An Output-Sensitive Approach for the L_1/L_∞ k-Nearest-Neighbor Voronoi Diagram*

Chih-Hung Liu[1], Evanthia Papadopoulou[2], and D. T. Lee[1,3]

[1] Research Center for Information Technology Innovation, Academia Sinica, Taiwan
chliu_10@citi.sinica.edu.tw dtlee@iis.sinica.edu.tw
[2] Faculty of Informatics, University of Lugano, Lugano, Switzerland
evanthia.papadopoulou@usi.ch
[3] Institute of Information Science, Academia Sinica, Taipei, Taiwan

Abstract. This paper revisits the k-nearest-neighbor (k-NN) Voronoi diagram and presents the first output-sensitive paradigm for its construction. It introduces the k-NN Delaunay graph, which corresponds to the graph theoretic dual of the k-NN Voronoi diagram, and uses it as a base to directly compute the k-NN Voronoi diagram in R^2. In the L_1, L_∞ metrics this results in $O((n+m)\log n)$ time algorithm, using segment-dragging queries, where m is the structural complexity (size) of the k-NN Voronoi diagram of n point sites in the plane. The paper also gives a tighter bound on the structural complexity of the k-NN Voronoi diagram in the L_∞ (equiv. L_1) metric, which is shown to be $O(\min\{k(n-k),(n-k)^2\})$.

1 Introduction

Given a set S of n point sites $\in R^d$ and an integer k, $1 \leq k < n$, the *k-nearest-neighbor Voronoi Diagram*, abbreviated as *k-NN Voronoi diagram* and denoted as $V_k(S)$, is a subdivision of R^d into regions, called *k-NN Voronoi regions*, each of which is the locus of points closer to a subset H of k sites, $H \subset S$, called a *k-subset*, than to any other k-subset of S, and is denoted as $V_k(H,S)$. The distance between a point p and a point set H is $d(p,H) = \max\{d(p,q), \forall q \in H\}$, where $d(p,q)$ denotes the distance between two points p and q.

A k-NN Voronoi region $V_k(H,S)$ is a polytope in R^d. The common face between two neighboring k-NN Voronoi regions, $V_k(H_1,S)$ and $V_k(H_2,S)$, is portion of the bisector $B(H_1,H_2) = \{r \mid d(r,H_1) = d(r,H_2), r \in R^d\}$. In R^2, the boundary between two neighboring k-NN Voronoi regions is a *k-NN Voronoi edge*, and the intersection point among more than two neighboring k-NN Voronoi

* This work was performed while the first and third authors visited University of Lugano in September/October 2010. It was supported in part by the University of Lugano during the visit, by the Swiss National Science Foundation under grant SNF-200021-127137, and by the National Science Council, Taiwan under grants No. NSC-98-2221-E-001-007-MY3, No. NSC-98-2221-E-001-008-MY3, and No. NSC-99-2918-I-001-009.

regions is a *k-NN Voronoi vertex*. In the L_p-metric the distance $d(s,t)$ between two points s,t is $d_p(s,t) = (|x_{1_s} - x_{1_t}|^p + |x_{2_s} - x_{2_t}|^p + \cdots + |x_{d_s} - x_{d_t}|^p)^{1/p}$ for $1 \leq p < \infty$, and $d_\infty(s,t) = \max(|x_{1_s} - x_{1_t}|, |x_{2_s} - x_{2_t}|, \ldots, |x_{d_s} - x_{d_t}|)$.

Lee [14] showed that the *structural complexity*, i.e., the size, of the k-NN Voronoi diagram in the plane is $O(k(n-k))$, and proposed an iterative algorithm to construct the diagram in $O(k^2 n \log n)$ time and $O(k^2(n-k))$ space. Agarwal et al. [3] improved the time complexity to be $O(nk^2 + n \log n)$. Based on the notions of geometric duality and arrangements, Chazelle and Edelsbrunner [8] developed two versions of an algorithm, which take $O(n^2 \log n + k(n-k)\log^2 n)$ time and $O(k(n-k))$ space, and $O(n^2 + k(n-k)\log^2 n)$ time and $O(n^2)$ space, respectively. Clarkson [9], Mulmuley [16], and Agarwal et al. [2] developed randomized algorithms, where the expected time complexity of $O(k(n-k)\log n + n\log^3 n)$ in [2] is the best. Boissonnat et al. [6] and Aurenhammer and Schwarzkopf [5] proposed on-line randomized incremental algorithms: the former in expected $O(n \log n + nk^3)$ time and $O(nk^2)$ space, and the latter in expected $O(nk^2 \log n + nk \log^3 n)$ time and $O(k(n-k))$ space. For higher dimensions, Edelsbrunner et al. [11] devised an algorithm to compute all $V_k(S)$ in R^d with the Euclidean metric for $1 \leq k < n$, within optimal $O(n^{d+1})$ time and space, and Clarkson and Shor [10] showed the total size of all $V_k(S)$ to be $O(k^{\lceil (d+1)/2 \rceil} n^{\lfloor (d+1)/2 \rfloor})$.

