Based on the proposed lower bound function, in Algorithm 2, we present the MatchFast with the modified subprocedure FindFast. At each iteration, it searches for all i-prefixes as in the same manner as the naive algorithm does except that it makes the test the current candidate i-prefix $Q[1..i]$ at Line 10 based on the lower bound function LB, and prunes the search when the test failed. From Theorem 1, this pruning is sound without eliminating any successful branches. Furthermore, the test at Line 8 ensures to avoid duplicated enumeration of the same 1-1 correspondence. From the above discussion, we have the main theorem of this section.

Theorem 2. *The algorithm* MatchFast *in Algorithm 2 solves the enumeration version of APSM(trans, rot, 1-1; RMSD), the approximate 3-D point set matching problem with minimum RMSD score under translation, rotation, 1-1 correspondence, in* $O(\binom{n}{k}k)) = n^{O(k)}k$ *time.*

Although the worst case time complexity of MatchFast still remains $n^{O(k)}k$ time, it can make early termination depending on the content of an input data.

4 A fixed-parameter-like algorithm using spatial constraint

In this section, we present the second modified algorithm MatchFP, which is inspired by fixed-parameter tractable algorithms [9], that is particularly fast

for small patterns on uniformly distributed data points. In the followings, let $d = 2, 3$ be a fixed dimension, and $\theta = (k, r, \ell)$ be the tuple of parameters such that k and ℓ is the size and radius of pattern P, $r > 0$ is a distance threshold, and $\varepsilon > 0$ is a positive number explained later.

4.1 Basic idea. For any positive number $\ell > 0$, we define the *ball* $B_{c,\ell} = \{ q \in \mathbb{R}^d \,|\, ||q - c|| \leq \ell \}$. with radius $\ell > 0$ centered at a point $c \in \mathbb{R}^d$, whose volume is $|B_{c,\ell}| = (4\pi/3)\ell^3$ for $d = 3$. Then, the *radius* of a point set P, denoted by *radius*(P), is the minimum radius of the ball containing all points of P. The *maximum neighbor distance* within a given ball $B \subseteq \mathbb{R}^d$ is the maximum of the distance to the nearest neighbor of each data point in B defined by $\varepsilon = \max_{p \in B} \min_{q \in T} ||p - q|| > 0$. The next lemma says that if there is a match between P and some k-subset $Q \subseteq T$, such a candidate can be found in a small ball around P when the size and radius of P, threshold r, and the maximum neighbor distance are bounded.

Theorem 3 (Locality of match). *Let P be any pattern set with the center $c \in \mathbb{R}^d$ and radius $\ell > 0$, and $r > 0$ be any number. If $RMSD(f(P), Q) \leq r$ holds for some transformation f in $\mathcal{RT}$ and some candidate k-point set $Q \subseteq T$, then Q must be contained within the ball centered at $c' = f(c) \in \mathbb{R}^d$ with radius*

$$L = L(k, r, \ell) := 2(k^{1/2} r + \ell) \tag{4}$$

Proof. Let $P = (p_i)_{i=1}^k$ and $Q = (q_i)_{i=1}^k$. It is sufficient to show $||f(c) - q_i|| \leq L$ holds for any i. Let $1 \leq i \leq k$ be any index. Note that $||f(c) - q_i|| \leq ||f(c) - f(p_i)|| + ||f(p_i) - q_i||$ (*) from the triangular inequation on L_2-norm. First, we see that $||f(c) - f(p_i)|| \leq \ell$ since $||c - p_i|| \leq$ *radius*$(P) = \ell$ by assumption. Next, by assumption, $r \geq RMSD(f(P), Q) = L_2(f(P) - Q)$ holds. Since $L_2(X) \geq (1/k)^{1/2} L_\infty(X)$ holds for any k-vector X, we have $L_2(f(P) - Q) \geq (1/k)^{1/2} L_\infty(f(P) - Q) \geq (1/k)^{1/2} ||f(p_i) - q_i||$(**). Multiplying the both side of (**) by $k^{1/2} \geq 0$, we have $||f(p_i) - q_i|| \leq k^{1/2} r$. By combining above arguments with (*), we have $||f(c) - q_i|| \leq \ell + k^{1/2} r$. Applying this formula again, we also see that $f(c)$ has the nearest data point $t_c \in T$ such that $||t_c - f(c)|| \leq \varepsilon$ for $\varepsilon := \ell + k^{1/2} r$. Thus, again from triangular inequality, we have the result $||t_c - f(p_i)|| \leq ||t_c - f(c)|| + ||f(c) - q_i|| \leq \varepsilon + (\ell + k^{1/2} r) = L$. □

