アルゴリズム 46−14
（1995．7．21）

# 二つの点集合の最大共通部分点集合を求めるランダマイズド・アルゴリズム

阿久津 達也
群馬大学 工学部 情報工学科

二つの点集合が与えられた時、各点集合の部分集合と合同で要素数最大の点集合を求める問題は、種々のパターンマッチング問題と関連するので重要である。二次元の場合において単純なアルゴリズムでは$O(n^5 \log n)$時間かかるが、本稿ではランダマイズド・アルゴリズムを用いることにより$O(n^4 \mathrm{polylog}(n))$時間で計算できることを示す。この結果は組合せ幾何の結果を利用することにより更に改善することが可能である。なお、このアルゴリズムは高次元に拡張することも可能であり、また、ここで使用した技法はタンパク質立体構造の近似アラインメントにも応用可能である。

# A Randomized Algorithm for Computing Largest Common Point Sets

Tatsuya Akutsu
Department of Computer Science, Gunma University
1-5-1 Tenjin, Kiryu, Gunma 376 Japan
e-mail: akutsu@cs.gunma-u.ac.jp

This paper considers the following problem: given two point sets in $d$-dimensions, find a point set of the maximum cardinality congruent to a subset of each set. This problem is important for several pattern matching problems. A naive deterministic algorithm works in $O(n^5 \log n)$ time in two-dimensions. This paper presents a randomized algorithm which works in $O(n^4 \mathrm{polylog}(n))$ time. Moreover the time complexities can be reduced using a known result in combinatorial geometry. The presented algorithm can be generalized for higher dimensions. Moreover a random sampling technique used in the algorithm can also be applied to a protein structure alignment algorithm.

# 1 Introduction

Finding a common part of two structures is important in several applications. In particular, it is important in molecular biology. For DNA sequences and amino acid sequences, such a problem has been studied as sequence alignment or string alignment. For three-dimensional protein structures, such a problem can be formalized as protein structure alignment. Several studies have been done for protein structure alignment [1, 8, 9]. However, the proposed algorithms do not seem sufficient since most of them have no theoretical proofs for the qualities of obtained alignments. Although we have developed a protein structure alignment algorithm which has a theoretical proof, the time complexity is high ($O(n^8)$) [1]. Although the average case time complexity can be reduced using several techniques [1], it is desirable to develop an algorithm with better worst-case time complexity. For that purpose, we consider the *largest common point set* problem since it is closely related to the protein structure alignment problem. Moreover this problem seems important in other pattern matching problems.

The largest common point set problem (LCP, in short) is, given two point sets $P$ and $Q$ in $d$-dimensional Euclidean space $R^d$, to find a point set of the maximum cardinality congruent to a subset of each set. In this paper, we present a randomized algorithm for LCP in $R^2$. It works in $\tilde{O}(\frac{n^4}{K})$ time, where $K$ denotes the size of the largest common point set, and $\tilde{O}(f(n))$ is a notation meaning $\tilde{O}(f(n)) = O(f(n)\text{polylog}(n))$ . Since a naive algorithm works in $\tilde{O}(n^5)$ time, it is a considerable improvement. Moreover, we show that the time complexity can be reduced using a result in combinatorial geometry.

Here we briefly review previous work. A lot of studies have been done for geometric pattern matching in computational geometry. Although most of them concern with approximate matching, several studies have been done for exact matching, which is considered in this paper. Alt et al. showed that the congruence of $d$-dimensional point sets can be tested in $O(n^{d-2}\log n)$ time [3]. Rezende and Lee showed that whether or not a point set of size $m$ matches any subset of a point set of size $n$ can be tested in $O(mn^d)$ time [7]. LCP for more than two point sets was discussed in [2]. Heffernan developed an approximate matching algorithm, which can also be applied to LCP [6]. However, the previous algorithms are not better than the naive algorithm if they are applied to LCP.

# 2 A Naive Algorithm

Let $P = \{\boldsymbol{p}_1, \cdots, \boldsymbol{p}_m\}$ and $Q = \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_n\}$ be point sets in $R^d$. We assume $m \leq n$ without loss of generality, from which $m = O(n)$ follows. A point set $C$ is called a *common point set* between $P$ and $Q$ if there exist subsets $P' \subseteq P$ and $Q' \subseteq Q$ each of which is congruent to $C$. $LCP(P,Q)$ denotes one of the largest common point sets between $P$ and $Q$. Although we focus on the two-dimensional case ($d = 2$) in this paper, the discussion can be generalized for higher dimensional cases.