All the above-mentioned algorithms [2, 5, 6, 8, 9, 14, 16] focus on the Euclidean metric. However, the computationally simpler, piecewise linear, L_1 and L_∞ metrics are very well suited for practical applications. For example, L_∞ higher-order Voronoi diagrams have been shown to have several practical applications in VLSI design e.g., [17–19]. Most existing algorithms compute k-NN Voronoi diagrams using reductions to arrangements or geometric duality [2, 5, 8, 9, 16] which are not directly applicable to the L_1, L_∞ metrics. Furthermore, none of the existing deterministic algorithms is output-sensitive, i.e., their time complexity does not only depend on the actual size of the k-NN Voronoi diagram. For example, the iterative algorithm in [14] needs to generate $V_1(S), V_2(S), \ldots, V_k(S)$, and thus has a lower bound of time complexity $\Omega(nk^2)$. The algorithm in [8] needs to generate $\Theta(n^2)$ bisectors, while not all the bisectors appear in $V_k(S)$.

In this paper, we revisit the k-NN Voronoi diagram and propose the first direct output-sensitive approach to compute the L_∞ (equiv. L_1) k-NN Voronoi diagram. We first formulate the k-NN Delaunay graph (Section 2), which is the graph-theoretic dual of the k-NN Voronoi diagram. Note that the k-NN Delaunay graph is a graph-theoretic structure, different from the k-Delaunay graph in [1, 12]. We then develop a traversal-based paradigm to directly compute the k-NN Delaunay graph of point sites in the plane (Section 3). In the L_∞ metric we implement our paradigm by applying segment-dragging techniques (Section 4), resulting in an $O((m+n)\log n)$-time algorithm for the L_∞ planar k-NN Delaunay graph of size m. As a by-product, we also derive a tighter bound on the structural complexity m of the L_∞ k-NN Voronoi diagram, which is shown to be $O(\min\{k(n-k), (n-k)^2\})$. Since the L_1 metric is equivalent to L_∞ under rotation, these results are also applicable to the L_1 metric. Due to the limit of space, we remove most proofs except Theorem 2.

2 The k-Nearest-Neighbor Delaunay Graph

Given a set S of n point sites in R^d, we define the k-NN Delaunay graph following the notion of the Delaunay Tessellation [4]. Let a *sphere* denote the boundary of an L_p-ball, and the interior of a sphere denote the interior of the corresponding L_p-ball. A sphere is said to *pass through* a set H of sites, $H \subset S$, if and only if its boundary passes through at least one site in H and it contains in its interior all the remaining sites in H. A sphere *contains* a set H of sites if and only if its interior contains all the sites in H. Given a sphere, sites located on its boundary, in its interior, and in its exterior are called *boundary sites*, *interior sites*, and *exterior sites*, respectively. A k-element subset of S is called a *k-subset*.

Definition 1 *Given a set S of n point sites $\in R^d$ and an L_p metric, a k-subset H, $H \subset S$, is called* valid *if there exists a sphere that contains H but does not contain any other k-subset; H is called* invalid*, otherwise. A valid k-subset H is represented as a graph-theoretic node, called a* k-NN Delaunay node*.*

A k-NN Delaunay node H can be embedded in R^d as a point in $V_k(H, S)$ i.e., as the center of a sphere that contains H but no other k-subset. A k-NN Delaunay node is a graph-theoretic node, however, it also has a geometric interpretation as it corresponds to the k-subset of sites it uniquely represents.

Definition 2 *Two k-Delaunay nodes, H_1 and H_2, are connected with a* k-NN Delaunay edge *(H_1, H_2) if and only if there exists a sphere that passes through both H_1 and H_2 but does not pass through any other k-subset. The graph $G(V, E)$, where V is the set of all the k-NN-Delaunay nodes, and E is the set of all k-NN Delaunay edges, is called a* k-NN Delaunay graph (under the corresponding L_p metric)*.*

Lemma 1 *Given a set S of point sites $\in R^d$, two k-NN Delaunay nodes H_1 and H_2, are joined by a k-NN Delaunay edge if and only if (1) $|H_1 \cap H_2| = k - 1$, and (2) There exists a sphere whose boundary passes through exactly two sites, $p \in H_1 \setminus H_2$ and $q \in H_2 \setminus H_1$, and whose interior contains $H_1 \cap H_2$ but does not contain any site $r \in S \setminus (H_1 \cup H_2)$.*

Let $H_1 \oplus H_2 = H_1 \setminus H_2 \cup H_2 \setminus H_1$. Following Lemma 1, a k-NN Delaunay edge (H_1, H_2), corresponds to a collection of spheres each of which passes through exactly two sites, p and $q \in H_1 \oplus H_2$, and contains exactly $k-1$ sites in $H_1 \cap H_2$ in its interior (Note that under the general position assumption, $|H_1 \oplus H_2| = 2$).

Theorem 1 *Given a set S of point sites $\in R^d$, the k-NN Delaunay graph of S is the graph-theoretic dual of the k-NN Voronoi diagram of S.*

3 Paradigm for the k-NN Delaunay Graph in R^2

In this section we present a paradigm to directly compute a k-NN Delaunay graph for a set S of n point sites $\in R^2$. We make the general position assumption

that no more than three sites are located on the same circle, where a circle is a sphere in R^2. Under the general position assumption, the k-NN Delaunay graph is a planar triangulated graph, in which any chordless cycle is a triangle, and we call it a k-NN Delaunay triangulation. This assumption is removed in Section 3.2.

