Chenglin Fan, Jun Luo, Wencheng Wang, Farong Zhong and Binhai Zhu. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY : 1– Mon. Year

On Some Proximity Problems of Colored Sets

Chenglin Fan[1], Jun Luo[1,2], Wencheng Wang[3], Farong Zhong[4] and Binhai Zhu[5]

[1]*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China*
[2]*Huawei Noah's Ark Laboratory, Hong Kong*
[3]*State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China*
[4]*College of Math, Physics and Information Sciences, Zhejiang Normal University, Jinhua, China*
[5]*Department of Computer Science, Montana State University, Bozeman, MT 59717, USA*

E-mail: cl.fan@siat.ac.cn, jun.luo@siat.ac.cn, whn@ios.ac.cn, zfr@zjnu.cn, bhz@cs.montana.edu

Abstract The maximum diameter color-spanning set problem (MaxDCS) is defined as follows: given n points with m colors, select m points with m distinct colors such that the diameter of the set of chosen points is maximized. In this paper, we design an optimal $O(n \log n)$ time algorithm using rotating calipers for MaxDCS in the plane. Our algorithm can also be used to solve the maximum diameter problem of imprecise points modeled as polygons. We also give an optimal algorithm for the all farthest foreign neighbor problem (AFFN) in the plane, and propose algorithms to answer the farthest foreign neighbor query (FFNQ) of colored sets in two and three-dimensional space. Furthermore, we study the problem of computing the closest pair of color-spanning set (CPCS) in d-dimensional space, and remove the $\log m$ factor in the best known time bound if d is a constant.

Keywords Computational Geometry, Colored Sets, Algorithms

1 Introduction

Computing the diameter of a set of n points in a d-dimensional space ($d = 1, 2, 3 ...$) has a long history of research [1, 2]. The diameter of a point set is the maximum Euclidean distance between any two points of the set. By a reduc-

This research was supported by International Science and Technology Cooperation Program of China (Grant No. 2010DFA92720), and National Natural Science Foundation of China (NSFC) under Grants No. 11271351, 60928006, 61379087.

Corresponding author: Jun Luo, jun.luo@siat.ac.cn

tion to set disjointness, the computation requires $\Omega(n \log n)$ operations in the algebraic computation tree model [3]. The paper [4] is completely devoted to this problem and several efficient algorithms are proposed.

However, these algorithms are based on the assumption that the positions of input points are precise. If a point may randomly appear at one of the many candidate positions, which are painted with the same color, how to compute the maximum possible diameter of the point set with different colors? The problem is called the Maximum Diameter Color-spanning Set (MaxDCS) problem. The solution to the problem is useful in large computer networks. For example, a large company tries to pool resources to solve a certain computational task. But some uncertain factors interfere with the accuracy of the data and the company want to know the worst (or the best) cost based on those imprecise data.

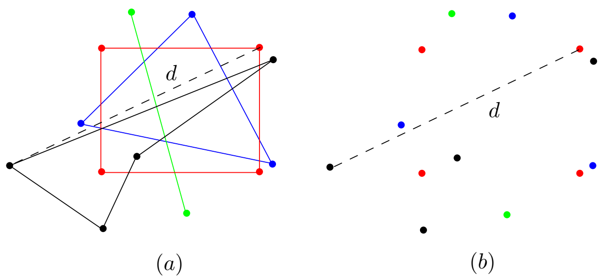


Fig. 1. Example of maximum diameter of imprecise points modeled as (a) polygons and (b) point set.

In 2007, Löffler *et al.* [5] studied the largest diameter problem and the smallest diameter problem based on imprecise data. But they used a continuous region model, such as disc and square. They showed that the largest diameter problem can be solved in $O(n \log n)$ time based on the square and the disc models. But their algorithm [5] cannot be adapted to the case when imprecise data are modeled as general polygons. The algorithm in our paper can be used to compute the maximum diameter of imprecise points in $O(n \log n)$ time, which can be modeled as polygons (see Figure 1), as the two points forming the largest diameter must be among the vertices of two polygons. On the other hand, the smallest diameter problem under the disc model is more complex and they proposed an $(1 + \epsilon)$-approximation and $O(n^{c\epsilon^{-\frac{1}{2}}})$ time algorithm for the problem, where $c \approx 6.66$ [5].

We comment that there is another model for geometric computing on uncertain data, which is called the *probabilistic* model [6–8]. In such a model, each point is assumed to appear with certain probability. But this model has little to do with the problems we investigate in the paper, so we will not go further along this direction.

The minimum diameter color-spanning set (MDCS) problem is firstly studied by Zhang *et al.* in spatial databases [9]. But Zhang *et al.* only proposed an $O(n^m)$ time algorithm for the problem. It is unfortunately a brute force algorithm. Then Chen *et al.* implemented the algorithm in a geographical tagging system [10]. Finally, Fleischer *et al.* [11] proved that the problem is NP-Complete in L_p ($p \geq 2$) metric and proposed an efficient constant factor approximation algorithm for the MDCS problem. Fan *et al.* [12] recently studied several other color-spanning problems; they gave an efficient randomized algorithm to compute a maximum diameter color-spanning set, and they showed that it is NP-hard to compute a largest closest pair color-spanning set and a planar minimum color-spanning tree.

