Algorithmica (2015) 71:429–449
DOI 10.1007/s00453-013-9809-9

The k-Nearest-Neighbor Voronoi Diagram Revisited

Chih-Hung Liu · Evanthia Papadopoulou · Der-Tsai Lee

Received: 16 October 2012 / Accepted: 22 June 2013 / Published online: 13 July 2013

Abstract We revisit the k-nearest-neighbor (k-NN) Voronoi diagram and present a new paradigm for its construction. We introduce the k-NN Delaunay graph, which is the graph-theoretic dual of the k-NN Voronoi diagram, and use it as a base to directly compute this diagram in R^2. We implemented our paradigm in the L_1 and L_∞ metrics, using segment-dragging queries, resulting in the first output-sensitive, $O((n+m)\log n)$-time algorithm to compute the k-NN Voronoi diagram of n points in the plane, where m is the structural complexity (size) of this diagram. We also show that the structural complexity of the k-NN Voronoi diagram in the L_∞ (equiv. L_1) metric is $O(\min\{k(n-k),(n-k)^2\})$. Efficient implementation of our paradigm in the L_2 (resp. L_p, $1 < p < \infty$) metric remains an open problem.

Keywords Voronoi diagrams · Higher-order Voronoi diagrams · k Neareast neighbors · Segment dragging queries

A preliminary version appeared in *Proc. European Symposium on Algorithms* (ESA) 2011 [22]. This work was supported in part by the Swiss National Science Foundation under Grants SNF-200021-127137, and SNF-20GG21-134355, the latter under the ESF EUROCORES program EuroGIGA/VORONOI, and by the National Science Council, Taiwan under Grants Nos. NSC-99-2911-I-001-506, NSC-99-2911-I-001-511, NSC-101-2221-E-005-019-MY2, and NSC-101-2221-E-005-026-MY2.

C.-H. Liu · D.T. Lee (✉)
Academia Sinica, Taipei, Taiwan
e-mail: dtlee@nchu.edu.tw

C.-H. Liu
e-mail: chliu_10@citi.sinica.edu.tw

E. Papadopoulou
Università della Svizzera Italiana, Lugano, Switzerland
e-mail: evanthia.papadopoulou@usi.ch

D.T. Lee
National Chung Hsing University, Taichung, Taiwan

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* of S, abbreviated as *k-NN Voronoi diagram*, partitions R^d into Voronoi regions such that all points in a Voronoi region share the same k nearest sites.

In the Euclidean plane, the diagram was first introduced by Shamos and Hoey [30] in order to deal with k nearest neighbor problem. Lee [21] showed that its structural complexity is $O(k(n-k))$ and proposed an iterative algorithm to construct the diagram within $O(k^2 n \log n)$ time and $O(k^2(n-k))$ space. Based on the notions of arrangements and geometric duality, Chazelle and Edelsbrunner [13] developed two versions of an algorithm, the first of which takes $O(n^2 \log n + k(n-k)\log^2 n)$ time and $O(k(n-k))$ space and the other of which takes $O(n^2 + k(n-k)\log^2 n)$ time and (n^2) space. Aurenhammer [6] proposed a new duality between k-NN Voronoi diagrams in E^d and the convex hulls in E^{d+1}, which implies a reasonably simple algorithm for computing the k-NN Voronoi diagram in E^2 within $O(k^2 n \log n)$ time and $O(k(n-k))$ space. Agarwal et al. [4] proposed an algorithm to compute the Voronoi diagram in the plane within $\Theta(n)$ time when all the sites are in convex position, and claimed that using their linear procedure, the time complexity of Lee's approach [21] can be reduced to $O(nk^2 + n \log n)$. In higher dimensions, Edelsbrunner et al. [16] showed the relation between higher-order Voronoi diagrams in E^d and arrangements of hyperplanes in E^{d+1}, and devised an algorithm to compute all $V_k(S)$ in E^d, for $1 \leq k \leq n-1$, within optimal $O(n^{d+1})$ time and space. Clarkson and Shor [15] showed that the size of the $\leq k$-NN Voronoi diagrams, $V_1(S), V_2(S), \ldots, V_k(S)$, is $O(k^{\lceil (d+1)/2 \rceil} n^{\lfloor (d+1)/2 \rfloor})$.

