www.toc.ui.ac.ir

Transactions on Combinatorics
ISSN (print): 2251-8657, ISSN (on-line): 2251-8665
Vol. x No. x (201x), pp. xx-xx.
© 2022 University of Isfahan

www.ui.ac.ir

APPROXIMATE k-NEAREST NEIGHBOR GRAPH ON MOVING POINTS

ZAHED RAHMATI

Abstract. In this paper, we introduce an approximation for the k-nearest neighbor graph (k-NNG) on a point set P in $\mathbb{R}^d$. For any given $\varepsilon > 0$, we construct a graph, that we call the *approximate k-NNG*, where the edge with the ith smallest length incident to a point p in this graph is within a factor of $(1+\varepsilon)$ of the length of the edge with the ith smallest length incident to p in the k-NNG.

For a set P of n moving points in $\mathbb{R}^d$, where the trajectory of each point $p \in P$ is given by d polynomial functions of constant bounded degree, where each function gives one of the d coordinates of p, we compute the number of combinatorial changes to the approximate k-NNG, and provide a kinetic data structure (KDS) for maintenance of the edges of the approximate k-NNG over time. Our KDS processes $O(kn^2 \log^{d+1} n)$ events, each in time $O(\log^{d+1} n)$.

1. Introduction

Let P be a set of n points in $\mathbb{R}^d$, $d = O(1)$, and let $q_i \in P$ be the ith nearest neighbor of $p \in P$. The *k-Nearest Neighbor Graph* (k-NNG for short) $G(P, E)$ is a graph whose edge set is $E = \{(p, q_i) : p \in P, i \le k\}$. Given any $\varepsilon > 0$, we call the graph $G(P, \hat{E})$ *the approximate k-NNG* whose edge set is $\hat{E} = \{(p, \hat{q}_i) : p \in P, i \le k\}$, where $\hat{q}_i$ is a point in P such that $|p\hat{q}_i| \le (1+\varepsilon) \cdot |pq_i|$; here, $|p\hat{q}_i|$ denotes the Euclidean length of the edge $(p, \hat{q}_i)$, and we call $\hat{q}_i$ some ε-approximation for q_i, or ε-approximate ith nearest neighbor to p.

In this paper, we consider a kinetic version of the approximate k-NNG, where the trajectory of each point p in d-dimensional space is given by d known polynomial functions of constant bounded degree, where each function gives one of the d coordinates of the point p. The *kinetic approximate k-NNG*

Communicated by Dariush Kiani.

MSC(2010): Primary: 05C85; Secondary: 68R05, 68W40.

Keywords: Approximation, k-Nearest Neighbor Graph, Moving Points.

Article Type: Research Paper.

Received: 04 December 2021, Accepted: 19 February 2022.

http://dx.doi.org/10.22108/TOC.2022.130533.1943

.

problem is to maintain (track and update) the edges in $\hat{E}$ over time. Here, we assume that there is a model of computation which solves any polynomial equation exactly in constant time.

Tracking the attributes of moving objects in computational geometry has been studied extensively over the over the last two decades; see [1, 2, 3, 4, 5, 6] and references therein. To maintain an attribute (*e.g.*, approximate k-NNG) we design a set of algorithms and data structures, namely the *kinetic data structure* (KDS) [9]; we update the KDS over time whenever some event occurs. Some events may change the desired attribute that are called *external events*, but some events do not change the attribute of interest that are called *internal events*, which only change some internal data structures. The *efficiency* of a KDS is the ratio of the worst-case number of internal events in the KDS to the worst-case number of external events, and would be nice to keep this criterion polylogarithmic. Another important KDS performance criterion is the *responsiveness*, which is the processing time to handle and event, and would like to have it polylogarithmic.

All the nearest neighbors of the points can be reported in time $O(n \log n)$ [10]. For any $k \geq 1$, all the k-nearest neighbors can be reported in time $O(kn \log n)$ [11], in order of increasing distance; reporting the unordered set takes time $O(n \log n + kn)$ [11, 12, 13].