LCP can be solved in polynomial time by the following naive algorithm [2]. Consider all combinations of $(\boldsymbol{p}_{i_1}, \boldsymbol{p}_{i_2})$ and $(\boldsymbol{q}_{j_1}, \boldsymbol{q}_{j_2})$. If $|\overline{\boldsymbol{p}_{i_1}\boldsymbol{p}_{i_2}}| = |\overline{\boldsymbol{q}_{j_1}\boldsymbol{q}_{j_2}}|$, this combination determines uniquely an isometric transformation $T$ such that $T(\boldsymbol{p}_{i_1}) = \boldsymbol{q}_{j_1}$ and $T(\boldsymbol{p}_{i_2}) = \boldsymbol{q}_{j_2}$ except a mirror image. Note that, in this paper, we ignore mirror images without loss of generality. We then count the points matched. Taking the maximum over all combinations, we can find $LCP(P,Q)$. This approach yields an $O(m^2 \times n^2 \times n\log n) = O(n^5 \log n)$ time algorithm.

# 3 A Randomized Algorithm

## 3.1 A Simple Algorithm

In this section, we apply a random sampling technique and obtain a faster randomized algorithm. To develop such an algorithm, we do not consider the optimization problem, but consider the following decision problem (LCPD): given a positive integer $K$ and two point sets $P$ and $Q$, output a common point set $C$ such that $|C| \geq K$ if such $C$ exists, otherwise output 'No'. Once we obtain a randomized algorithm for LCPD, we can obtain a randomized algorithm for LCP by means of the following proposition, where randomized algorithms considered in this paper are ones with one-sided error (i.e., they always output 'No' if there is no solution).

**Proposition 1:** If there is an $O(f(n))$ time randomized algorithm for LCPD with success probability at least $p$, we can obtain an $\tilde{O}(f(n))$ time randomized algorithm which outputs $LCP(P,Q)$ with probability at least $q$, where $0 < p < 1$ and $0 < q < 1$ are any fixed constants.
**Proof:** First note that the success probability for LCPD can be increased to $1-(1-p)^{b \log\log n}$ by repeating the algorithm $\lceil b \log\log n \rceil$ times. Then we apply binary search to solve LCP. Choosing an appropriate constant $b$, the success probability becomes $(1-(1-p)^{b\log\log n})^{\log n} \geq q$ for sufficiently large $n$. □

Next, we describe propositions useful for developing a randomized algorithm for LCPD.

**Proposition 2:** Let $S$ be a set with $n$ elements, and $S'$ be a subset of $S$ with $K$ elements. If $U$ is a multi-set with $\lceil \frac{n}{K} \ln \frac{1}{1-q} \rceil$ elements randomly drawn from $S$, $|U \cap S'| \geq 1$ holds with probability at least $q$.
**Proof:** If $|U| = t$, the probability that $U$ contains no element in $S'$ is $(1 - \frac{K}{n})^t$. To satisfy $1 - (1 - \frac{K}{n})^t \geq q$, it is sufficient that $t \geq \frac{n}{K} \ln \frac{1}{1-q}$ holds since

$$\ln(1-q) \geq \ln(1-\frac{K}{n})^t = -t(\frac{K}{n} + \frac{1}{2}(\frac{K}{n})^2 + \frac{1}{3}(\frac{K}{n})^3 + \cdots).$$ □

**Proposition 3:** Let $S$ be a set with $n$ elements and $S'$ be a subset of $S$ with $K$ elements. If $U$ is a multi-set with $2\lceil \frac{n}{K} \ln \frac{1}{1-q} \rceil$ elements randomly drawn from $S$, $|U \cap S'| \geq 2$ holds with probability at least $(1 - \frac{1}{K})q^2$.

From these propositions, we need not test all combinations of two elements from each point set. If we test all combinations between all pairs from $U \subseteq P$ and all pairs from $Q$ where $|U| = O(\frac{n}{K})$, we can find a common subset with $K$ elements with high probability if $|LCP(P,Q)| \geq K$. This yields an $\tilde{O}(\frac{n^5}{K^2})$ time randomized algorithm.