While all-pairs nearest neighbors in any fixed dimension d can be computed in optimal $O(n \log n)$ time [13], no algorithm with similar efficiency is known for the all-pairs farthest neighbors. Agarwal *et al.* [14] showed that the three-dimensional all-pairs farthest neighbors can be computed in $O(n^{4/3} \log^{4/3} n)$ expected time, and posed closing the gap between this and the only lower bound of $\Omega(n \log n)$ as a challenging open problem. Cheong [15] *et al.* studied the all farthest pair problem when all the points are at the convex positions in R^3, and gave an expected $O(n \log^2 n)$ time algorithm to compute it.

The bichromatic closest (resp. farthest) pairs BCP (resp. BFP) [16] is formulated as follows: Given a set n red and m blue points in R^d, find a red point p and a blue point q such that the distance between p and q is minimum among all red-blue pairs, which can be calculated in $\Gamma_d(n, m)$ time and $\Gamma_d(n, n) = O(n^{1+\varepsilon})$ for $d \geq 3$ and $\Gamma_2(n, n) = O(n \log n)$.

Dumitrescu *et al.* [17] studied the BCP (resp. BFP) problems when each point is colored with one of the $m(\geq 2)$ colors, they gave algorithms to solve BCP (resp. BFP) problems in $T_d^{min}(m, n)$ (resp. $T_d^{max}(m, n)$) time in d-dimensional space. They showed that $T_2^{min}(m, n) = O(n \log n)$, $T_d^{min}(m, n) = T_d^{min}(2, n) \cdot \log m$ and $T_d^{max}(m, n) = T_d^{max}(2, n) \cdot \log m$ [17], where $T_d^{min}(2, n) = \Gamma_d(n, n)$ for $d \geq 3$. The trivial lower bound of $T_d^{min}(2, n)$ is $\Omega(n)$ and the trivial upper bound of $T_d^{min}(2, n)$ is $O(n^2)$. Ramos gave an optimal deterministic algorithm for computing the diameter of a three-dimensional point set, which can also be used to solve the bichromatic diameter in three-dimensional space in $O(n \log n)$ time [18]. Combining the above two results, we can solve BFP of m colors in $O(n \log^2 n)$ time.

In paper [19], Agarwal *et al.* gave an $O(n \log n)$ time algorithm for the following problem: Given a collection of sets with total of n points in the plane, find for each point a closest neighbor that does not belong to the same set, which is the earliest version to compute all foreign nearest pairs of points in the plane.

In this paper, all the problems we study are based on Euclidean distance. We propose an optimal time algorithm for MaxDCS. To the best of our knowledge, this is the first $O(n\log n)$ time algorithm for MaxDCS. We also give an optimal algorithm for all farthest foreign neighbors problem (AFFN) in the plane, and propose algorithms to answer the the farthest foreign neighbor query (FFNQ) of colored sets in two and three-dimensional space. At last we study the problem of computing the closest pair of color-spanning set (CPCS) in d-dimensional space, and remove the factor $\log m$ off the best known time bound if we treat d as a constant.

Table 1 lists the results of our algorithms and previous algorithms.

Table 1. An overview of the time complexity of various problems of colored sets. The number in parenthesis is the number of dimension, and m denotes the number of colors of point set.

Problem	Previous results	Our results
MaxDCS(2)	$O(n\log^2 n)$ [17, 18]	$O(n\log n)$
AFFN(2)	none	$O(n\log n)$
FFNQ(2)	none	$O(\log n)$
FFNQ(3)	none	$O(\log^2 n)$
CPCS(d)	$T_d^{min}(2,n)\log m$ [17]	$T_d^{min}(2,n)$

2 The Algorithm for MaxDCS

Problem 1. *Suppose that we are given n points with m colors. How to select m points with m different colors such that the diameter of the m selected points is maximized?*

Let CH^* be the convex hull of all n input points $\{p_1, p_2, ..., p_n\}$ and let $\langle v_1, v_2, ..., v_t\rangle$ be the vertices of CH^* in clockwise order. We can divide $\langle v_1, v_2, ..., v_t\rangle$ into r sequences $G_1, G_2, ..., G_r$ such that all vertices in one sequence have the same color and appear consecutively on CH^*. Denote the first vertex in G_i as v_i^s and the last vertex in G_i as v_i^t (in clockwise order). Let the convex hull constructed from v_{i-1}^t, v_{i+1}^s and all the vertices in G_i be CH_i, which is called the *Associate Convex Hull* of G_i. Let D_i be the set of points inside CH_i with colors different from the color of the vertices in G_i. We can construct a minimum convex chain C_i such that it starts from v_{i-1}^t, ends at v_{i+1}^s and encloses all the points of D_i. We call C_i the *Associate Chain* of G_i (see Figure 2).

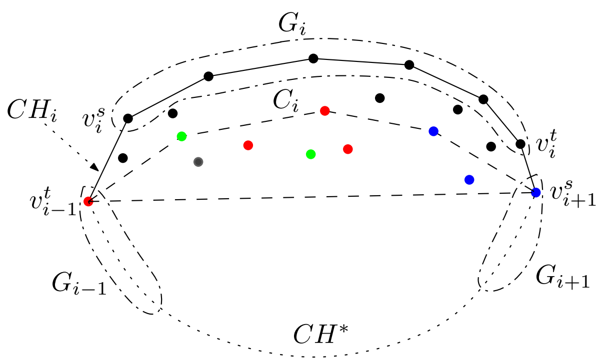


Fig. 2. Illustration of *Associate Convex Hull* CH_i and *Associate Chain* C_i.