In Algorithm 3, we present the algorithm MATCHFP for APSM(trans, rot, 1-1; RMSD) based on Theorem 3 that runs particularly fast on uniformly distributed data points for fixed parameter values $\theta = (k, r, \ell)$. This algorithm first computes the parameter $L = L(k, r, \ell)$ according to Theorem 3. Then, it works on iterations with each t among n data points in T. In Step 1, it computes the *local data set* T_t in $O(polylog(n) \times |T_t|)$ time using, e.g., the range tree index [7]. In Step 2, the algorithm computes matchings on T_t using MATCHFAST in Sec. 3 in $t = O(N(B_{t,L})^k k)$ time and $s = O(N(B_{t,L}))$ working space. Then, it repeats the above process for all of n data point t.

Lemma 1. *If $L \geq L(k, r, \ell)$, then the algorithm* MATCHFP *in Algorithm 3 solves the 3-D APSM(trans, rot, 1-1; RMSD) problem in $t = O(\sum_{t \in T} N(B_{t,L})^k k)$ time and $s = O(\max_{t \in T} N(B_{t,L}))$ working space.*

Algorithm 3 A fixed-parameter algorithm MATCHFP that solves the enumeration version of 3-D approximate PSM with RMSD score under translation, rotation, and 1-1 correspondence.

Given a data point et T, a pattern point set P, distance threshold $r > 0$, and real number $0 < \delta \leq 1$, the algorithm MATCHFP executes the following steps for each data point t in T:

- **Step 1**: Compute the set T_t of all data points in the ball $B_{t,L}$ with radius $L = L(k, r, \ell) := 2(k^{1/2}r + \ell)$.
- **Step 2**: Apply the algorithm MATCHFAST(T_t, P, r) in Sec. 3 to the restricted data set T_c centered at t to find and output all matchings $Q \subseteq T_t$ within T_t.

4.2 Probabilistic analysis. The *spatial Poisson process* (SPP, for short) with mean parameter $\lambda > 0$ is a model of uniform distribution of random points in $\mathbb{R}^d$ [14] having density λ. In SPP, (S1) for any ball B, the distribution of count $N(B)$ obeys *Poisson distribution with mean* $\lambda|B| > 0$, i.e., $Pr(N(B) = k) = ((\lambda|B|)^k/k!)e^{-\lambda|B|}$. Moreover, (S2) for any disjoint regions $A_1, \ldots, A_m$, $N(A_1), \ldots, N(A_m)$ are independent. Now, we show the main theorem of this section which says that MATCHFP runs particularly fast when parameter k, r, and ℓ are small constant over uniformly generated data sets.

Theorem 4. *Suppose that data points are generated by SPP with density* $\lambda > 0$. *We fix the following parameters* $\theta = (k, \ell, r)$*: the maximum size* $k > 0$ *and radius* $\ell > 0$ *of a pattern, a distance threshold* $r > 0$. *Then, for any* $\delta > 0$*, the following conditions holds:*

> *For any data set* T *of* n *points of arbitrary radius and a pattern set* P *of* k *points with radius at most* ℓ*, if we set the radius of the local data set* T_t *to be* $L = L(k, r, \ell)$ *in Theorem 3, then* MATCHFP *in Algorithm 3 solves the 3-D* APSM*(trans, rot, 1-1; RMSD) problem in* $O(n \log^k n)$ *time and* $O(\log n)$ *working space with probability at least* $1 - \delta$.

Proof. We give a proof sketch. By assumption of SPP, for any ball $B_L = B_{t,L}$, $Pr(N(B_L) = k)$ is given by Poisson distribution with mean $\lambda|B_L|$ regardless of the location of t. Let $c > 0$ be a number that will be specified later. From Lemma 1, we see that the algorithm has the claimed complexity if the following situation does not happen; err_*: the local set size $|T_i|$ exceeds $c\lambda|B_L|$. Assuming SPP, we can show upper bounds of the failure probability $Pr(err_*)$ as follows. By applying tail bound for Poisson distribution to the union bound, we can show that $Pr(err_*) \leq \sum_{t \in T} Pr\{N(B_{t,L}) > c\lambda|B_{t,L}|\} = n \cdot Pr\{N(B_L) > c\lambda|B_L|\} \leq \delta$ for some $c \geq (1/\lambda|B|)\{-\ln(\delta) + \ln(n) + \ln(2)\} = const(\delta) + O(\ln(n))$. From the above discussion, the local set size satisfies $|T_i| \leq c\lambda|B| = O(\ln(n))$ with probability at least $1 - \delta$. Hence, the theorem follows from Lemma 1. □

5 Experiments

In this section, we give experimental results on real point data sets to evaluate the efficiency of the proposed algorithms in Sec.3.