Aside from these deterministic algorithms, Clarkson [14] proposed a randomized approach that takes $O(kn^{1+\varepsilon})$ expected time and space, and Mulmuley [24] proposed a randomized incremental algorithm of expected $O(nk^2 + n \log n)$ time and $O(nk^2)$ space. Agarwal et al. [2] developed a randomized incremental approach to compute the k-level in an arrangement of n planes, which yields a randomized algorithm for computing the k-NN Voronoi diagram of n sites in the plane within expected time $O(k(n-k)\log n + n \log^3 n)$, which is very close to the asymptotic lower bound of $O(k(n-k) + n \log n)$. Chan [11] and Ramos [28] later improved the expected time complexity to $O(n \log n + nk \log k)$ and $O(n \log n + nk 2^{O(\log^* k)})$, respectively, which is near optimal.

For on-line algorithms, Boissonnat et al. [9] developed an on-line randomized incremental algorithm, which takes expected $O(n \log n + nk^3)$ time and expected $O(nk^2)$ space. After that, Aurenhammer and Schwarzkopf [8] proposed a simple on-line randomized incremental algorithm for computing the k-NN Voronoi diagram within expected $O(nk^2 \log n + nk \log^3 n)$ time and optimal $O(k(n-k))$ space. They also provided a more sophisticated version to compute the k-NN Voronoi diagram within expected $O(nk^2 + nk \log^2 n)$ time. Furthermore, they proved that for any randomized incremental construction of the k-NN Voronoi diagram, if all intermediate k-NN Voronoi diagrams are maintained, the expected number of Voronoi vertices that appear at some intermediate stage during the construction is $\Theta(nk^2)$. This proof implies that the lower bound of the expected time complexity of any on-line k-NN

Voronoi diagram construction is $\Omega(nk^2)$, and the time complexity of their algorithm is very close to the optimal one.

All the above algorithms [2, 8, 9, 13, 14, 21, 24] 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 practical applications in VLSI design, see e.g., [25–27]. Most existing algorithms compute k-NN Voronoi diagrams using reductions to arrangements or geometric duality [2, 8, 13, 14, 24] which are not directly applicable to the L_1, L_∞ metrics. Furthermore, none of the existing deterministic algorithms is output-sensitive. Among them the $O(k^2 n \log n)$ time complexity of [6, 21] is best for small k, and the $O(n^2 + k(n-k)\log^2 n)/O(n^2 + k(n-k)\log^2 n)$ time complexity of [13] is the best for large k. The algorithms in [6, 21] iteratively derive the k-NN Voronoi diagram from the $(k-1)$-NN Voronoi diagram, and thus those algorithms will generate $V_1(S), V_2(S), \ldots, V_k(S)$, which leads to a lower bound of time complexity $\Omega(nk^2)$. The algorithm in [13] needs to generate $\Theta(n^2)$ bisectors for the n arrangements (each arrangement corresponds to a site and corresponds to $n-1$ bisectors), while not all bisectors appear in $V_k(S)$.

In this paper, we revisit the k-NN Voronoi diagram and propose an output-sensitive approach to compute the L_∞ (equiv. L_1) k-NN Voronoi diagram. We first formulate the k-NN Delaunay graph, which is the graph-theoretic dual of the k-NN Voronoi diagram.[1] We then develop a traversal-based paradigm to directly compute the k-NN Delaunay graph of point sites in the plane. We implement our paradigm in the L_∞ metric using segment-dragging techniques, and present an $O((m+n)\log n)$-time algorithm for the L_∞ planar k-NN Delaunay graph of size m. Efficiently implementing our paradigm in the L_2 or the L_p, $1 < p < \infty$, metric is left as an open problem. We also prove that the structural complexity of the L_∞ k-NN Voronoi diagram is $O(\min\{k(n-k), (n-k)^2\})$, a tighter than $O(k(n-k))$ bound for the L_∞ metric. Note that the L_∞ farthest-neighbor Voronoi diagram has size $O(1)$. Since the L_1 metric is equivalent to L_∞ under rotation, these results are also applicable to the L_1 metric

The organization of this paper is as follows. Section 3 formulates the k-NN Delaunay graph and derives its basic properties. Section 4 presents a traversal-based paradigm to construct the planar k-NN Delaunay graph directly. Section 5 develops an algorithm to compute the L_∞ planar k-NN Delaunay graph. Section 6 analyzes the structural complexity of the L_∞ k-NN Voronoi diagram.

2 Problem Formulation

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 $1 \le p < \infty$, and $d_\infty(s,t) = \max(|x_{1_s} - x_{1_t}|, |x_{2_s} - x_{2_t}|, \ldots, |x_{d_s} - x_{d_t}|)$. For two points $p, q \in R^d$, the bisector $B(p,q)$ is $\{r \in R^2 \mid d_p(r,p) = d_p(r,q)\}$. In this paper, the distance between a point p and a set H of points is the farthest distance $d_f(p,H) = \max\{d_p(p,q), \forall q \in H\}$.