Lemma 1. *All Associate Chains C_i $(i = 1, 2, ..., r)$ can be computed in $O(n\log n)$ time.*

Proof. First of all, we can construct CH^* in $O(n \log n)$ time. Then the vertices of CH^* are separated in r groups $G_1, G_2, ..., G_r$ by traversing the vertices of CH^* in $O(n)$ time. At the same time all CH_i's can be constructed during the traversal. For each point p_k $(k = 1, 2, ..., n)$, we can decide which CH_i it is located in $O(\log n)$ time (note that each point p_k could belong to at most two neighboring *Associate Convex Hulls*). Thus this step in total also takes $O(n \log n)$ time. For all points in CH_i, we can get rid of those points with the same color as those in G_i to obtain D_i which can be done in linear time. Since one point can only belong to at most two neighboring *Associate Convex Hulls*, we have $\sum_{i=1}^{r} |D_i| = O(n)$. Therefore, we can construct all *Associate Chains* in $O(n \log n)$ time. □

Let the two points realizing MaxDCS be p' and p''. For two points p_i and p_j, the distance between them is denoted as $d(p_i, p_j)$ and the color of p_i and p_j is denoted as $Col(p_i)$ and $Col(p_j)$ respectively. We have the following lemmas:

Lemma 2. *At least one of p' and p'' is a vertex of CH^*.*

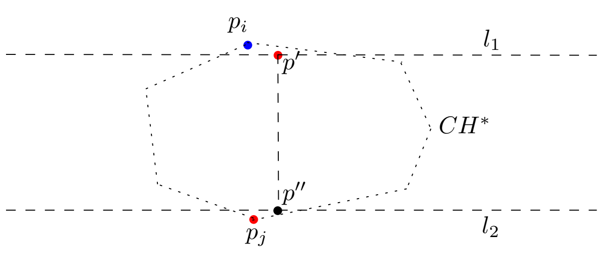


Fig. 3. At least one of the p' and p'' is a vertex of CH^*.

Proof. Assume that neither of p' and p'' is a vertex of CH^* as is shown in Figure 3. We draw two parallel lines l_1 and l_2 through p' and p'' respectively which are perpendicular to the line segment $\overline{p'p''}$. According to the property of convex hull, at least one vertex, say p_i, must lie above or on l_1 and another vertex, say p_j, must lie below or on l_2. Therefore, $d(p', p_j) \geq d(p', p'')$, $d(p'', p_i) \geq d(p', p'')$ and $d(p_i, p_j) \geq d(p', p'')$. Since $Col(p') \neq Col(p'')$ according to our assumption, $Col(p') \neq Col(p_j)$ or $Col(p'') \neq Col(p_i)$ or $Col(p_i) \neq Col(p_j)$. Then at least one pair of (p', p_j), (p'', p_i) and (p_i, p_j) is a better candidate for realizing MaxDCS than the pair of (p', p''). This is a contradiction and the lemma is proved. □

Now, without loss of generality, let p' be a vertex of CH^*.

Lemma 3. *If p'' is not a vertex of CH^*, then it must be the vertex of some* Associate Chain.

Proof. We can draw two parallel lines l_1 and l_2 through p' and p'' respectively which are perpendicular to $\overline{p'p''}$, similar to the proof of lemma 2 (see Figure 4). Since p' is a vertex of CH^*, there is no point above l_1. For the points below l_2, the color of these points must be the same as the color of p'. (Otherwise, any one of these points together with p' is a better candidate for realizing MaxDCS than the pair (p', p'').) Assume that the vertices of CH^* below l_2 are in G_i, then v_{i-1}^t and v_{i+1}^s are between l_1 and l_2. Therefore, p'' must be a vertex of C_i since all points below l_2 have the same color as p' and they do not belong to C_i. The lemma is proved. □

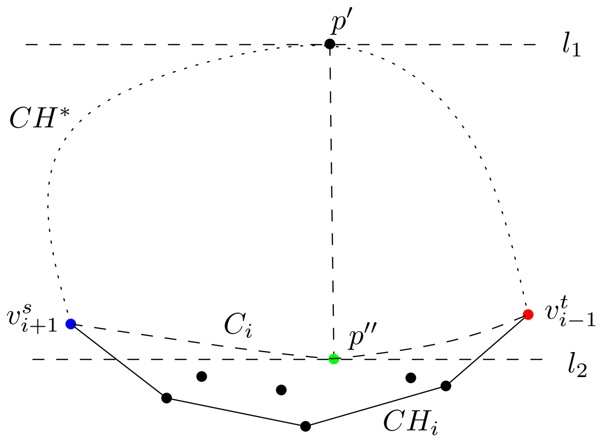


Fig. 4. p'' is the vertex of an *Associate Chain*.

According to Lemma 2 and Lemma 3, there are two cases for p' and p'':

1. p' and p'' are the vertices of CH^*. Actually, we need to find l_1 and l_2 with the maximum distance. We can use the rotating calipers method [2] to compute the diameter of a point set ((p', p'') is called an antipodal pair in the rotating calipers method). The only difference is that in our algorithm, we do not need to record the distance between an antipodal pair with the same color.