Our paradigm consists of the following two steps:

1. (Section 3.1) Compute the *k-NN Delaunay hull* and all the *extreme k-NN Delaunay circuits*, defined in Section 3.1 (Definition 4).
2. (Section 3.2) For each extreme k-NN Delaunay circuit find a *k-NN Delaunay triangle* of a *k-NN Delaunay component*, defined in Section 3.1 (Definition 4), traverse from the k-NN Delaunay triangle to its adjacent k-NN Delaunay triangles, and repeatedly perform the traversal operation until all triangles of the k-NN Delaunay component have been traversed.

3.1 *k*-NN Delaunay Triangles, Circuits, Components and Hull

A *k-NN Delaunay triangle*, denoted as $T(H_1, H_2, H_3)$, is a triangle connecting three k-NN Delaunay nodes, H_1, H_2, and H_3, by three k-NN Delaunay edges. The circle passing through all the three k-subsets H_1, H_2, and H_3 is called the *circumcircle* of $T(H_1, H_2, H_3)$. Note that the circumcircle of $T(H_1, H_2, H_3)$ is a circle induced by three points $p \in H_1, q \in H_2$, and $r \in H_3$.

Lemma 2 *Given a set S of point sites in R^2, three k-NN Delaunay nodes, H_1, H_2, and H_3, form a k-NN Delaunay triangle if and only if (1) $|H_1 \cap H_2 \cap H_3| = k-1$ or $k-2$, and (2) there exists a circle that passes through $p \in H_1 \setminus H_2$, $q \in H_2 \setminus H_3$, and $r \in H_3 \setminus H_1$, contains $H_1 \cap H_2 \cap H_3$ in its interior, but does not contain any site $t \in S \setminus H_1 \cup H_2 \cup H_3$ in its interior or boundary. This circle is exactly the unique circumcircle of $T(H_1, H_2, H_3)$.*

Following Lemma 2, a k-NN Delaunay triangle $T(H_1, H_2, H_3)$ is also denoted as $T(p, q, r)$, where $p \in H_1 \setminus H_2$, $q \in H_2 \setminus H_3$, and $r \in H_3 \setminus H_1$ are the boundary sites of its circumcircle.

Definition 3 *An* unbounded circle *is a circle of infinite radius. A k-NN Delaunay node H is called* extreme *if there exists an unbounded circle that contains H but does not contain any other k-subset. A k-NN Delaunay edge is* extreme *if it connects two extreme k-NN Delaunay nodes.*

Definition 4 *The* k-NN Delaunay hull *is a cycle connecting all the extreme k-NN Delaunay nodes by the extreme k-NN Delaunay edges. An* extreme k-NN Delaunay circuit *is a simple cycle consisting of extreme k-NN Delaunay nodes and extreme k-NN Delaunay edges.* A k-NN Delaunay component *is a maximal collection of k-NN Delaunay triangles bounded by an extreme k-NN Delaunay circuit.*

Remark 1 *In the L_1, L_∞ metrics a k-NN Delaunay hull may consist of several extreme k-NN Delaunay circuits. In addition, a k-NN Delaunay graph may consist of several k-NN Delaunay components, and some k-NN Delaunay components may share a k-NN Delaunay node, called k-NN Delaunay cut node.*

A triangulated graph TG is *triangularly connected* if for each pair of triangles T_s and $T_t \in$ TG, there exists a sequence of triangles, $T_s = T_1, T_2, \cdots, T_l = T_t \in$ TG, where T_i and T_{i+1} are adjacent to each other (i.e., they share a common edge) for $1 \leq i < l$. A k-NN Delaunay component is triangularly connected.

To compute the k-NN Delaunay hull, we first find an extreme k-NN Delaunay edge, then traverse from it to its adjacent k-NN Delaunay edge, and repeatedly perform the traversal operation until all the extreme k-NN Delaunay edges have been traversed. Lemma 3 implies that there must always exist at least one extreme k-NN Delaunay edge.

Lemma 3 *Consider a set S of point sites in R^2. There exist two k-subsets, H_1 and H_2 and an unbounded circle that passes through H_1 and H_2 but does not pass through any other k-subset.*

Consider two adjacent extreme k-NN Delaunay edges (H_1,H_2) and (H_2, H_3), where $H_1 = H \cup \{p, r\}$, $H_2 = H \cup \{q, r\}$, and $|H| = k-2$. Since $|H_2 \cap H_3| = k-1$, by Lemma 1, H_3 is either $H \cup \{r, t\}$ or $H \cup \{q, t\}$ for some site $t \notin H$. If $H_3 = H \cup \{r, t\}$, there exists an unbounded circle that passes through q and t and contains $H \cup \{r\}$ but does not contain any other site. If $H_3 = H \cup \{q, t\}$, there exists an unbounded circle that passes through r and t and contains $H \cup \{q\}$ but does not contain any other site. For the former case, to identify H_3, it is enough to compute a site $t \notin H_1 \cup H_2$ such that the unbounded circle formed by q and t contains $H_1 \cap H_2$ but does not contain any other site. For the latter case, we compute a site $r \in H_1 \cap H_2$ and a site $t \notin H_2$ such that the unbounded circle formed by r and t contains $H_2 \setminus \{q\}$ but does not contain any other site. The details of the hull construction in the L_∞ metric are discussed in Section 4.1.