[1] Note the k-NN Delaunay graph is different from the geometric k-Delaunay graph of [1, 18].

The bisector between two sets of points H_1 and H_2 is $B(H_1, H_2) = \{r \in R^d \mid d_f(r, H_1) = d_f(r, H_2)\}$.

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* of S, 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 $H \subset S$ of cardinality k than to any other subset $H' \subset S$ of cardinality k, i.e., $d_f(H, S) \leq d_f(H', S)$, and is denoted as $V_k(H, S)$. A subset of S of cardinality k is called a *k-subset*.

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 a portion of the bisector $B(H_1, H_2)$. 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 regions is a *k-NN Voronoi vertex*.

3 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 (see e.g., [7]). An L_p ball with center p and radius r is the locus of points x satisfying $d_p(x, p) \leq r$. The boundary of an L_p-ball is called a *sphere* for $d > 2$ or a *circle* for $d = 2$. A sphere is said to *pass through* a set H of sites, $H \subset S$, if its boundary passes through at least one site in H and its interior contains all the other sites in H. Note that if a sphere passes through a set H of sites, its radius is exactly the farthest distance between its center and 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.

Definition 1 Given a set S of n point sites $\in R^d$, 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 by a graph-theoretic node, called a *k-NN Delaunay node*.

Definition 2 Two k-NN Delaunay nodes, H_1 and H_2, are connected with a *k-NN Delaunay edge* (H_1, H_2) if there exists a sphere that passes through both H_1 and H_2 but does not pass through any other valid 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)*.

A k-NN Delaunay node H is a graph-theoretic node, which can be interpreted geometrically as a point within $V_k(H, S)$. Note that the k-NN Delaunay graph is a graph-theoretic structure, defined on k-NN Delaunay nodes, and it is different from the order-k Delaunay graph in [1, 18], which is a geometric graph on the point set S.

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 $r \in S \setminus (H_1 \cup H_2)$.

Proof

Necessity: Let H_1, H_2 be two Delaunay nodes joined by a k-NN Delaunay edge. Then $|H_1 \cap H_2| \leq k-1$, as otherwise H_1 and H_2 cannot be two distinct k-subsets. If $|H_1 \cap H_2| = k' < k-1$, the sphere satisfying Definition 2 will contain $2k - k' > k+1$ sites in its interior and boundary, which must form at least one other valid k-subset, contradicting Definition 2. Therefore, $|H_1 \cap H_2| \geq k-1$, and thus $|H_1 \cap H_2| = k-1$. If condition (1) holds, the sphere satisfying Definition 2 will also satisfy condition (2).

Sufficiency: Suppose both conditions (1) and (2) hold. It is clear that the sphere satisfying condition 2 also satisfies Definition 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, $p \in H_1 \setminus H_2$ and $q \in H_2 \setminus H_1$, where $p, q \in H_1 \oplus H_2$, 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.*

Proof By the definitions so far it is clear that a k-NN Voronoi region $V_k(H, S)$ is non-empty if and only if the k-subset H is valid. Thus, it is enough to prove that two Voronoi regions share a $(d-1)$-dimensional face, called a *facet*, if and only if their corresponding k-NN Delaunay nodes are joined by a k-NN Delaunay edge in the k-NN Delaunay graph.

Let $\{a\} \cup H$ and $\{b\} \cup H$, where $|H| = k-1$, be two valid k-subsets. If their corresponding k-NN Voronoi regions share a facet then the elements of $\{a, b\} \cup H$ are the $k+1$ nearest neighbors of all vertices along this facet, and sites a, b are the kth nearest neighbors of these vertices. Hence, there exists a sphere that passes through $\{a\} \cup H$ and $\{b\} \cup H$ but does not contain any other valid k-subset, implying that the two k-NN Delaunay nodes, $\{a\} \cup H$ and $\{b\} \cup H$, share a k-NN Delaunay edge. Conversely, by Definition 2, if the two k-NN Delaunay nodes, $\{a\} \cup H$ and $\{b\} \cup H$, share a k-NN Delaunay edge, there exists a sphere that passes through $\{a\} \cup H$ and $\{b\} \cup H$ but does not contain any other valid k-subsets. Therefore, the sites in $\{a, b, \} \cup H$ are the $k+1$ nearest neighbors of the center of the sphere, implying that the two Voronoi regions associated respectively with $\{a\} \cup H$ and $\{b\} \cup H$ share a facet. □