2. p' is a vertex of CH^* and p'' is a vertex of some C_i. In this case, we can rotate l_1 along CH^* and at the same time rotate l_2 along the associate chain. However, an associate chain may intersect with its neighboring associate chain and it may cause trouble when rotating l_2. Fortunately, associate chains that are not neighbors do not intersect. We can construct convex hull CH_{even} which is the convex hull of all C_i's where i is an even number (see Figure 5). Observe that all vertices of C_i (i is even) are the vertices of CH_{even}. Similarly, we can construct convex hull CH_{odd} which includes all the vertices of C_i (i is odd). Then l_2 needs to be rotated twice, once along CH_{even} and the other along CH_{odd}. For l_1, it just rotates along CH^* twice accordingly. Note that there is one minor case: when the number of the associate chains is odd, the first and the last chain of CH_{odd} intersect. We can deal with this case easily by just rotating l_2 along the odd associate chains from the first one to the last one and l_1 along CH^* accordingly as before. The only difference is that l_1 does not rotate exactly one round. It rotates a little bit more than one round but definitely no more than two rounds.

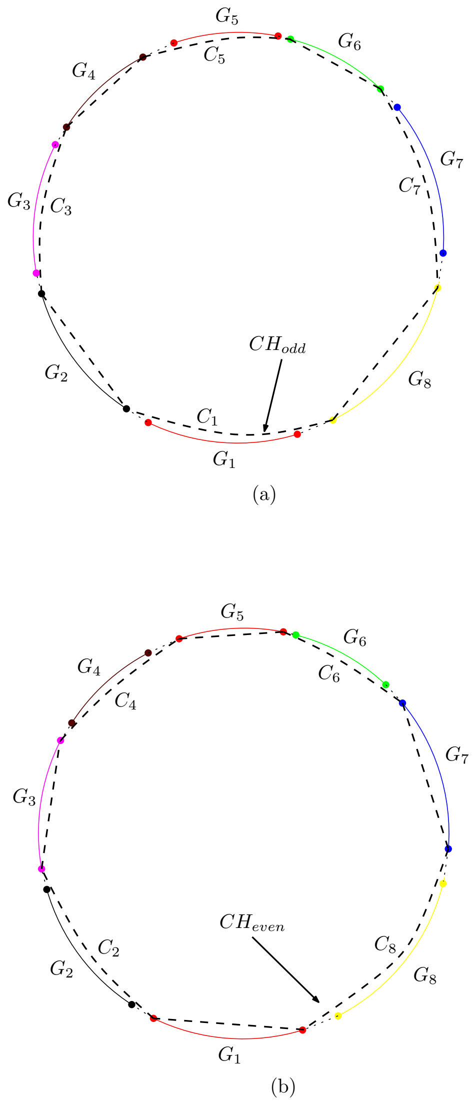


Fig. 5. Illustration of (a) CH_{odd} and (b) CH_{even}.

After three rounds of applying the rotating calipers, we can obtain p' and p'' which has the maximum distance. Each round of rotation can be finished in $O(n)$ time if CH^* and all C_i's are already computed. Thus we have the following theorem:

Theorem 1. *The maximum diameter color-spanning set problem can be solved in $O(n \log n)$ time, which is optimal.*

3 The Algorithms for AFFN and FFNQ

Problem 2. *All farthest foreign neighbors of colored set (AFFN): Given n colored points $P = \{p_1, p_2, ..., p_n\}$, for each point p_i, finding the farthest neighbor which has a color different from p_i. Let p_i^f denote the farthest colored neighbor of p_i, which satisfy $\{d(p_i, p_i^f) \geq d(p_i, p_j)$ for all p_j with the color of p_j different from the color of p_i.*

We construct the farthest-point Delaunay triangulation (the dual graph of farthest-point Voronoi diagram) of all the points P in the plane, $FPDT(P)$. The farthest-point Voronoi diagram $FPVD(P)$ can be computed in $O(n \log n)$ time. A point of P has a cell in the farthest-point Voronoi diagram if and only if it is a vertex of the convex hull of P. What is more, let $cw(p)$ and $ccw(p)$ denote the adjacent point of p in clockwise and counterclockwise direction of convex hull respectively, then the cell will come in between the cells of $cw(p)$ and $ccw(p)$ [20].

We construct the convex hull CH for the point set P, and let the point set on the convex hull be P_{CH}. We then construct the associate chains C_i and groups G_i as in Lemma 1 in $O(n \log n)$ time.

Then we divide $FPDT(P)$ into the same colored sets $F_j (1 \leq j \leq m)$. Let F'_j denote the set of points adjacent to points in F_j. Since the vertices of $FPDT(P)$ are the vertices of CH, the point set F_j is composed of several groups of G_i. All the points in each group G_i have the same color. If one point in G_i belong to F_j, then all the points in G_i belong to F_j. The cells controlled by G_i are sorted in counterclockwise direction when the points in G_i are sorted in counterclockwise direction on CH. We use CC_j to denote the set of those associate chains corresponding to F_j (each associate chain C_i in CC_j corresponds to G_i in F_j). For each associate chain C_i in CC_j, the start point and endpoint in C_i belong to F'_j.

Lemma 4. *For each point q lying in the cell controlled by point p in F_j, if q have the same color with the point p, then the farthest foreign neighbor of q belongs to set CC_j, otherwise the farthest foreign neighbor of q is p.*

Proof. If q lies in the cell controlled by p in F_j and has a color different from p, then p is q's farthest foreign neighbor. Otherwise we remove all the points in E_j where E_j is the set of all points in P with the same color as p and q ($F_j \subseteq E_j$), then the convex hull of $P \setminus E_j$ is the point set $P_{CH} \setminus F_j \cup CC_j$. For each point p' belonging to $P_{CH} \setminus F_j \setminus F'_j$, p' does not control any region of $cell(F_j)$ in $FPVD(P)$ after all the points in E_j are removed because otherwise if p' controls some regions of $cell(q)$ in $FPVD(P \setminus E_j)$, then p' controls some disjoint regions in $FPVD(P \setminus E_j)$ which contradicts the property of Voronoi diagram, that is the region controlled by each point in farthest point Voronoi diagram should be connected.