Basch *et al.* [9] provided a KDS for maintenance of the closest pair in $\mathbb{R}^2$, where the trajectory of each point is given by two polynomial functions of degree bounded by constant s. Their KDS uses $O(n)$ space and processes $O(n^2\beta_{2s+2}(n)\log n)$ events, each in time $O(\log^2 n)$. Here, $\beta_s(n)$ is an extremely slow-growing function. For higher dimensional space, in $\mathbb{R}^d$, Basch *et al.* [14] used range trees to maintain the closest pair; their KDS uses $O(n\log^{d-1} n)$ space and processes $O(n^2\beta_{2s+2}(n)\log n)$ events, each in worst-case time $O(\log^d n)$. Then, Agarwal *et al.* [15] provided data structures to maintain the closest pair and all the nearest neighbors in $\mathbb{R}^d$. The closest pair KDS by Agarwal *et al.* uses $O(n\log^{d-1} n)$ space and processes $O(n^2\beta_{2s+2}(n)\log n)$ events, each in amortized time $O(\log^d n)$. Agarwal *et al.* gave the first KDS to maintain all nearest neighbors in $\mathbb{R}^d$. Their KDS uses $O(n\log^d n)$ space and processes $O(n^2\beta^2_{2s+2}(n)\log^{d+1} n)$ events with the total expected time $O(n^2\beta^2_{2s+2}(n)\log^{d+2} n)$ for processing all the events. Rahmati *et al.* [8] provided the first KDS to maintain the Semi-Yao graph (*i.e.*, theta graph) in $\mathbb{R}^2$. Their Semi-Yao graph KDS uses $O(n)$ space and processes $O(n^2\beta_{2s+2}(n))$ events with total processing time $O(n^2\beta_{2s+2}(n)\log n)$. Their all nearest neighbors KDS, which improved the previous KDS by Agarwal *et al.*, uses $O(n)$ space and processes $O(n^2\beta^2_{2s+2}(n)\log n)$ events with total processing time $O(n^2\beta^2_{2s+2}(n)\log^2 n)$.

For linearly moving points in fixed dimension d, Chan and Rahmati [16] gave a method for approximating the minimum closest pair distance and nearest neighbor distances. For any constant $\epsilon > 0$, their algorithm computes a $(1+\epsilon)$-factor approximation to the minimum closest pair distance in time $\tilde{O}(n^{5/3})$. Assuming $n \leq m \leq n^5$, they build a data structure, which uses $\tilde{O}(m)$ preprocessing time and space, to answer queries. For any given linearly moving query point q in any time, their algorithm computes a $(1+\epsilon)$-factor approximation of the minimum nearest neighbor distance to q in time $\tilde{O}(\frac{n}{m^{1/5}})$. Here, the notation $\tilde{O}$ hides polylogarithmic factors in the complexity.

Rahmati *et al.* [7, 17] presented the first KDS for maintenance of all the k-nearest neighbors in $\mathbb{R}^d$, for any $k \geq 1$, whereas the previous works considered the $k = 1$ case only. Their KDS for maintenance of all the 1-nearest neighbors in $\mathbb{R}^d$ uses $O(n \log^d n)$ space and processes $O(n^2\beta^2_{2s+2}(n)\log n))$ events with the total processing time $O(n^2\beta_{2s+2}(n)\log^{d+1} n)$. For any $k \geq 1$, their KDS for maintenance of all the k-nearest neighbors handles $O(\phi(n)n^2)$ events, each in time $O(\log n)$ in an amortized sense. Here, $\phi(n)$ is the complexity of the k-level of a set of n partially-defined polynomial functions, such that each pair of them intersects at most s times, *i.e.* [17],

$$\phi(n) = \begin{cases} O(n^{3/2}\log n), & \text{for } s = 2; \\ O(n^{5/3}\text{poly}\log n), & \text{for } s = 3; \\ O(n^{31/18}\text{poly}\log n), & \text{for } s = 4; \\ O(n^{161/90-\delta}), & \text{for } s = 5, \text{ for some constant } \delta > 0; \\ O(n^{2-1/2s-\delta_s}), & \text{for odd } s, \text{ for some constant } \delta_s > 0; \\ O(n^{2-1/2(s-1)-\delta_s}), & \text{for even } s, \text{ for some constant } \delta_s > 0. \end{cases}$$