4 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 $\in R^2$. For simplicity, we make the general position assumption that no more than three sites are located on the same circle. Under the general position assumption, since the degree of each Voronoi vertex of $V_k(S)$ is exactly 3, the k-NN

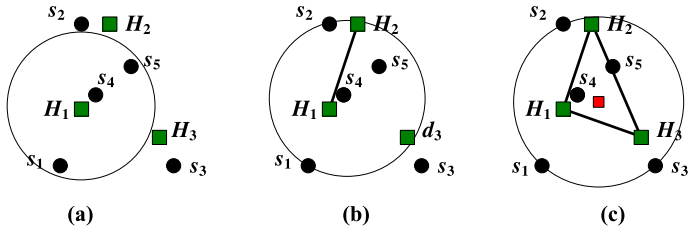

Fig. 1 A 3-NN Delaunay triangle $T(H_1, H_2, H_3)$, where $H_1 = \{s_1, s_4, s_5\}$, $H_2 = \{s_2, s_4, s_5\}$, and $H_3 = \{s_3, s_4, s_5\}$. (**a**) 3-NN Delaunay node H_1. (**b**) 3-NN Delaunay edge (H_1, H_2). (**c**) $T(H_1, H_2, H_3)$ and its circumcircle

Delaunay graph, the graph-theoretic dual of $V_k(S)$, is a planar triangulated graph, in which any chordless cycle is a triangle, called a k-NN Delaunay triangulation. Note that the unbounded face is not triangulated. The general position assumption is removed in Sect. 4.5.

Our paradigm consists of two steps as follows; definitions and details are given in the sections to follow.

1. Compute the *k-NN Delaunay hull* and all the *extreme k-NN Delaunay circuits*. For definitions and algorithms see Sects. 4.2 and 4.3, respectively.
2. For each extreme k-NN Delaunay circuit, determine a *k-NN Delaunay triangle* of a *k-NN Delaunay component*, defined in Sect. 4.1, 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. See Sect. 4.4.

The remaining of this section is organized as follows. Section 4.1 defines a k-NN Delaunay triangle and describes its properties. Section 4.2 defines a k-NN Delaunay hull, an extreme k-NN Delaunay circuit, and a k-NN Delaunay component. Section 4.3 presents a mechanism to construct a k-NN Delaunay hull and all extreme k-NN Delaunay circuits. Section 4.4 illustrates a traversal-based operation that computes all the k-NN Delaunay triangles of a k-NN Delaunay component for an extreme k-NN Delaunay circuit. Finally, Sect. 4.5 discusses degenerate cases removing the general position assumption.

4.1 k-NN Delaunay Triangles

Definition 3 Given a set S of point sites $\in R^2$, 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 three k-subsets H_1, H_2, and H_3 is called the *circumcircle* of $T(H_1, H_2, H_3)$.

Definition 3 is illustrated in Fig. 1. In Fig. 1(a) there exists a circle that contains $H_1 = \{s_1, s_4, s_5\}$ but does not contain any other valid 3-subset, thus H_1 is valid and induces a 3-NN Delaunay node. Similarly, there exist two 3-NN Delaunay nodes, H_2 and H_3. In Fig. 1(b), there exists a circle that passes through H_1 and H_2 but does

not pass through or contain any other valid 3-subset, thus there exists a 3-NN Delaunay edge (H_1, H_2). Similarly, there exist two other 3-NN Delaunay edges, (H_2, H_3) and (H_1, H_3). Figure 1(c) illustrates $T(H_1, H_2, H_3)$ and its circumcircle. The boundary sites, interior sites, and exterior sites of $T(H_1, H_2, H_3)$ are exactly the boundary, interior, and exterior sites of the circumcircle.

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 $T(H_1, H_2, H_3)$.*

Proof

Necessity: Let $T(H_1, H_2, H_3)$ be a k-NN Delaunay triangle. Then by Lemma 1, $|H_1 \cap H_2| = k-1$, $|H_2 \cap H_3| = k-1$, and $|H_3 \cap H_1| = k-1$. Thus, either $H_1 = H \cup \{p\}$, $H_2 = H \cup \{q\}$, and $H_3 = H \cup \{p\}$, where $|H| = k-1$, or $H_1 = H \cup \{p, q\}$, $H_2 = H \cup \{q, r\}$, and $H_3 = H \cup \{p, r\}$, where $|H| = k-2$, implying that $|H_1 \cap H_2 \cap H_3| = k-1$ or $k-2$. (As it will be defined later in Sect. 4.1, $T(H_1, H_2, H_3)$ is called *new* in the former case and *old* in the latter case.) By Lemma 1, there exists one circle which exactly passes through p, q and exactly contains $H_1 \cap H_2$, one circle which exactly passes through q, r and exactly contains $H_2 \cap H_3$, and one circle which exactly passes through p, r and exactly contains $H_3 \cap H_1$, implying that there exists a unique circle that passes through p, q, and r and contains $H_1 \cap H_2 \cap H_3$.