## 3.2 An Improved Algorithm

When $K$ is large, the above randomized algorithm is a considerable improvement of the naive algorithm. But, when $K$ is small, it still requires $\tilde{O}(n^5)$ time. To develop an $o(n^5)$ time algorithm, we make some improvement.

Let $T_{\boldsymbol{xy}}$ denote an isometric transformation such that $T_{\boldsymbol{xy}}(\boldsymbol{x}) = \mathbf{o}$ and $\overrightarrow{\mathbf{o}T_{\boldsymbol{xy}}(\boldsymbol{y})}$ is parallel to $x$-axis, where $\mathbf{o}$ denotes the zero vector. Then, procedure $LCPD(P,Q,K)$ describes the randomized algorithm for LCPD, where the followings are associated with each $B_k$ (see Fig. 1). $dist(B_k)$ denotes the distance between the points of each pair in $B_k$. $set(B_k)$ denotes a set of points $\{T_{\boldsymbol{q}_{j_1}\boldsymbol{q}_{j_2}}(\boldsymbol{q}_j) \mid \boldsymbol{q}_j \in Q, (\boldsymbol{q}_{j_1}, \boldsymbol{q}_{j_2}) \in B_k\}$. For each $\boldsymbol{q} \in set(B_k)$, $list(\boldsymbol{q})$ denotes a subset $\{(\boldsymbol{q}_{j_1}, \boldsymbol{q}_{j_2}) \mid (\exists \boldsymbol{q}' \in Q)(T_{\boldsymbol{q}_{j_1}\boldsymbol{q}_{j_2}}(\boldsymbol{q}') = \boldsymbol{q})\}$ of $B_k$. $B_k[1..|B_k|]$ is an array of integer, where each index is associated with an element (a pair of points) in $B_k$.

**Procedure** $LCPD(P, Q, K)$
  Randomly choose a multi-set $U \subseteq P$ such that $|U| = O(\frac{m}{K})$;
  Let $B$ be the set of all pairs of distinct points in $Q$;
  Partition $B$ into $B_1, \cdots, B_h$ so that each set consists of the pairs of the same distance;
  **for all** pairs $(\boldsymbol{p}_{i_1}, \boldsymbol{p}_{i_2})$ of distinct points in $U$ **do**
    Find $B_k$ such that $dist(B_k) = |\overline{\boldsymbol{p}_{i_1}\boldsymbol{p}_{i_2}}|$ ;
    Initialize each element of an array $B_k[1..|B_k|]$ to 0;
    **for all** $\boldsymbol{p} \in T_{\boldsymbol{p}_{i_1}\boldsymbol{p}_{i_2}}(P)$ **do**
      **if** there is $\boldsymbol{q} \in set(B_k)$ having the same coordinates as $\boldsymbol{p}$ **then**
        **for all** $s \in list(\boldsymbol{q})$ **do** Increment $B_k[s]$;
        **if** $B_k[s] \geq K$ for some $s$ **then**
          Output a common point set with $K$ elements; **halt**;
  Output 'No';

**Theorem:** Procedure $LCPD(P, Q, K)$ solves LCPD in $\tilde{O}(\frac{n^4}{K})$ time with high probability.
**Proof:** First, note that the total size of $set(B_k)$'s is $O(n^3)$ and they can be computed and preprocessed in $\tilde{O}(n^3)$ time.

For each element $s \in B_k$, $B_k[s]$ is incremented no more than $K$ times. Otherwise, a common point set would be output and the procedure would halt. Since $|B_k|$ is at most $O(n^2)$ for each $B_k$, the number of increments does not exceed $Kn^2$ for each pair $(\boldsymbol{p}_{i_1}, \boldsymbol{p}_{i_2})$. Since $O((\frac{m}{K})^2)$ pairs are tested, the total computation time is $\tilde{O}(n^3 + (\frac{m}{K})^2 Kn^2) = \tilde{O}(\frac{n^4}{K})$. □

From Proposition 1 and the fact that $\frac{n^4}{K}$ decreases if $K$ increases, we can see that $LCP$ can also be solved in $\tilde{O}(\frac{n^4}{K})$ time with high probability.