Therefore q belongs to the cell controlled by point p'' in CC_j in $FPVD(P \setminus E_j)$, and all the points in CC_j have some different colors from q. Then the farthest foreign neighbor of q belongs to set CC_j. □

Theorem 2. *The all farthest foreign neighbors of colored set problem in the plane can be solved in $O(n \log n)$ time and $O(n)$ space, which is optimal.*

Proof. The steps to compute AFFN in the plane are as follows:

1. $FPVD(P)$ and $FPDT(P)$ can be constructed in $O(n \log n)$ time [20]. $FPDT(P)$ has $O(n)$ edges. All the groups of G_i and C_i can be computed in $O(n \log n)$ time, and $\sum (|G_i| + |C_i|) = O(n)$ according to Lemma 1.

2. Find the same colored set F_j and F'_j of $FPDT(P)$, which takes $O(n)$ time as $\sum (|F_j| + |F'_j|) = O(n)$.

3. For each set CC_j, construct the farthest point Voronoi diagram of CC_j. The total time cost is $\sum |CC_j| \log(|CC_j|) = O(n \log n)$.

4. For the point p located in $cell(p')$ in $FPVD(P)$, if p' has a color different from p, then p' is the farthest foreign neighbor of p. Otherwise we locate p in $FPVD(CC_j)$

to find its farthest foreign neighbor according to Lemma 4, where the color of p is the same as the color of F_j. Hence the location for each point takes $O(\log n)$ time, and total time is $O(n \log n)$.

□

The above theorem can be easily extended to the farthest foreign neighbor query (FFNQ) of a point:

Theorem 3. *Given a colored sets P of n points in the plane and a point q (q might not be in P) with some color, the farthest foreign neighbor query (FFNQ) of q can be answered in $O(\log n)$ time in the plane with $O(n \log n)$ preprocessing time and $O(n)$ space.*

Now we consider the FFNQ problem in three dimensions. We first compute the three-dimensional convex hull $CH(3)$ in $O(n \log n)$, and let O be a vertex of $CH(3)$. Let $P_{CH(3)}$ denote the point set of $CH(3)$.

The $CH(3)$ of n points in space consists of $n_f \le 2n - 4$ faces and $n_e \le 3n - 6$ edges [20]. We assume that all the faces of $CH(3)$ are triangles (namely simplicial polytope), otherwise we just add some edges to triangulate it.

For each face f_i of $CH(3)$, we use f_i and the point O to construct a tetrahedron t_i. Then the space surround by $CH(3)$ consist of n_f tetrahedra.

For each point p inside $CH(3)$, we located it to find which tetrahedron t_i it belongs to in $O(\log^2 n)$ using $O(n \log n)$ space [21]. Let S_i denote the point set of P located in t_i.

If we remove a vertex p_j ($p_j \ne O$) from $CH(3)$, those faces with vertex p_j will disappear. Let $face_j$ denote the set of those disappeared faces, T_j denote the set of tetrahedra with one face in $face_j$, and T'_j denote those points of P inside T_j but without those points of the same color as p_j. For each face f_i, since f_i has only three vertices, f_i appears on at most three different tetrahedra sets like T_j. Let T''_j denote those point set of $\{T'_i | p_i \in F_j\}$, hence $\sum |T''_j| = O(n)$. Let FT_j denote those point set $F'_j \cup T''_j$. Lemma 4 still holds in three-dimensional space when FT_j replaces CC_j. The remaining part of the algorithm is similar to the two dimensional case. The time complexity analysis is as follows:

The complexity of $FPVD(P)$ and $FPDT(P)$ in d dimension is $O(n^{\lceil d/2 \rceil})$ [22], which can be constructed in $O(n^{\lceil d/2 \rceil} + n \log n)$ time [23]. Hence the $FPDT(P)$ in three-dimensional space can be constructed in $O(n^2)$ time and the complexity of $FPDT(P)$ is $O(n^2)$. However, the size of $FPDT(P)$ is smaller than the bound $O(n^{\lceil d/2 \rceil})$ in general case [24]. Chan *et al* [25] give an output sensitive algorithm to compute $FPDT(P)$ in three-dimensional space in $O((n + f) \log^2 f)$ time, where f ($f \in [n, n^2]$) is the size of $FPDT(P)$.

We compute the farthest-point Voronoi diagram for each FT_j. Then the total time is

$$\sum_{j=1}^{M} |FT_j|^2 \leq \sum_{k=1}^{f/n} |U'_k|^2 = O(f/n \times n^2) = O(fn)$$

where U'_k is the union of several FT_j such that $n \leq |U'_k| \leq 2n$. Since $|FT_j| \leq n$ and $\sum |FT_j| = O(f)$, then $k = O(f/n)$).

The total size of all farthest-point Voronoi diagrams is also $O(fn)$ using above analysis. In three-dimensional space, it is possible to answer point location queries in $O(\log^2 N)$ using $O(N \log N)$ space and $O(N \log N)$ preprocessing time of size N [21]. Therefore we can answer the nearest foreign neighbor query in $O(\log^2 n)$ using $O(fn \log n)$ space and $O(fn \log n)$ preprocessing time .