Sufficiency: Suppose that conditions (1) and (2) hold. Let's first assume that $|H_1 \cap H_2 \cap H_3| = k-2$. Then $H_1 = \{p, r\} \cup H$, $H_2 = \{q, r\} \cup H$, and $H_3 = \{p, q\} \cup H$, where $|H| = k-2$. Moving the unique circle along $B(p, q)$ to include r will form a circle which exactly passes through p and q and exactly contains $H_1 \cap H_2$, implying that H_1 and H_2 must be joined by a k-NN Delaunay edge. For the same reason, there exists a k-NN Delaunay edge between H_2 and H_3 and a k-NN Delaunay edge between H_1 and H_3, respectively.

Let's now assume that $|H_1 \cap H_2 \cap H_3| = k-1$. Then $H_1 = \{p\} \cup H$, $H_2 = \{q\} \cup H$, and $H_3 = \{p\} \cup H$, where $|H| = k-1$. Moving the unique circumcircle along $B(p, q)$ to exclude r induces a circle that passes through p and q and contains $H_1 \cap H_2$, implying that H_1 and H_2 must be joined by a k-NN Delaunay edge. For the same reason, there exists a k-NN Delaunay edge between H_2 and H_3 and a k-NN Delaunay edge between H_1 and H_3, respectively. □

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.

4.2 k-NN Delaunay Circuits, Components and Hull

Definition 4 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

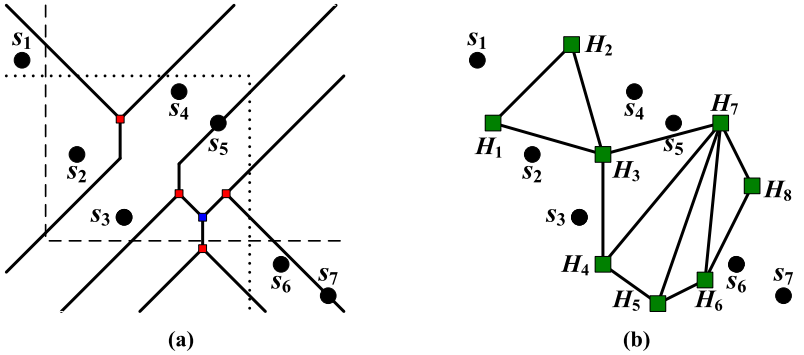

Fig. 2 (**a**) $V_4(S)$ (*solid segments*). (**b**) 4-NN Delaunay triangulation, the graph-theoretic dual of (**a**), where $H_1 = \{s_1, s_2, s_3, s_4\}$, $H_2 = \{s_1, s_2, s_4, s_5\}$, $H_3 = \{s_2, s_3, s_4, s_5\}$, $H_4 = \{s_2, s_3, s_5, s_6\}$, $H_5 = \{s_2, s_3, s_6, s_7\}$, $H_6 = \{s_3, s_5, s_6, s_7\}$, $H_7 = \{s_3, s_4, s_5, s_6\}$, and $H_8 = \{s_4, s_5, s_6, s_7\}$. The *dashed curve* and the *dotted curve* represent two unbounded circles which both exactly contain $H_3 = \{s_2, s_3, s_4, s_5\}$

not contain any other k-subset. A k-NN Delaunay edge is called *extreme* if it connects two extreme k-NN Delaunay nodes.

In the L_p metric, $1 < p < \infty$, an unbounded circle is a straight line, and its interior is one of the two open half-planes separated by the straight line [20]. A k-subset H of S that can be separated from S by a straight line is called a k-set [17, 29]. Since a k-set must be fully contained in one of the two open half-planes separated by the defining straight line, by Definition 4, a k-set always corresponds to an extreme k-NN Delaunay node in the L_p metric, $1 < p < \infty$. Therefore, the number of k-sets is equivalent to the number of extreme k-NN Delaunay nodes in the L_p metric, $1 < p < \infty$.