5.1 Data and method. As a real data set, we used the molecular 3-D point set of one variation [4] of the protein called *Hemagglutinin HA1 chain of influenza A virus* (H10N8) from RCSB Protein Data Bank (PDB) [5]. In the followings, the units of length and distance are Å(=0.1nm = 1.0×10^{-10}m). For each parameter n up to 100, among 3722 atoms including 477 C_α atoms in the original data, we formed a data set of size n by extracting a subset of n locations of C_α carbon atoms, which give the approximated skeleton of a part of the molecule. The radius and the average distance between neighbor atoms were 45.31 unit and 2.2 unit, respectively. For each k up to 50, a pattern set of size k was formed by randomly selecting k points from the data set.

We implemented the naive algorithm in Sec.2 (naive) and the modified algorithm in Sec.3 (pruned) in C++ with Eigen linear algebra package [6]. As the experimental environment, we used a PC (CPU Intel Core i5 2.6 GHz, 8GB memory) and compiler (g++, Apple LLVM version 6.0, clang-600.0.54) with -O3 option. In the experiments, we measured the average of running time over four trials as well as the number of matches by varying input size $n = |T|$, pattern size $k = |P|$, and distance threshold r. We used default values of $n = 100$, $k = 3$, and $r = 0.1$ otherwise stated.

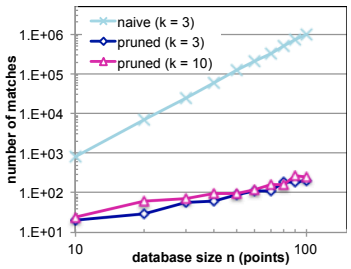
5.2 Results. We show the experimental results in of (a)–(d) of Fig. 1.

Exp 1: Running time and number of visited prefixes varying input data size: First, in (a) and (b) of Fig. 1, we show the number of visited candidates and running time by varying the data set size n from 10 to 150. From (a), we observed that the proposed branch-and-bound algorithm pruned using the lower bound function in Sec. 3 could effectively reduce the number of visited candidates to the 1/4 to 1/4800 of the original naive. From (b), we also observed that by this reduction of the number of visited candidates, pruned successfully achieved 50 to 600 times speed-up over naive,

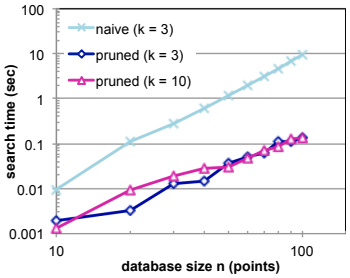
Exp 2: Running time varying pattern size: In (c) of Fig. 1, we show the running time by varying the pattern size k from 1 to 50, where $n = 100$ and $r = 0.1$. Note that we could run naive up to $k = 3$ because its running time exceeded the upper bound of 120 seconds for $k > 3$. From the figure, we see that the running time of naive showed exponential growth in k as expected from theoretical upper bound $n^{\Theta(k)}$, while that of pruned quickly grew up to $k = 2$ as same as naive and was slowly increasing after $k = 3$. Hence, we can conclude that the proposed pruning method with lower bound function is effective for matching with large pattern size k.

Exp 3: Running time varying distance threshold: In (d) of Fig. 1, we show the running time by varying distance threshold r from 0.01 to 10.0. Note that the average neighbor distance and radius of the data set is 2.2 and 45.3 units, respectively. From the figure, we observed that the time reduction ratio of pruned to naive gets larger when r goes smaller, while the ratio approaches almost one when r goes larger. Thus, our technique is more effective for smaller r.