Therefore we have the following theorems:

Theorem 4. *Given a colored set P of n points in three dimensions and a point q (q might not be in P) with color, the farthest foreign neighbor query of q can be answered in $O(\log^2 n)$ time with $O(fn \log n)$ preprocessing time and $O(fn \log n)$ preprocessing space, where f is the size of $FPDT(P)$.*

4 The Algorithm for $CPCS(d)$

Problem 3. *The closest pair of color-spanning set in d-dimensional space (CPCS(d)): Given n input points P in d dimensions, find a pair (p, q), satisfying $\{d(p, q) \leq d(p', q'),\ Col(p) \neq Col(q), Col(p') \neq Col(q')\}$ for any p', q' in the space.*

We use the well separated pairs decomposition (WSPD) method together with a compressed quadtree to deal with this problem. Well separated pairs decomposition was defined by Callahan and Kosaraju [26]. We use the version of WSPD (very roughly) from [28].

The steps of our algorithm are as follows:

1. Construct the smallest enclosing box of P in d-dimensional space (see Figure 6). Using quadtree subdivision to divide the box into smaller boxes (child boxes) until the points in each disjoint box have the same color (see Figure 7).

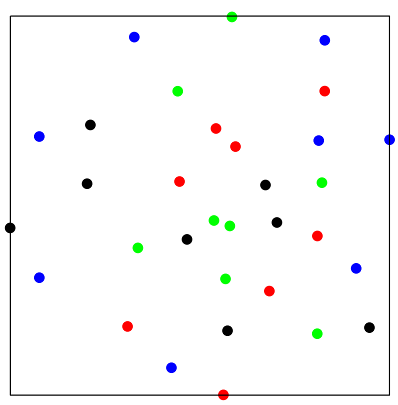


Fig. 6. The smallest enclosing box of P.

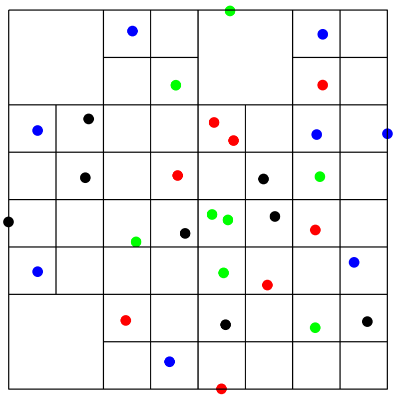


Fig. 7. The resulting subdivision after step 1 of algorithm CPCS(d).

2. Find the smallest box B of the subdivision in which there exist at least two points in B with different colors. Let d_0 denote the diameter of B and b_0 denote the edge length of the side of B. Then the distance of $CPCS(d)$ is less than or equal to d_0. Obviously, the box B is divided into child boxes following step 1 and the points in each child box have the same color which has a side length $b_0/2$ (see Figure 8).

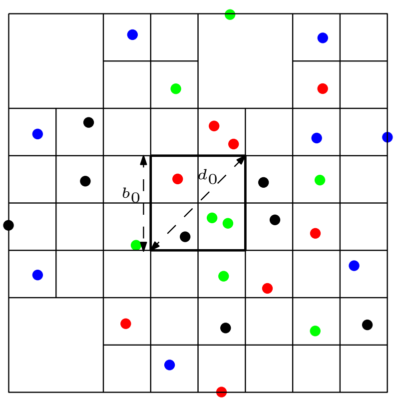


Fig. 8. Illustration of step 2 of algorithm CPCS(d).

3. For the box obtained thus far whose diameter is larger than $d_0/2$, we divide it into smaller boxes until its side length is $b_0/2$. Now the side length of any such base box is $b_0/2$ (see Figure 9).

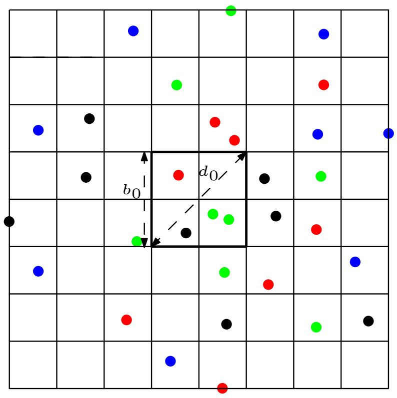


Fig. 9. Illustration of step 3 of algorithm CPCS(d).

4. For two disjoint base boxes u, v, let $dis(u, v) = min||p - q||$, where $p \in u, q \in v$. If two boxes contain points of the same color, then we can ignore them. Otherwise we compute the $dis(u, v)$ and the distance of $CPCS(d)$ is the minimum of all those $dis(u, v)$ (see Figure 10).

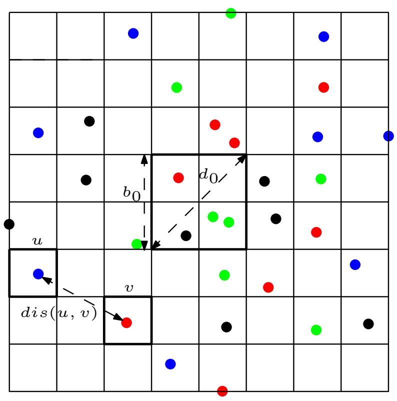


Fig. 10. Illustration of step 4 of algorithm CPCS(d).

Lemma 5. *For each base box u, there are at most $O(2^d d^{d/2})$ disjoint base boxes v satisfying*

$dis(u, v) \leq d_0$.