In the L_1 and L_∞ metrics, an unbounded circle is an L-shaped curve, and its interior is a quarter plane as defined by the two rays of the L-shaped curve. For example, both the dashed curve and the dotted curve in Fig. 2(a) are an unbounded circle. Since the dashed curve (or the dotted curve) exactly contains $H_3 = \{s_2, s_3, s_4, s_5\}$ in Fig. 2(a), H_3 is a 4-NN extreme Delaunay node in Fig. 2(b).

Consider a planar embedding of the k-NN Delaunay graph obtained by mapping a k-NN Delaunay node to a point within $V_k(H, S)$. Then an extreme k-NN Delaunay vertex and an extreme k-NN Delaunay edge must be incident to the outer face of the embedded k-NN Delaunay graph. We define a k-NN Delaunay hull to represent the outer boundary of a planar k-NN Delaunay graph.

Definition 5 A *k-NN Delaunay hull* is a cycle connecting all the extreme k-NN Delaunay nodes by all the extreme k-NN Delaunay edges. An *extreme k-NN Delaunay circuit* is a simple cycle of the k-NN Delaunay hull. *A k-NN Delaunay component* is a maximal collection of k-NN Delaunay triangles bounded by an extreme k-NN Delaunay circuit. A k-NN Delaunay node shared by more than one k-NN Delaunay components is called a k-NN Delaunay *cut* node.

Remark 1 In the L_p metric, $1 < p < \infty$, a k-NN Delaunay hull is always a simple cycle, the k-NN Delaunay graph consists of only one component, and no k-NN

Delaunay *cut* node exists (see Lemma 3). In contrast, in the L_1, L_∞ metric a k-NN Delaunay hull may consist of several extreme k-NN Delaunay circuits. In addition, a k-NN Delaunay graph may contain several k-NN Delaunay components, and some k-NN Delaunay components may share a common k-NN Delaunay *cut* node.

If H is a k-NN Delaunay cut node, two different unbounded circles must exist whose interiors both exactly contain H and whose intersection is bounded. For example, since both the dashed curve and the dotted curve in Fig. 2(a) exactly contain $H_3 = \{s_2, s_3, s_4.s_5\}$ and their intersection is bounded, in Fig. 2(b), H_3 is a 4-NN cut node. This is because if the intersection is unbounded, the two unbounded circles can be transformed into each other by rotating and translating without hitting any other sites, and if each pair of two unbounded circles which exactly contain H can be transformed into each other, H has at most two extreme k-NN Delaunay edge, contradicting the fact that H is a k-NN Delaunay cut node.

Lemma 3 *Given a set S of sites* $\in R^2$, *the k-NN Delaunay graph in the* L_p *metric*, $1 < p < \infty$, *contains exactly one extreme k-NN Delaunay component*.

Proof In the L_p metric, $1 < p < \infty$, the interior of an unbounded circle is an open half-plane. Since the intersection of two half-planes is unbounded, for any k-subset H, there cannot exist two unbounded circles whose interiors contain H and whose intersection is bounded. Therefore, there can be no k-NN Delaunay cut node, and thus the k-NN Delaunay graph contains exactly one k-NN Delaunay component. □

Corollary 1 *Given a set S of sites* $\in R^2$, *the k-NN Delaunay graph in the* L_p *metric*, $1 < p < \infty$, *contains no cut node*, *i.e.*, *no multiple k-NN Delaunay components*.

A graph is called a *triangulated graph* if any chordless cycle in the graph is a triangle. A triangulated graph is said to be *triangularly connected* if for every pair of triangles T_s and T_t, there exists a sequence of triangles, $T_s = T_1, T_2, \ldots, T_l = T_t$, such that for every i, $1 \leq i < l - 1$, T_i and T_{i+1} are adjacent to each other (i.e., T_i and T_{i+1} share a common edge).

Lemma 4 *An extreme k-NN Delaunay component is triangularly connected*.

Proof If a triangulated subgraph is not triangularly connected, then there must exist a cut node. However, by Definition 5, there exits no cut node within an extreme k-NN Delaunay component. □

By Lemma 4, if one can find an interior k-NN Delaunay triangle of a k-NN Delaunay component, it is possible to traverse from it to all the other triangles by moving always from one triangle to an adjacent triangle.