A bound $f(n)$ of $\phi(n)$ can be expressed to the k-sensitive bound $O(f(k)(n/k)\beta(n/k))$. The complexity of the $(\leq k)$-level is $O(kn\beta(n/k))$ [22, 23, 24, 25], where $\beta(n)$ is an extremely slow-growing function. They also provided a KDS for all approximate nearest neighbors, which uses $O(n\log^d n)$ space and processes $O(n^2\log^d n)$ events, each in worst-case time $O(\log^d n \log\log n)$.

In this paper, we provide a KDS for maintenance of the edges of the approximate k-NNG on a set P of moving points, where the trajectory of each point $p \in P$ is given by d polynomial functions of constant bounded degree. The number of events in our approximate k-NNG KDS is $O(kn^2\log^{d+1} n)$, and each event can be handled in worst-case time $O(\log^{d+1} n)$.

In Section 2 we present the preliminaries and the required lemmas we use in our proofs in next sections. Section 3 shows how to construct the approximate k-NNG on a set of stationary points; we present $O(kn\log^{d+1} n)$ time algorithm for computing the approximate k-NNG. We give our main results in Section 4.

2. Preliminaries

The d-dimensional space around the origin can be covered by a collection of $O(1/\theta^{d-1})$ interior-disjoint polyhedral cones of opening angle θ [18]. Denote by C the set of all such cones C_l, $1 \leq l \leq O(1/\theta^{d-1})$, around the origin. Let $C_l(p)$ be a translation of C_l with apex at p. Let x_l denote an axis inside C_l with the same origin as of C_l.

Lemma 2.1. [17, Lemma 3.1] *Let q_i be the ith nearest neighbor of p among a set P of points in $\mathbb{R}^d$, and let $C_l(q_i)$ be the cone of q_i that contains p such that its opening angle $\theta \leq \pi/3$. Then the point p is among the first i points in $P \cap C_l(q_i)$ having the smallest projection on x_l-axis.*

Lemma 2.2. [18, Lemma 2.1.] *Let C be a collection of polyhedral cones of of opening angle θ, where $\cos 2\theta - \sin 2\theta \geq 1/(1+\varepsilon)$. Let q and s be two points in $C_l(p)$ such that the x_l-coordinate of s is less than or equal to the x_l-coordinate of q. Then $|ps| + (1+\varepsilon) \cdot |sq| \leq (1+\varepsilon)|pq|$.*

Consider a sequence of n points where the y-coordinate of each points is a polynomial function of constant bounded degree. One can build a *kinetic* sorted list on the points to track the order of the points based on their y-coordinates over time. The kinetic sorted list on a set of n moving points processes $O(n^2)$ events, each in time $O(\log n)$. The following lemma gives the properties of a dynamic version of a kinetic sorted list.

Lemma 2.3. [17, Theorem 2.5.] *Given a sequence of m insertions and deletions into a kinetic sorted list whose maximum size at any time is n. The kinetic sorted list generates $O(mn)$ events. Each update/event can be handled in $O(\log n)$ time. A kinetic sorted list on n elements can be constructed in $O(n \log n)$ time.*

The d-dimensional range trees can be used to efficiently report the points inside any polyhedral cone $C_l(p)$ [19]. For a set of n moving points, the following introduces a variant of the range trees, so called *rank-based range tree* (RBRT), that can be used to efficiently report the points inside any polyhedral cone over time.