Proof. For any base box b, the length of the side of b is $b_0/2$ and $d_0 = b_0 * d^{1/2}$. Hence there are at most $O(d_0/(b_0/2))^d = O(2^d d^{d/2})$ disjoint base boxes whose distance from b is less than or equal to d_0. □

Theorem 5. *The time complexity for computing the distance of* $CPCS(d)$ *is* $T_d^{min}(m, n) = O(dn \log n) + O(n \log n + 2^d n) + O(2^{2d} d^d) * T_d^{min}(2, n) = T_d^{min}(2, n)$*, if* d *is a constant.*

Proof. Let the compressed quadtree after step 3 be $\mathbb{T}$ which can be constructed in $O(dn \log n)$ time [27]. Then we construct a ϵ^{-1} WSPD W for $\mathbb{T}$. Let (u, v) be a pair of boxes in W and $\epsilon = 1/2$, then there are only two possible cases:

1. $max\{diam(u), diam(v)\} > d_0/2$. Since we know u, v are ϵ^{-1} well separated, then $dis(u, v) \geq \epsilon^{-1} * max\{diam(u), diam(v)\} > \epsilon^{-1} d_0/2 = d_0$, that means we do not need to compute the distance of those pairs.

2. $diam(u) = diam(v) = d_0/2$. Those are the pairs we need to compute at step 4.

According to the Lemma 5.1 in [28], one can construct a 2-WSPD of size $O(2^d n)$ with the construction time being $O(n \log n + 2^d n)$. Of course, there is a little difference in our algorithm, as the box with diameter $d_0/2$ does not need to be divided further, but that does not affect the time complexity of our algorithm.

At step 4, let the time to compute the distance between the box pair (u, v) be $CP(|u|, |v|)$, where $|u|$ and $|v|$ denote the number of points in u and v respectively. Then $CP(|u|, |v|) = T_d^{min}(2, |u| + |v|)$ since u contains the points of one color and v contains the points of the other color. Then this problem is exactly the BCP problem. Because $T_d^{min}(2, n) = \Omega(n)$, we have $T_d^{min}(2, |x|) + T_d^{min}(2, |y|) \leq T_d^{min}(2, |x| + |y|)$. Because we only need to compute the box pair whose distance is less than or equal to d_0 and according to Lemma 5, each box appears at most $O(2^d d^{d/2})$ times in those pairs, then $\sum_{dis(u,v) \leq d_0} (|u| + |v|) = O(2^d d^{d/2} n)$. So we have $\sum_{dis(u,v) \leq d_0} CP(|u|, |v|) \leq T_d^{min}(2, \sum_{dis(u,v) \leq d_0} (|u| + |v|)) \leq T_d^{min}(2, O(2^d d^{d/2} n)) = O((2^d d^{d/2})^2) T_d^{min}(2, n) = O(2^{2d} d^d) T_d^{min}(2, n)$ as $T_d^{min}(2, n) = O(n^2)$. If we treat d as a constant, then the time to compute the distance of $CPCS(d)$ is $T_d^{min}(m, n) = = O(dn \log n) + O(n \log n + 2^d n) + O(2^{2d} d^d) * T_d^{min}(2, n) = T_d^{min}(2, n)$. □

5 Conclusions

In this paper, we propose an optimal $O(n \log n)$ time algorithm for the maximum diameter color-spanning set problem. Our algorithm can also be used to solve the maximum diameter problem of imprecise points modeled as polygons since the candidate pair of points must be vertices of two polygons, and the vertices of each polygons are painted in the same color.

We also give $O(n \log n)$ time and $O(n)$ space algorithms for AFFN problems in the plane . For the query of the farthest foreign neighbor in two dimension, we propose $O(\log n)$ query time algorithms with $O(n \log n)$ preprocessing time and $O(n)$ preprocessing space. For the three dimension query problems, we give $O(\log^2 n)$ query time algorithms with $O(fn \log n)$ preprocessing time and $O(fn \log n)$ preprocessing space, where f is the size of Farthest point Delaunay triangulation of P. We also give an algorithm to improve the best known bound of the CPCS(d) problem, and conclude that the $CPCS(d)$ of m colors can be computed in the same time with $CPCS(d)$ of two colors when d is a constant. In the future, we will focus on the problems of computing the farthest foreign pair in higher dimensional space, and approximate nearest neighbor query of color point set.

References

[1] M. I. Shamos. *Computational geometry*, Ph.D. thesis, Yale University, 1978.

[2] G. Toussaint. Solving geometric problems with the rotating calipers. *Proc. MELECON'83*, 1983.

[3] F. P. Preparata and M. I. Shamos. *Computational geometry: an introduction*, Springer-Verlag, New York, NY, USA, 1985.

[4] G. Malandain and J. Boissonnat. Computing The Diameter of a Point set. *International Journal of Computational Geometry and Applications*, 12(6):489–509, 2002.

[5] M. V. Kreveld and M. Löffler. Largest bounding box, smallest diameter, and related problems on imprecise points. *Proc. 10th Workshop on Algorithms and Data Structures (WADS'07)*, LNCS 4619, 447–458, 2007.

[6] P. Kamousi, T.M. Chan, and S. Suri. Stochastic minimum spanning trees in Euclidean spaces. *Proceedings of the 27th Annual ACM symposium on Computational Geometry*, pp. 65–74, 2011.

[7] P. K. Agarwal, A. Efrat, S. Sankararaman, and W. Zhang. Nearest-neighbor searching under uncertainty. *Proceedings of the 31st Symposium on Principles of Database Systems*, pp. 225–236, 2012.