4.3 Hull Construction

To compute the k-NN Delaunay hull, we first find an extreme k-NN Delaunay edge, then traverse from it to its adjacent extreme k-NN Delaunay edge, and repeatedly

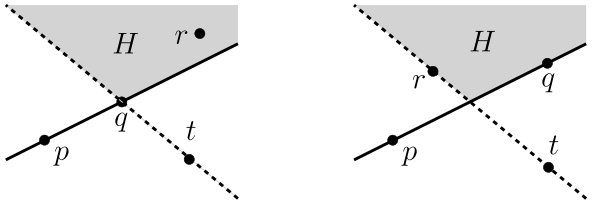

Fig. 3 Two adjacent extreme k-NN Delaunay edges, (H_1, H_2) and (H_2, H_3), in the Euclidean metric. An unbounded circle is a straight line in the Euclidean metric, and in this figure, the interior of each unbounded circle is the *upper half-plane*. (**a**) $H_1 = H \cup \{p, r\}$, $H_2 = H \cup \{q, r\}$, and $H_3 = H \cup \{q, t\}$. (**b**) $H_1 = H \cup \{p, r\}$, $H_2 = H \cup \{q, r\}$, and $H_3 = H \cup \{r, t\}$

perform the traversal operation until all the extreme k-NN Delaunay edges have been traversed. Lemma 5 implies that there must always exist at least one extreme k-NN Delaunay edge.

Lemma 5 *Consider a set S of point sites in R^2. There exist at least two valid 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 valid k-subset.*

Proof An unbounded circle passing through two sites is a straight line in the L_p metric, $1 < p < \infty$, and an L-shaped curve in the L_1 and the L_∞ metrics. Let H_1 be the first k sites as we scan sites from left to right. We first consider $1 < p < \infty$. Let $\mathrm{CH}(H_1)$ and $\mathrm{CH}(S \setminus H_1)$ be the convex hulls of H_1 and $S \setminus H_1$, respectively. Let L be an internal common tangent line between $\mathrm{CH}(H_1)$ and $\mathrm{CH}(S \setminus H_1)$, passing trough points $p \in \mathrm{CH}(H_1)$ and $q \in \mathrm{CH}(S \setminus H_1)$. Let H_2 be $(H_1 \cup \{q\}) \setminus \{p\}$. Then L is an unbounded circle that passes through H_1 and H_2 but does not pass through any other valid k-subset.

For the L_∞ metric, let q be the leftmost site of $S \setminus H_1$; let p be the topmost site of H_1, if it is higher than q, otherwise, let p be the bottommost site of H_1. Let H_2 be $(H_1 \cup \{q\}) \setminus \{p\}$. Then p and q define an axis-parallel L-shaped curve which passes through H_1 and H_2 but does not pass through any other valid k-subset. Equivalently for L_1. □

We describe a relation between two adjacent extreme k-NN Delaunay edges, which will lead to a k-NN hull construction in Sect. 5.1. 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 $|H| = k-2$, where $p, q, r \in S \setminus H$. 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 $t \in S \setminus (H \cup \{q, r\})$. In the former case, $|H_1 \cap H_2 \cap H_3| = k-1$, and in the latter case, $|H_1 \cap H_2 \cap H_3| = k-2$. For example, in Fig. 3, the solid line corresponds to the extreme k-NN Delaunay edge between H_1 and H_2, and the dashed line corresponds to the extreme k-NN Delaunay edge between H_2 and H_3. If $H_3 = H \cup \{r, t\}$, then there exists an unbounded circle that passes through q and t and contains $H \cup \{r\}$ but does not contain any other site. For example, in Fig. 3(a), the dashed line passes through q and t, and its upper half-plane only contains $H \cup \{r\}$. If $H_3 = H \cup \{q, t\}$, then there exists an unbounded circle

that passes through r and t and contains $H \cup \{q\}$ but does not contain any other site. For example, in Fig. 3(b) the dashed line passes through r and t, and its upper half-plane only contains $H \cup \{q\}$. To identify H_3 in the first case, 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. In 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.

Since an extreme k-NN Delaunay node in L_p, $1 < p < \infty$, corresponds to a k-set, an L_p, $1 < p < \infty$, k-NN hull corresponds to an ordered sequence of k-sets. In [17], the authors used geometric duality to transform all k-sets into a k-belt, and developed an $O(b \log^2 n)$ algorithm to compute the k-belt, where b is the size of the k-belt. Thus, we can use the algorithm of [17] to compute the k-NN hull in the Euclidean or more generally in the L_p, $1 < p < \infty$, metric. In addition, Brodal and Jacob [10] proposed a dynamic data structure for planar convex hulls in which a query takes $O(\log n)$ time and an insertion and a deletion take $O(\log n)$ amortized time. With the new dynamic data structure, the time complexity of the algorithm in [17] can improve to $O(b \log n)$. Therefore, the k-NN hull in the L_p metric, $1 < p < \infty$, can be computed in $O(b \log n)$ time.

The k-NN hull construction in the L_∞ metric will be given in Sect. 5.1.