Lemma 2.4. [18, Lemma 2.7.] *A d-dimensional rank-based range tree (RBRT) uses $O(n \log^d n)$ storage and can be constructed in $O(n \log^d n)$ time. It can be described as a set of pairs $\Psi = \{(B_1, R_1), ..., (B_m, R_m)\}$ with the following properties.*

- *For any two points p and q in P where $q \in C(p)$, there is a unique pair $(B_j, R_j) \in \Psi$ such that $p \in B_j$ and $q \in R_j$.*
- *For any pair $(B_j, R_j) \in \Psi$, if $p \in B_j$ and $q \in R_j$, then $q \in C(p)$ and $p \in \bar{C}(q)$.*
- *Each point $p \in P$ is in $O(\log^d n)$ pairs of (B_j, R_j).*
- *For any point $p \in P$, all the sets B_j (resp. R_j) where $p \in B_j$ (resp. $p \in R_j$) can be found in time $O(\log^d n)$.*
- *The set $P \cap C(p)$ (resp. $P \cap \bar{C}(p)$) of points is the union of $O(\log^d n)$ sets R_j (resp. B_j), where the subscript j is such that $p \in B_j$ (resp. $p \in R_j$).*

For a set of n moving points, where the trajectories are polynomial functions of constant bounded degree, the RBRT can be maintained by processing $O(n^2)$ events, each in worst-case time $O(\log^d n)$.

3. Computing Approximate k-NNG

Here, we give an efficient algorithm for computing the approximate k-NNG on a set P of n stationary points in $\mathbb{R}^d$. Fix a polyhedral cone $C_l \in C$ of opening angle θ and consider its corresponding normal vectors $u_1, \ldots, u_d$, and its corresponding axis x_l.

Let $\Psi_l = \{(B_1, R1), \ldots, (B_m, R_m)\}$ be a set of pairs obtained from a RBRT corresponding to C_l (see Lemma 2.4). Figure 1 depicts a polyhedral cone C_l in the plane and some $(B_j, Rj) \in \Psi_l$. Let

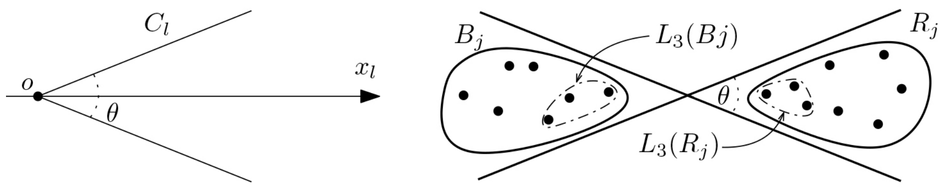

FIGURE 1. The cone C_l and a member (B_j, R_j) of Ψ_l in $\mathbb{R}^2$.

$L_k(B_j)$ (resp. $L_k(R_j)$) be the first k points in B_j (resp. R_j) having the maximum (resp. minimum) x_l-coordinates; Figure 1 shows the set of points in $L_k(B_j)$ and $L_k(R_j)$ for $k = 3$. For any point $b_j \in L_k(B_j)$ and any point $r_j \in L_k(R_j)$, we create an edge (b_j, r_j). Let E_l denote the set of such edges, for all $j \in \{1, \dots, m\}$. We call the graph $G(P, \cup_l E_l)$ the *BR graph*. The following lemma gives the properties of the BR graph.