[8] S. Suri, K. Verbeek, and H. Yildiz. On the most likely convex hull of uncertain points. *Proceedings of ESA'2013*, LNCS 8125, pp. 791-802, 2013.

[9] D. Zhang, Y. M. Chee, A. Mondal, A.K.H. Tung and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. *Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE'09)*, 688–699, 2009.

[10] Y. Chen, S. Shen, Y. Gu, M. Hui, F. Li, C. Liu, L. Liu, B.C. Ooi, X. Yang, D. Zhang and Y. Zhou. MarcoPolo: A community system for sharing and integrating travel information on maps. *Proceedings of the 12th International Conference on Extending Database Technology (EDBT'09)*, 1148–1151, 2009.

[11] R. Fleischer and X. Xu. Computing Minimum Diameter Color-Spanning Sets. *Proceedings of the 4th International Workshop on Frontiers in Algorithmics (FAW'10)*, LNCS 6213, 285–292, 2010.

[12] C. Fan, W. Ju, J. Luo and B. Zhu. On Some Geometric Problems of Color-Spanning Sets. *Journal of Combinatorial Optimization*, 26(2):266–283, 2013.

[13] P. M. Vaidya. An $O(n \log n)$ algorithm for the all-nearest-neighbors problem. *Discrete Comput. Geom.*, 4:101–115, 1989.

[14] P. K. Agarwal, J. Matousek and S. Suri. Farthest Neighbors, Maximum Spanning Trees and Related Problems in Higher Dimensions. *Comput. Geom. Theory Appl.*, 1(4):189–201, 1992.

[15] O. Cheong, C. S. Shin, and A. Vigneron. Computing farthest neighbors on a convex polytope. *Theor. Comput. Sci.*, 296(1):47–58, 2003.

[16] P. K. Agarwal, H. Edelsbrunner, O. Schwarzkopf and E. Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. *Proceedings of the sixth annual symposium on Computational geometry*, 203–210, 1990.

[17] A. Dumitrescu and S. Guha. Extreme Distances in Multicolored Point Sets. *Proc. Int. Conf. Computational Science (ICCS'02)*, PartIII, LNCS2331, 14–25, 2002.

[18] E. A. Ramos. An Optimal Deterministic Algorithm for Computing the Diameter of a Three-Dimensional Point Set. *Discrete Comput. Geom.*, 26:233–244, 2001.

[19] A. Aggarwal, H. Edelsbrunner, P. Raghavan and P. Tiwari. Optimal Time Bounds for Some Proximity Problems in the Plane. *Information Processing Letters*, 42:55–60. 1992.

[20] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*, third edition, Springer-Verlag, 2008.

[21] J. Snoeyink. Point location. *Handbook of Discrete and Computational Geometry*, J.E.Goodman and J.O'Rourke, editors, 2nd edn., chapter 34, 2004.

[22] V. Klee. On the complexity of d-dimensional Voronoi diagrams. *Archiv der Mathematik*, 34: 75–80, 1980.

[23] B. Chazelle. An optimal convex hull algorithm and new results on cuttings. *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, 29–38, 1991.

[24] R. A. Dwyer. Higher-dimensional voronoi diagrams in linear expected time. *Discrete Comput. Geom.*, 6(1):343–367, 1991.

[25] T. M. Chan, J. Snoeyink and Chee-Keng Yap. Primal Dividing and Dual Pruning: Output-Sensitive Construction of Four-Dimensional Polytopes and Three-Dimensional Voronoi Diagrams. *Discrete Comput. Geom.*, 18(4):433–454, 1997.

[26] P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. Assoc. Comput. Mach.*, 42:67–90, 1995.

[27] S. Har-Peled. *Geometric Approximation Algorithms*, American Mathematical Society, 2011.

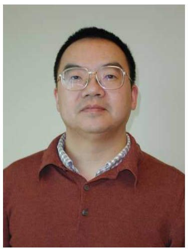
[28] S. Har-Peled and M. Mendel. Fast construction of nets in low dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006.

Chenglin Fan Chenglin Fan is currently an assistant researcher in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. He received his MS degree in computer science and technology from Central South University, Changsha, in 2011. His research interests are in algorithm design and analysis, and computational geometry. He has published over 10 journal and conference papers in these areas.

Jun Luo Jun Luo received his PhD in computer science from the University of Texas at Dallas, USA, in 2006. Then he spent two years as postdoc at Utrecht University, the Netherlands. He is currently a researcher in Huawei Noah's Ark Laboratory in Hong Kong. Before that he was an associate professor in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests are in algorithm design and analysis, computational geometry, GIS and data mining. He has published over 70 journal and conference papers in these areas.

Wencheng Wang Wencheng Wang is a professor in the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, where he leads a research group on Computer Graphics and Image Processing. He received his PhD degree in software from the Institute of Software, Chinese Academy of Sciences, in 1998, and the Excellent PhD Dissertation Award from the Degree Committee of State Council of China and Ministry of Education of China in 2001. His research interests include computational geometry, visualization, virtual reality and image editing. He is a member of ACM and IEEE.

Farong Zhong Farong Zhong is a professor of computer science at Zhejiang Normal University. He received his PhD degree in computer science at Shanghai Jiaotong University in 2005. His research interests include BDD-based network reliability analysis, and algorithms. He has published over 30 journal and conference papers in these areas.



Binhai Zhu Binhai Zhu is currently a professor in computer science at Montana State University, USA. He obtained his PhD in Computer Science from McGill University, Canada, in 1994. He was a post-doctoral 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. 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 150 papers in these areas. He is a member of ACM.