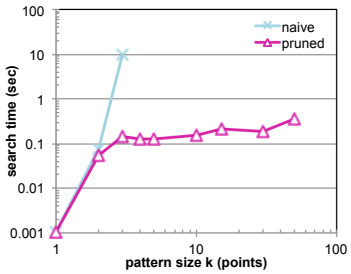
[4] The variation with structure ID 4XQ5 of H10N8 in PDB.
[5] http://www.rcsb.org/pdb/
[6] http://eigen.tuxfamily.org/



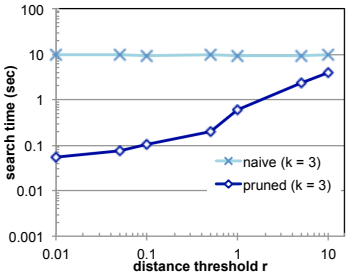
(a) Number of matches varying input size n, where $r = 0.1$, and $k = 3$ or 10.



(b) Running time varying input size n, where $r = 0.1$, and $k = 3$ or 10.



(c) Running time varying pattern size k, where $n = 100$ and $r = 0.1$.



(d) Running time varying distance threshold r, where $n = 100$ and $k = 3$.

Fig. 1: Comparison of the naive and proposed algorithms, naive and pruned, resp., where we set $n = 100$, $k = 3$, and $r = 0.1$ unless they are explicitly specified.

6 Conclusion

In this paper, we considered the approximate 3-D point set matching problem with the MinRMSD score under rotation, translation, and 1-1 correspondence, and then presented an efficient branch-and-bound algorithm based on a lower bound function. We also presented a FPT-style algorithm for fixed parameters. Experimental results showed that the first algorithm was one to two order of magnitude faster than the naive algorithm.

It will be a future work to compare the proposed algorithms and the existing constant approximation algorithms such as [6,10] to study trade-off between the time and accuracy. We also plan to apply the proposed algorithms to real world data sets in bioinformatics, 3D-modeling, and spatio-temporal data. *Point subset mining* [4,13] is a problem of finding point subsets from a point data set that meet a given criterion. Hence, it will be an interesting research problem how to use the proposed technique to speed-up *point subset mining* [4,13].

Acknowledgements. The authors are grateful to anonymous reviewers for their comments which significantly improved the correctness and the presentation of this paper, and also to Takeaki Uno, Kunihiko Sadakane, Koji Tsuda, Shin-ichi Minato, and Yutaka Akiyama for their comments on this work. This

research is supported in part by MEXT Grant-in-Aid for Scientific Research (A), 24240021, and the second author is also supported in part by CREST, JST, "*Foundations of Innovative Algorithms for Big Data*".

References

1. T. Akutsu. On determining the congruence of point sets in d dimensions. *Computational Geometry*, 9(4):247–256, 1998.
2. H. Alt and L. Guibas. *Discrete geometric shapes: Matching, interpolation, and approximation*, page 121153. Elsevier Science Publishers B.V. North-Holland, 1999.
3. H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity and symmetries of geometric objects. *Discret. Comput. Geom.*, 3:237–256, 1988.
4. H. Arimura, T. Uno, and S. Shimozono. Time and space efficient discovery of maximal geometric graphs. In *Discovery Science*, pages 42–55. Springer, 2007.
5. M. Carpentier, S. Brouillet, and J. Pothier. Yakusa: a fast structural database scanning method. *Proteins*, 61(1):137–151, 2005.
6. M. Cho and D. M. Mount. Improved approximation bounds for planar point pattern matching. *Algorithmica*, 50(2):175–207, 2008.
7. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2000.
8. P. J. de Rezende and D. Lee. Point set pattern matching in d-dimensions. *Algorithmica*, 13(4):387–404, 1995.
9. R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer, 1999.
10. M. T. Goodrich, J. S. Mitchell, and M. W. Orletsky. Approximate geometric pattern matching under rigid motions. *IEEE Trans. PAMI*, 21(4):371–379, 1999.
11. W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, A32(5):922–923, 1976.
12. V. Mäkinen and E. Ukkonen. Point pattern matching. In M. Kao, editor, *Encyclopedia of Algorithms*, pages 657–660. Springer, 2008.
13. S. Nowozin and K. Tsuda. Frequent subgraph retrieval in geometric graph databases. In *8th IEEE Int'l Conf. on Data Mining*, pages 953–958, 2008.
14. M. Pinsky and S. Karlin. *An introduction to stochastic modeling*. Academic press, 2010.
15. J. T. Schwartz and M. Sharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *The Int'l J. of Robotics Res.*, 6(2):29–44, 1987.
16. T. Shibuya. Geometric suffix tree: Indexing protein 3-d structures. *Journal of the ACM*, 57(3):15, 2010.
17. G. K. Tam et al. Registration of 3d point clouds and meshes: a survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graphics*, 19(7):1199–1217, 2013.