3.2 Traversal-Based Operation among Triangles

Under the general position assumption, a k-NN Delaunay triangle is dual to a k-NN Voronoi vertex (Theorem 1). A k-NN Delaunay triangle $T = T(H_1, H_2, H_3)$ is categorized as *new* or *old* according to the number of interior sites of its circumcircle as follows: if $|H_1 \cap H_2 \cap H_3| = k-1$, T is *new*, and if $|H_1 \cap H_2 \cap H_3| = k-2$, T is *old*. The terms new and old follow the corresponding terms for k-NN Voronoi vertices in [14].

We propose a circular wave propagation to traverse from $T_1 = T(H_1, H_2, H_3) = T(p, q, r)$ to $T_2 = T(H_1, H_2, H_4) = T(p, q, t)$. As mentioned in Section 2, a k-NN Delaunay edge (H_1, H_2) corresponds to a collection of circles whose boundary sites are exactly two sites, p and q, $\in H_1 \oplus H_2$ and whose interior sites are exactly $k-1$ sites $\in H_1 \cap H_2$. Therefore, traversal from T_1 to T_2 is like having a specific circular wave propagation which begins as the circumcircle of T_1, then follows the collection of circles corresponding to Delaunay edge (H_1, H_2) in a continuous order, and ends as the circumcircle of T_2. During the propagation, the circular wave keeps touching p and q and contains exactly the $k-1$ sites in $H_1 \cap H_2$ in its interior, while moving its center along $B(H_1, H_2) = B(p, q)$.

If T_1 is new, the circular wave moves along the direction of $B(p, q)$ that excludes r, and if T_1 is old, the circular wave moves along the opposite direction

of $B(p, q)$ to include r. Otherwise, the circular wave would not contain $k-1$ sites in its interior, and thus it would not correspond to the common k-NN Delaunay edge (H_1, H_2). The circular wave terminates when it touches a site $t \notin \{p, q, r\}$. If $t \in H_1 \cap H_2 \cap H_3$, the resulting circle contains $k-2$ sites in its interior and T_2 is old; if $t \notin H_1 \cup H_2 \cup H_3$, the resulting circle contains $k-1$ sites in its interior and T_2 is new.

Using the traversal operation and assuming that we can identify a k-NN Delaunay triangle in an extreme k-NN Delaunay circuit as a starting triangle, we can compute the entire incident k-NN Delaunay component. Since we already have all the extreme k-NN Delaunay edges after the k-NN Delaunay hull construction, we can use an extreme k-NN Delaunay edge to compute its incident k-NN Delaunay triangle and use it as a starting triangle.

If we remove the general position assumption, the dual of a k-NN Voronoi vertex becomes a chordless cycle of the k-NN Delaunay graph, called a *k-NN Delaunay cycle*, and Lemma 2 generalizes to Lemma 4.

Lemma 4 *Given a set S of point sites $\in R^2$, l k-NN Delaunay nodes, $H_1, H_2, \cdots$, and H_l, form a k-NN Delaunay cycle $(H_1, H_2, \cdots, H_l)$ if and only if (1) $k+1-l \leq |H_1 \cap H_2 \cap \cdots \cap H_l| \leq k-1$ and $|H_i \setminus H_{i+1}| = 1$, for $1 \leq i \leq l$, where $H_{l+1} = H_1$, (2) there exists a circle that passes through $c_1 \in H_1 \setminus H_2$, $c_2 \in H_2 \setminus H_3$, $\cdots$, and $c_l \in H_l \setminus H_1$, and contains $\bigcap_{1 \leq i \leq l} H_i$ but does not contain any site $t \in S \setminus \bigcup_{1 \leq i \leq l} H_i$ in its interior or boundary. This circle is the unique circumcircle of the l k-NN Delaunay nodes.*

We use Fig. 1 to illustrate Lemma 4. Fig. 1(a) shows a circle passing through five sites and containing three sites. According to Lemma 4, the circle corresponds to a k-NN Delaunay cycle, $4 \leq k \leq 7$. Fig. 1(b) shows a 5-NN Delaunay cycle. As shown in Fig. 1(c), in order to traverse from this 5-NN Delaunay cycle to its adjacent 5-NN Delaunay cycle via the 5-NN Delaunay edge (H_1, H_2), the corresponding circular wave will follow $B(H_1, H_2) = B(s_1, s_3)$ to exclude s_4 and s_5 and to include s_2 such that it contains $k-1=4$ sites, s_2, s_6, s_7, and s_8.

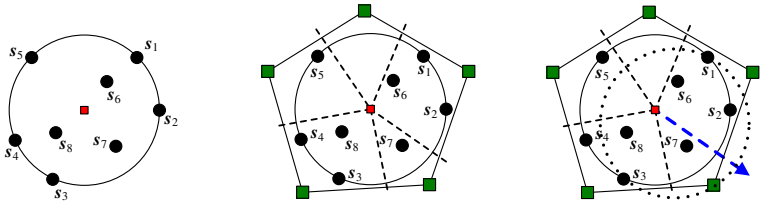

Fig. 1. (a) A circle passing through five sites (s_1, s_2, s_3, s_4, and s_5) and containing three sites (s_6, s_7, and s_8). (b) A 5-NN Delaunay cycle associated with H_1=$\{s_1, s_2, s_6, s_7, s_8\}$, H_2=$\{s_2, s_3, s_6, s_7, s_8\}$, H_3=$\{s_3, s_4, s_6, s_7, s_8\}$, H_4=$\{s_4, s_5, s_6, s_7, s_8\}$, and H_5=$\{s_1, s_5, s_6, s_7, s_8\}$. (c) The circular wave touches two sites (s_1 and s_3) and contains $k-1=4$ sites (s_2, s_6, s_7, and s_8).