Lemma 3.1. *The BR graph $G(P, \cup_l E_l)$ on a set P of n points can be constructed in time $O(kn \log^{d+1} n)$, and the degree of each point in the BR graph is $O(k \log^{d+1} n)$.*

Proof. Corresponding to each C_l, we need to build a $(d+1)$-dimensional RBRT, where the first d-levels of the RBRT are for keeping the points sorted based on their u_i-coordinates, $i \in \{1, \dots, d\}$, and the last level of the RBRT is for having the points sorted based on their x_l-coordinates. From Lemma 2.2, the RBRT can be built in time $O(n \log^{d+1} n)$, and each point $p \in P$ is in $O(\log^{d+1} n)$ pairs of (B_j, R_j). If p is in $L_k(B_j)$ (resp. $L_k(R_j)$), we create at most k edges with all the points in $L_k(R_j)$ (resp. $L_k(B_j)$), adding to E_l, which implies that the number of all edges incident to each point p in E_l is $O(k \log^{d+1} n)$. Therefore the degree of each point in the BR graph $G(P, \cup_l E_l)$ is $O(k \log^{d+1} n)$ and the BR graph can be constructed in time $O(kn \log^{d+1} n)$. $\square$

Lemma 3.2. *The approximate k-NNG is a subgraph of the BR graph.*

Proof. We show that among all the edges incident to each point $p \in P$ in BR graph, there are k edges $(p, \hat{q}_i)$, $1 \le i \le k$, such that $\hat{q}_i$ is some ε-approximate ith nearest neighbor to p.

Let $q_k \in P$ be the kth nearest neighbor of $p \in P$. Assume $p \in \bar{C}_l(q_k)$. By Lemma 2.4, for any two points p and q_k in P, there exists a unique pair $(B_j, R_j) \in \Psi_l$ such that $p \in B_j$ and $q_k \in R_j$; here $\bar{C}_l(\cdot)$ is the reflection of $C_l(\cdot)$. By Lemma 2.1, and from the fact that $B_j \subseteq \bar{C}_l(q_k)$, p is among the first k points having the maximum x_l-coordinates among the points in B_j, *i.e.*, $p \in L_k(B_j)$.

Now we show that some point in $L_k(R_j)$ gives some ε-approximate kth nearest neighbor to p. Note that $q_k \in C_l(p)$. By Lemma 2.2, any point in $C_l(p)$ with x_l-coordinate less than or equal to the x_l-coordinate of q_k is some ε-approximate kth nearest neighbor to p. Since $q_k \in R_j$ and $R_j \subseteq C_l(p)$, it is easy to see that there is a point in $L_k(R_j)$ as some ε-approximate kth nearest neighbor to p. $\square$

Theorem 3.3. *The approximate k-NNG can be constructed in time $O(kn \log^{d+1} n)$.*

Proof. The BR graph can be constructed in time $O(kn\log^{d+1} n)$ and the degree of each point p in the BR graph is $O(k\log^{d+1} n)$ (by Lemma 3.1). Since BR graph is a supergraph of the approximate k-NNG (by Lemma 3.2), by tracing the edges incident to each point p in BR graph and picking the first k edges with smallest Euclidean lengths, we can find k points as some ε-approximate k-nearest neighbors to p; this can be done in time $O(k\log^{d+1} n)$, since the kth smallest element can be found in linear time. For all points it takes $O(kn\log^{d+1} n)$ time. □

4. Kinetic Approximate k-NNG

In this section, we assume the trajectory of each point p is given by d polynomial functions of constant bounded degree, where each function gives one of the d coordinates of p. Next, we provide a KDS for maintenance of the edges of the approximate k-NNG on the moving points.

Corresponding to each C_l, we need the points of each $(B_j, R_j) \in \Psi_l$ sorted in ascending order of their x_l-coordinates. Thus we build a $(d+1)$-dimensional RBRT T_l on the points, where the last level of T_l is built based on ascending order of their x_l-coordinates; T_l gives the sorted lists $L(B_j)$ and $L(R_j)$ of the points in B_j and R_j, respectively, in ascending order of the their x_l-coordinates.

To track the required changes into the RBRT T_l, we build $d+1$ kinetic sorted lists $L(u_1), \ldots, L(u_d)$ and $L(x_l)$ based on the order of the points along axes $u_1, \ldots, u_d$ and x_l, respectively. Note that the lists $L(B_j)$ and $L(R_j)$ remain unchanged, as long as the orders of the points along each axis u_i $(1 \le i \le d)$ and x_l remain unchanged.