**Corollary:** LCP can be solved in $\tilde{O}(\frac{n^4}{K})$ time with high probability.

The time complexity can be reduced combining with the results in combinatorial geometry. For example, it is known that the number of unit distances determined by $n$ points in the plane is $O(n^{\frac{4}{3}})$ [4]. Using this, we can prove that the time complexity of $LCPD(P, Q, K)$ is bounded by $\tilde{O}(n^3 + (\frac{m}{K})^2 Kn^{\frac{4}{3}}) = \tilde{O}(n^3 + \frac{n^{(10/3)}}{K})$. Although the time complexity of the deterministic algorithm can be reduced using a similar discussion, it is still $\tilde{O}(Kn^{\frac{10}{3}})$. Thus, the randomized algorithm is still better than the deterministic one. Further improvement would be possible combining with further results in combinatorial geometry.

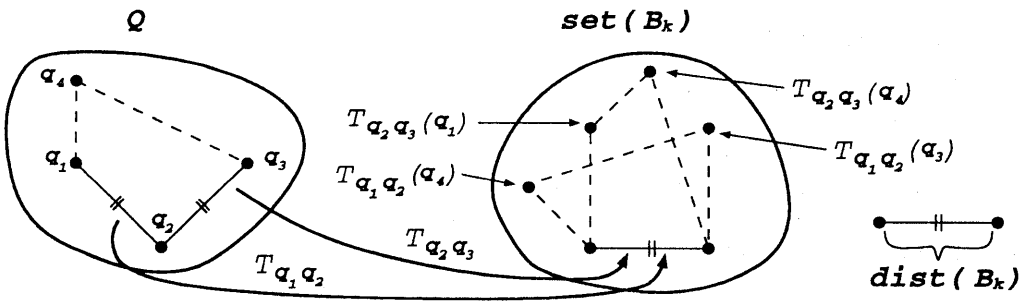


Figure 1: Construction of $set(B_k)$ in procedure $LCPD(P, Q, K)$, where $T_{q_2q_1}$ and $T_{q_3q_2}$ are omitted in this figure.

# 4 Concluding Remarks

In this paper, we have presented a randomized algorithm for computing a largest common point set of two point sets in two-dimensions. The algorithm can be generalized for higher dimensions. Moreover, the random sampling technique used in the algorithm can be applied to an approximate alignment algorithm for three-dimensional protein structures [1]. Using the technique, the time complexity can be reduced from $O(n^8)$ to $O(n^8/K^3)$, where $K$ denotes the size of an obtained alignment. Although the quality of an obtained alignment becomes slightly worse, the randomized algorithm is much more useful since $K$ is usually large. Details of implementation and the experimental result will be reported elsewhere.


## Acknowledgement

This research was partially supported by the Grant-in-Aid for Scientific Research on Priority Areas, "Genome Informatics", of the Ministry of Education, Science and Culture of Japan.


# References


[1] T. Akutsu. "Substructure search and alignment algorithms for three-dimensional protein structures," 94-AL-41, 1-8, IPSJ, 1994.

[2] T. Akutsu and M. M. Halldórsson. "On the approximation of the largest common subtrees and the largest common point sets," *Proc. ISAAC'94 (Lecture Notes in Computer Science No. 834)*, pp. 405-413, 1994.

[3] H. Alt, K. Melhorn, H. Wagener and E. Welzl. "Congruence, similarity, and symmetries of geometric objects," *Discrete & Computational Geometry*, vol. 3, pp. 237-256, 1988.

[4] K. L. Clarkson, H. Edelsbrunner, L. J. Guibas, M. Sharir and E. Welzl. "Combinatorial complexity bounds for arrangements of curves and spheres," *Discrete & Computational Geometry*, vol. 5, pp. 99-160, 1990.

[5] M. T. Goodrich, J. S. B. Mitchell and M. W. Orletsky. "Practical methods for approximate geometric pattern matching under rigid motions," *Proc. ACM Symp. Computational Geometry*, pp. 103-112, 1994.

[6] P. J. Heffernan. "Generalized approximate algorithms for point sets congruence," *Proc. Workshop on Algorithms and Data Structures*, pp. 373-384, 1993.

[7] P. J. de Rezende and D. T. Lee. "Point set pattern matching in $d$-dimensions," *Algorithmica*, vol. 13, pp. 387-404, 1995.

[8] W. R. Taylor and C. A. Orengo. "Protein structure alignment," *J. Molecular Biology*, vol. 208, pp. 1-22, 1989.

[9] G. Vriend and C. Sander. "Detection of common three-dimensional substructures in proteins," *PROTEINS: Structure, Function, and Genetics*, vol. 11, pp. 52-58, 1991.