4 Planar k-NN Delaunay Graph in the L_∞ Metric

We implement our paradigm in the L_∞ metric such that the hull construction takes $O(n \log n)$ time (Section 4.1) and each traversal operation between two triangles takes $O(\log n)$ time (Section 4.2). Since the number of traversal operations is bounded by the number m of k-NN Delaunay edges, we have an $O((n+m) \log n)$-time algorithm to directly compute the L_∞ planar k-NN Delaunay graph. In the L_∞ metric, general position is augmented with the assumption that no two sites are located on the same axis-parallel line.

4.1 L_∞ k-NN Delaunay Hull Computation

To compute the k-NN Delaunay hull, we traverse from one extreme k-NN Delaunay edge to all the others. In the L_∞ metric an unbounded circle passing through two sites is an axis-parallel L-shaped curve. An L-shaped curve partitions the plane into two portions, where one portion is a *quarter-plane*, illustrated shaded in Fig. 2. Therefore, an extreme k-NN Delaunay edge (H_1, H_2) corresponds to an L-shaped curve which passes through exactly two sites, p and $q \in H_1 \oplus H_2$, and whose quarter-plane exactly contains $k-1$ sites $\in H_1 \cap H_2$ in its interior. All the extreme k-NN Delaunay edges can be classified into four categories, $\{\text{NE}, \text{SE}, \text{SW}, \text{NW}\}$, according to the orientation of their corresponding quarter-plane.

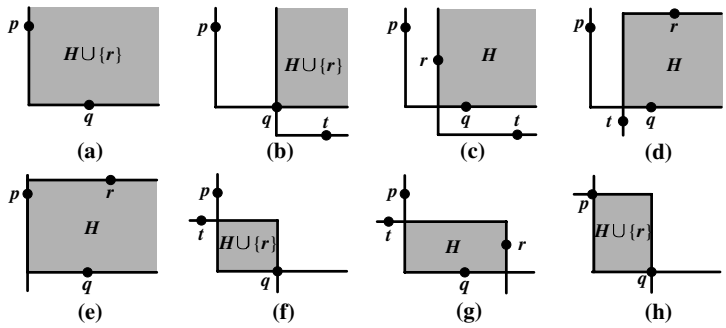

Fig. 2. $e_1 = (H_1, H_2)$ and $e_2 = (H_2, H_3)$, where e_1 is NE. (a) $H_1 = H \cup \{p, r\}$, and $H_2 = H \cup \{q, r\}$. (b)–(c) e_2 is NE. (b) $H_3 = H \cup \{r, t\}$. (c) $H_3 = H \cup \{q, t\}$. (d)–(e) e_2 is SE. (d) $H_3 = H \cup \{q, t\}$. (e) $H_3 = H \cup \{p, q\}$. (f)–(h) e_2 is SW. (f) $H_3 = H \cup \{r, t\}$. (g) $H_3 = H \cup \{q, t\}$. (h) $e_1 = e_2$.

Given an extreme k-NN Delaunay edge $e_1 = (H_1, H_2)$, we propose an approach to find its clockwise adjacent extreme k-NN Delaunay edge $e_2 = (H_2, H_3)$. We only discuss the cases where e_1 is NE, If $H_1 = H \cup \{p, r\}$ and $H_2 = H \cup \{q, r\}$, where $|H| = k-2$, then H_3 is either $H \cup \{r, t\}$ or $H \cup \{q, t\}$, as shown in Fig. 2(b) and Fig. 2(c). Below, we discuss the two cases of H_3 assuming that e_2 is NE, SE, SW, and NW, respectively. The coordinates of p, q, r, and t are denoted as (x_p, y_p), (x_q, y_q), (x_r, y_r), and (x_t, y_t), respectively. Moreover, ε is any value between 0 and the minimum distance among the sites.