Recall that each point p is in $O(\log^{d+1} n)$ pairs of $(B_j, R_j) \in \Psi_l$; consider such pairs where p is in. There are two types of edges incident to p: If $p \in B_j$, consider edges (p, r_j) for all $r_j \in L_k(R_j)$. If $p \in R_j$, consider edges (b_j, p) for all $b_j \in L_k(B_j)$. We build a kinetic sorted list $L(p)$ on the union of these two types of edges (p, r_j) and (b_j, p) based on their Euclidean lengths.

Theorem 4.1. *Given a set P of n points in $\mathbb{R}^d$, where the trajectory of each point is given by d polynomial functions of constant bounded degree, where each function gives one of the d coordinates. Our KDS for maintenance of the edges of the approximate k-NNG on P processes $O(kn^2\log^{d+1} n)$ events, each in worst-case time $O(\log^{d+1} n)$.*

Proof. Since the kinetic sorted list $L(p)$ contains the ε-approximate k-nearest neighbors to p (see the proof of Lemma 3.2), the first k edges in $L(p)$ give some ε-approximate k-nearest neighbors to p.

Let m_p be the number of insertion and deletion into $L(p)$. The maximum size of $L(p)$ at any time is $O(k\log^{d+1} n)$. From Lemma 2.3, the kinetic sorted list $L(p)$ processes $O(m_p \cdot k\log^{d+1} n)$ events, each in time $O(\log k + \log\log n)$; here, $d = O(1)$. Therefore, the number of changes to incident edges of all the points is $O(k\log^{d+1} n \cdot \sum_p m_p)$.

Since the number of all events in the RBRT T_l, $L(u_i), \ldots, L(u_d)$, and $L(x_l)$ is $O(n^2)$ (by Lemmas 2.3 and 2.4), the number of all changes into all $L(B_j)$ and $L(R_j)$ is $O(n^2)$. This implies that the number of all insertions and deletions into all the kinetic sorted lists $L(p)$, for all points $p \in P$, is $O(n^2)$;

i.e., $\sum_p m_p = O(n^2)$. Therefore, our KDS for maintenance of the edges of the approximate k-NNG processes $O(kn^2 \log^{d+1} n)$ events.

From above discussion, and the fact that the processing time of each event in the $d+1$-dimensional RBRT is $O(\log^{d+1} n)$ (by Lemma 2.4), the proof obtains. □

5. Discussion

We have presented a method for computing an approximation for the k-NNG on a set of n points in $\mathbb{R}^d$, and then provided a KDS for maintenance of the approximate k-NNG on a set P of n moving points, where the trajectory of each point is given by d polynomial functions of fixed degree.

For a set P of n moving points, the number of changes to the exact k-NNG is equal to the complexity of the $(\leq k)$-level, which is $O(kn^2\beta(n/k))$ [17]. The previous KDS for maintenance of the exact k-NNG processes $O(n^2\phi(n))$ events [17], whereas our KDS for maintenance of the edges of the approximate k-NNG processes $O(kn^2 \log^{d+1} n)$ events.

References

[1] Z. Rahmati, *Simple, faster kinetic data structures*, PhD. Thesis, University of Victoria (2014).

[2] M. Abam, *New Data Structures and Algorithms for Mobile Data*, PhD. Thesis, Eindhoven University of Technology, (2007).

[3] D. Russel, *Kinetic Data Structures in Practice*, PhD. Thesis, Stanford University, (2007).

[4] J. Basch, *Kinetic Data Structures*, PhD. Thesis, Stanford University, (1999).

[5] Z. Rahmati, V. King and S. Whitesides, Kinetic Data Structures for All Nearest Neighbors and Closest Pair in the Plane, in: *Proceedings of the 29th annual symposium on Symposuim on computational geometry (SoCG '13)*, (2013) 137–144.