1. e_2 is **NE** (Fig. 2(b)–(c)): We first drag a vertical ray, $[(x_p, y_q), (x_p, \infty)]$, right to touch a site $v \in H_2$. Then, we drag a horizontal ray, $[(x_v, y_q), (\infty, y_q)]$, down to touch a site $t \notin H_1 \cup H_2$. If $v = q$, $H_3 = H \cup \{r, t\}$; otherwise, $v = r$ and $H_3 = H \cup \{q, t\}$.
2. e_2 is **SE** (Fig. 2(d)–(e)): If H_3 is $H \cup \{r, t\}$, H_3 cannot exist since there does not exist an SE L-shaped curve which passes through q and a site $t \notin H_2$ and whose quarter half-plane contains $H \cup \{r\}$. Therefore, we first drag a horizontal ray, $[(x_p + \varepsilon, \infty), (\infty, \infty)]$, from infinity downward to touch a site r. Then, we drag a vertical ray, $[(x_p + \varepsilon, y_q), (x_p + \varepsilon, \infty)]$, right to touch a site v. As last, we drag a vertical ray, $[(x_v, y_r), (x_v, -\infty)]$, left to touch a site t. In fact, t is possibly p. Fig. 2(d)–(e) shows the two cases $t \neq p$ and $t = p$.
3. e_2 is **SW** (Fig. 2(f)–(h)): We first drag a vertical ray, $[(\infty, y_q), (\infty, \infty)]$, left to touch a site v. Then, we drag a horizontal ray, $[(x_p + \epsilon, \infty), (\infty, \infty)]$, from infinity downward to touch a site u. At last, we drag a horizontal ray, $[(x_p, y_u), (-\infty, y_u)]$, up to touch a site t. If $v = q$ and $t \neq p$, $H_3 = H \cup \{r, t\}$. If $v \neq q$, $H_3 = H \cup \{q, r\}$. If $v = q$ and $t = p$, $e_2 = e_1$, i.e., e_1 is also SW.
4. e_2 is **NW**: If e_2 is not NE, SE, or SW, e_1 must also be SW, and we consider this case as the case where e_1 is SW.

For each extreme k-NN Delaunay edge, this approach uses a constant number of segment dragging queries to compute its adjacent one. Chazelle [7] proposed an algorithmic technique to answer each orthogonal segment dragging query in $O(\log n)$ time using $O(n \log n)$-time preprocessing and $O(n)$ space.

A sequence of extreme k-NN Delaunay edges in the same category is called *monotonic*. An L_∞ k-NN Delaunay hull consists of at most four monotonic sequences of extreme k-NN Delaunay edges whose clockwise order follows $\{\text{NE}, \text{SE}, \text{SW}, \text{NW}\}$. Therefore, the number of extreme k-NN Delaunay edges is $O(n - k)$, and it takes $O(n \log n)$ time to compute the k-NN Delaunay hull. During the traversal procedure, once an extreme k-NN Delaunay node has been traversed an even number of times, an extreme k-NN Delaunay circuit has been constructed.

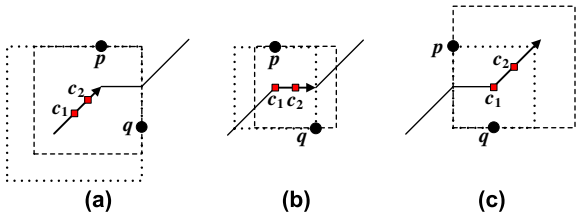

Fig. 3. Square wave propagation along $B(p, q)$. Solid segments are $B(p, q)$, arrow heads are moving directions, and c_1 and c_2 are the centers of dot squares and dash squares, respectively. (a) square contraction. (b) square movement. (c) square expansion.

4.2 L_∞ k-NN Traversal Operation among Triangles

As mentioned in Section 3.2, a traversal operation between triangles corresponds to a circular wave propagation whose center is located on the bisector $B(p, q)$

and which passes through p and q and contains $k-1$ sites in its interior. Since the circular wave propagation will terminate when it touches a site t, the problem reduces to computing the first site to be touched during the circular wave propagation.

In the L_∞ metric, a circle is an axis-parallel square, and a bisector between two points may consist of three parts as shown in Fig. 3. Therefore, a square wave propagation along an L_∞ bisector would consist of three stages: square contraction, square movement, and square expansion. As shown in Fig. 3, square contraction occurs when the center of the square wave moves along ray towards the vertical or horizontal segment, square movement occurs when the center moves along the vertical or horizontal segment, and square expansion occurs when the center moves along a 45° or 135° ray to infinity.

Below, we discuss the square wave propagation for k-NN Delaunay triangles. Without loss of generality, we only discuss the case that the three boundary sites are located on the left, bottom, and right sides of the corresponding square, respectively. The remaining cases are symmetric.

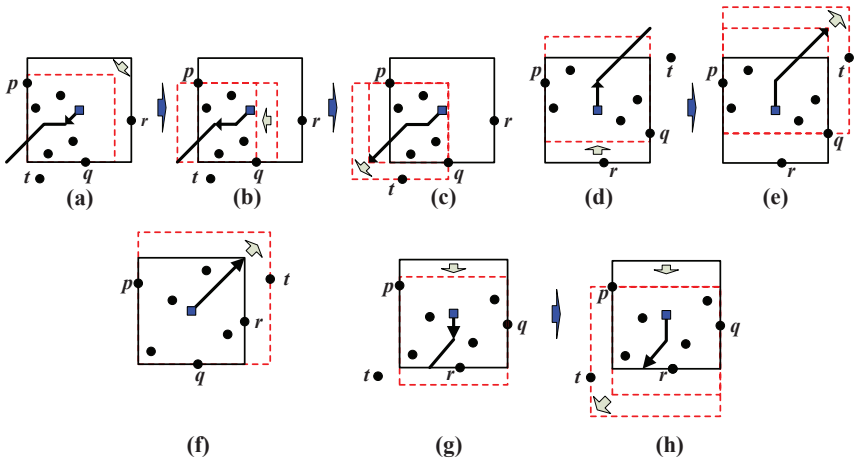

Fig. 4. Square wave propagation for $T(p,q,r)$. (a)–(e) $T(p,q,r)$ is new. (a)–(c) p and q are on adjacent sides. (d)–(e) p and q are on opposite sides. (f)–(h) $T(p,q,r)$ is old. (f) p and q are on adjacent sides. (g)–(h) p and q are on opposite sides.

For a new Delaunay triangle $T(p,q,r)$, the square wave propagation, traversing from $T(p,q,r)$ to its neighbor $T(p,q,t)$, will move along $B(p,q)$ to exclude r until it touches t. If p and q are located on adjacent sides, as shown in Fig. 4(a)–(c), the corresponding portion of bisector $B(p,q)$ may consist of three parts. This square wave propagation will begin as square contraction, then become square movement, and finally turn into square expansion until it first touches a site t. If p and q are located on parallel sides, as shown in Fig. 4(d)–(e), the corresponding portion of bisector $B(p,q)$ may consist of two parts. This square wave propagation will begin as square movement, and finally turn into square expansion until it first touches a site t.

For an old Delaunay triangle $T(p, q, r)$, the square wave propagation, traversing from $T(p, q, r)$ to its neighbor $T(p, q, t)$, will move along $B(p, q)$ to include r until it touches t. If p and q are located on adjacent sides, as shown in Fig. 4(f), the corresponding portion of bisector $B(p, q)$, is just a 45° or 135° ray, and thus this square wave propagation is just square expansion until it first touches a site t. If p and q are located on parallel sides, as shown in Fig. 4(g)–(h), the corresponding portion of bisector $B(p, q)$ may consist of two parts. This square wave propagation will begin as square movement, and then turn into square expansion until it first touches a site t.

Square contraction is equivalent to dragging two axis-parallel segments perpendicularly and then selecting the closer one of their first touched sites (see Fig. 5(a)). Square movement is similar to square contraction, but the two dragged segments are parallel to each other (see Fig. 5(b)). Since an orthogonal segment-dragging query can be computed in $O(\log n)$ time after $O(n \log n)$-time preprocessing [7], both square contraction and square movement can be answered in $O(\log n)$ time. On the other hand, square expansion is equivalent to four segment-dragging queries (see Fig. 5(c)). However, two of the four segment-dragging queries fall into a new class, in which one endpoint is located on a fixed vertical or horizontal ray, and the other endpoint is located on a fixed 45° or 135° ray. In [15], Mitchell stated that this class of segment dragging queries can be transformed into a point location query in a specific linear-space subdivision (see Fig. 5(d)), and thus this class of segment dragging queries can be answered in $O(\log n)$ time using $O(n \log n)$-time preprocessing and $O(n)$ space. Therefore, square expansion can also be answered in $O(\log n)$ time.

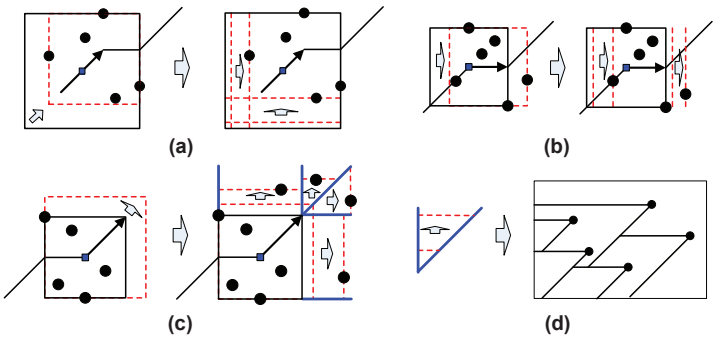

Fig. 5. Square wave propagation and segment-dragging queries. (a) square contraction. (b) square movement. (c) square expansion. (d) the second class of segment-dragging query and the point location query.

To conclude, each traversal operation for any k-NN Delaunay triangle takes at most one square contraction, one square movement, and one square expansion, and thus it can be computed in $O(\log n)$ time.

4.3 Complexity of L_∞ k-NN Voronoi diagram

In this section, we use the Hanan grid [13] to derive a tighter bound on the structural complexity of the L_∞ k-NN Voronoi diagram. Given a set S of n point sites in the plane, the Hanan grid is derived by drawing the axis-parallel lines through every point in S.

Given the L_∞ k-NN Delaunay graph, the L_∞ circumcircle of a k-NN Delaunay triangle is a unique square, called *k-NN Delaunay square*, that passes through three sites (assuming general position i.e., no two points are on the same axis parallel line) and contains $k-1$ (in case of a new triangle) or $k-2$ (in case of an old triangle) sites in its interior. The center of a k-NN Delaunay square is exactly a k-NN Voronoi vertex.

Lemma 5 *In the L_∞ metric, a k-NN Delaunay square must have at least two corners on the Hanan grid.*

Theorem 2 *The structural complexity of the L_∞ k-NN Voronoi diagram is $O((n-k)^2)$.*

Proof. Let us number the rows and columns of the Hanan grid 1 to n from right to left and from top to bottom respectively. Let us assume that points are in general position i.e., no two points are on the same axis parallel line. Let p be a point on the Hanan grid such that p is the NW corner of a k-NN Delaunay square D. Square D must enclose exactly $k+1$ or $k+2$ sites, including sites on its boundary, and thus no point on the Hanan grid past column $(n-k)$ or below row $(n-k)$ can serve as a NW corner to a k-NN Delaunay square. Hence, there are at most $(n-k)^2$ Hanan grid points that can serve as NW corners of a k-NN Delaunay square. Similarly for all four corner types of a k-NN Delaunay square. In addition, point p can be the NW corner of at most two k-NN Delaunay squares, one containing $k+1$ sites and the other containing $k+2$ sites. By Lemma 5, a k-NN Delaunay square must have at least two corners on the Hanan grid. Thus, there can be at most $O((n-k)^2)$ distinct k-NN Delaunay squares, and $O((n-k)^2)$ distinct k-NN Delaunay triangles.