[6] M. Abam, Z. Rahmati and A. Zarei, Kinetic Pie Delaunay Graph and its Applications, in: *Proceedings of the 13th Scandinavian Symposium and Workshops on Algorithms Theory (SWAT '12)*, (2012) 48–58.

[7] Z. Rahmati, M. Abam, V. King and S. Whitesides, Kinetic data structures for the Semi-Yao graph and all nearest neighbors in $\mathbb{R}^d$, in: *Proceedings of the 26th Canadian Conference on Computational Geometry (CCCG '14)*, (2014) 2–10.

[8] Z. Rahmati, M. A. Abam, V. King, S. Whitesides and A. Zarei, A simple, faster method for kinetic proximity problems, *Comput. Geom.*, **48** (2015) 342–359.

[9] J. Basch, L. J. Guibas and J. Hershberger, Data structures for mobile data, in: *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '97)*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997, 747–756.

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

[11] M. T. Dickerson and D. Eppstein, Algorithms for proximity problems in higher dimensions, *Comput. Geom*, **5** (1996) 277–291.

[12] 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** (1995) 67–90.

[13] K. L. Clarkson, Fast algorithms for the all nearest neighbors problem, in: *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS '83)*, IEEE Computer Society, Washington, DC, USA, (1983) 226–232.

[14] J. Basch, L. J. Guibas and L. Zhang, Proximity problems on moving points, in: *Proceedings of the 13th Annual Symposium on Computational Geometry (SoCG '97)*, ACM, New York, NY, USA, (1997) 344–351.

[15] P. K. Agarwal, H. Kaplan, M. Sharir, Kinetic and dynamic data structures for closest pair and all nearest neighbors, *ACM Trans. Algorithms*, **5** (2009) 37 pp.

[16] T. M. Chan and Z. Rahmati, Approximating the minimum closest pair distance and nearest neighbor distances of linearly moving points, in: *Proceedings of the 27th Canadian Conference on Computational Geometry (CCCG '15)*, (2015) 136–140.

[17] Z. Rahmati, M. Abam, V. King and S. Whitesides, Kinetic k-Semi-Yao graph and its applications, *Comput. Geom.*, **77** (2019) 10–26.

[18] M. A. Abam and M. de Berg, Kinetic spanners in $\mathbb{R}^d$, *Discrete Comput. Geom.*, **45** (2011) 723–736.

[19] M. d. Berg, O. Cheong, M. v. Kreveld and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd Edition, Springer-Verlag TELOS, Santa Clara, CA, USA, 2008.

[20] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman and A. Y. Wu, An optimal algorithm for approximate nearest neighbor searching in fixed dimensions, Journal of the ACM 45 (6) (1998) 891–923.

[21] M. Connor, P. Kumar, Fast construction of k-nearest neighbor graphs for point clouds, *IEEE Transactions on Visualization and Computer Graphics*, **16** (2010) 599–608.

[22] P. K. Agarwal, B. Aronov, T. M. Chan and M. Sharir, On levels in arrangements of lines, segments, planes, and triangles, *Discrete Comput. Geom.*, **19** (1998) 315–331.

[23] T. M. Chan, On levels in arrangements of curves, ii: A simple inequality and its consequences, *Discrete Comput. Geom.*, **34** (2005) 11–24.

[24] T. M. Chan, On levels in arrangements of curves, iii: further improvements, in: *Proceedings of the 24th annual Symposium on Computational Geometry (SoCG '08)*, ACM, New York, NY, USA, (2008) 85–93.

[25] M. Sharir, On k-sets in arrangements of curves and surfaces, *Discrete Comput. Geom.*, **6** (1991) 593–613.

Zahed Rahmati

Department of Mathematics and Computer Science, Amirkabir University of Technology, P.O.Box 15875-4413, Tehran, Iran.

Email: zrahmati@aut.ac.ir