As shown in [14], the size of the k-NN Voronoi diagram is $O(k(n-k))$ and the bound remains valid in the L_P metric. Thus, the following corollary is implied.

Corollary 1 *The structural complexity of the L_∞ k-NN Voronoi diagram, equivalently the k-NN Delaunay graph, is $O(\min\{k(n-k),(n-k)^2\})$.*

5 Conclusion

Based on our proposed traversal-based paradigm, we develop an $O((n+m)\log n)$-time algorithm for the L_∞ k-NN Voronoi diagram of size m using segment-dragging queries. This bound is output-sensitive and it can be considerably smaller than the time complexities of previous methods in the Euclidean metric, $O(nk^2\log n)$ of [14] and $O(n^2\log n + k(n-k)\log^2 n)$ or $O(n^2 + k(n-k)\log^2 n)$ of [8]. Since the L_∞ k-NN Voronoi diagram can be computed directly, it is likely that the L_2 k-NN Voronoi diagram can also be computed in a similar manner.

References

1. M. Abellanas, P. Bose, J. Garcia, F. Hurtado, C. M. Nicolas, and P. A. Ramos. On structural and graph theoretic properties of higher order Delaunay graphs. *Internat. J. Comput. Geom. Appl.*, 19(6), pp. 595-615, 2009
2. P. K. Agarwal, M. de Berg, J. Matousek, and I, Schwarzkopf, "Constructing levels in arrangements and higher order Voronoi diagrams," *Siam J. on Computing*, Vol. 27, No.3, pp. 654–667, 1998.
3. A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor, "A linear-time algorithm for computing Voronoi diagram of a convex polygon," *Discrete and Computational Geometry*, Vol. 4, pp. 591–604, 1984.
4. F Aurenhammer and R Klein, "Voronoi Diagrams," *Handbook of Computational Geometry*, Elseiver, 2000 .
5. F. Aurenhammer and O. Schwarzkopf, "A simple on-line randomized incremental algorithm for computing higher order Voronoi diagrams," *Internat. J. Comput. Geom. Appl.*, Vol. 2, pp. 363–381, 1992.
6. J. D. Boissonnat, O. Devillers, and M. Teillaud, "A semidynamic construction of higher-order Voronoi diagrams and its randomized analysis," Algorithmia, Vol. 9, pp. 329–356, 1993.
7. B. Chazelle,"An algorithm for segment dragging and its implementation," *Algorithmica*, Vol. 3, pp. 205–221, 1988.
8. B. Chazelle and H. Edelsbrunner, "An improved algorithm for constructing kth-order Voronoi Diagram," *IEEE Trans. on Computers*, Vol. 36, No.11, pp. 1349–1454, 1987.
9. K. L. Clarkson, "New applications of random sampling in computational geometry," *Discrete and Computational Geometry*, Vol. 2, pp. 195–222, 1987.
10. K. L. Clarkson and P. W. Shor, "Applications of random sampling in computational geometry, II," *Discrete and Computational Geometry*, Vol. 4, pp. 387-421, 1989.
11. H. Edelsbrunner, J. O'Rourke, and R. Seidel, "Constructing arrangements of lines and hyperplanes with applications," *SIAM J. on Computing*, Vol. 15, pp. 341–363, 1986.
12. J. Gudmundsson, M. Hammar, and M. van Kreveld, "Higher order Delaunay triangulations," *Computaional Geometry*, Vol. 23, No. 1, pp. 85–98, 2002.
13. M. Hanan, "On Steiner's problem with rectilinear distance", SIAM J. on Applied Mathematics, Vol. 14, pp. 255–265, 1966.
14. D.-T. Lee, "On k-nearest neighbor Voronoi Diagrams in the plane," *IEEE Trans. on Computers*, Vol. 31, No. 6, pp. 478–487, 1982.
15. J. S. B. Mitchell, "L1 Shortest Paths Among Polygonal Obstacles in the Plane," *Algorithmica*, Vol. 8, pp. 55–88, 1992.
16. K. Mulmuley, "On levels in arrangements and Voronoi diagrams," *Discrete and Computational Geometry*," Vol. 6, pp. 307–338, 1991.
17. E. Papadopoulou, "Critical Area computation for missing material defects in VLSI circuits," *IEEE Trans. on CAD*, Vol. 20, No.5, pp. 583–597, 2001.
18. E. Papadopoulou, "Net-aware Critical area extraction for opens in VLSI circuits via higher-order Voronoi diagrams", *IEEE Trans. on CAD*, 30(5), 704–716, 2011.
19. E. Papadopoulou and D.-T. Lee, "The L_∞ Voronoi Diagram of Segments and VLSI Applications," *Internat. J. Comput. Geom. Appl.*, Vol. 11, No. 5, pp. 503–528, 2001.