4.4 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 [21].

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 Sect. 3, 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. The reason is that if 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 thus 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 thus 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 the extreme k-NN Delaunay edges are available after the k-NN Delaunay hull construction, we can use any extreme k-NN Delaunay edge to compute its incident k-NN Delaunay triangle and use it as a starting triangle.

To conclude, according to Sects. 4.3, 4.4, the time complexity of our paradigm is $O(\mathrm{HC} + m * \mathrm{TO})$, where HC is the time to compute the k-NN Delaunay hull, TO is the time complexity of one traverse operation, and m is the number of all k-NN Delaunay edges, i.e., the structural complexity of the k-NN Delaunay graph.

In Sect. 5, we will implement our traversal-based paradigm in the L_∞ metric, resulting in an output-sensitive algorithm that directly computes the planar L_∞ k-NN Delaunay graph in $O((n+m)\log n)$ time.

In the Euclidean metric, such a traversal-based operation, i.e., the circular wave movement which touches two sites and stops when hitting another site, can transformed into a ray-shooting query in three dimensions. For a point $p = (\sigma_1, \sigma_2) \in R^2$, let $h(p)$ be a hyperplane $x_3 = 2\sigma_1 x_1 + 2\sigma_2 x_2 - (\sigma_1^2 + \sigma_2^2)$ in R^3, and let $h(S)$ be $\{h(p) \mid \forall p \in S\}$. Now if a circle touches three sites, p, q, and r, in R^2, its center corresponds to the intersection vertex among $h(p)$, $h(q)$, and $h(r)$ in R^3, i.e., its center is the vertical projection of the intersection onto the plane $x_3 = 0$. Let l be the common line between $h(p)$ and $h(q)$, $p, q \in S$. Then the vertical projection of l onto $x_3 = 0$ is actually the bisector between p and q in R^2. Let c be the center of a circle touching p and q, and let c' be the vertical projection of c onto l. Under these circumstances, a circular wave movement whose center begins at c and which touches p and q first hits a site $r \in S \setminus \{p, q\}$ if and only if a ray shooting from c' along the corresponding direction of l first hits $h(r) \in h(S) \setminus \{h(p), h(q)\}$.

Agarwal and Matoušek [3] proposed a data structure that can answer such a ray shooting query in $O(n^\epsilon)$ time after $O(n \log n)$-time preprocessing. As noted in Sect. 4.3, the k-NN hull construction in the Euclidean metric takes $O(b \log n)$ time, where b is the number of extreme k-NN Delaunay edges. Since there are $O(k(n-k))$ Voronoi edges in $V_k(S)$, b is trivially $O(k(n-k))$, implying that the hull construction takes $O(k(n-k)\log n)$ time. Thus, using the data structure of Agarwal and Matoušek, our paradigm can be implemented in the Euclidean metric in $O(k(n-k)n^\epsilon)$ time.

4.5 Degenerate Cases

Definition 6 Given a set S of sites in R^2, a *k-NN Delaunay cycle*, denoted as $\mathrm{DC}(H_1, H_2, \ldots, H_l)$, where $3 \le l \le n$, is a chordless cycle of the k-NN Delaunay graph, which connects the l k-NN Delaunay nodes, $H_1, H_2, \ldots,$ and H_l, using k-NN Delaunay edges, where H_i and H_{i+1} form a k-NN Delaunay edge for $1 \le i \le l$ and $H_{l+1} = H_1$. A k-NN Delaunay cycle is associated with $H_1, H_2, \ldots,$ and H_l.

If we remove the general position assumption, the dual of a k-NN Voronoi vertex becomes a *k-NN Delaunay cycle*, and Lemma 2 generalizes to Lemma 6 in a straightforward way.

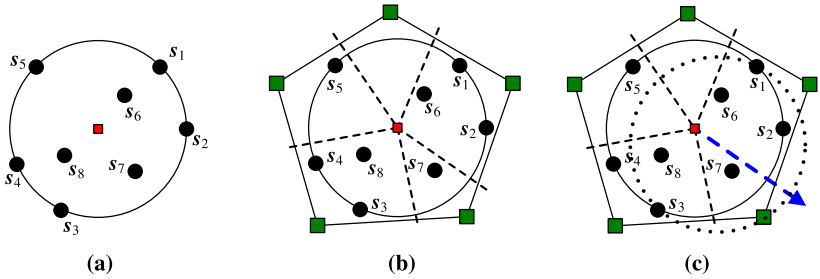

Fig. 4 (**a**) A *circle* passing through five sites (s_1, s_2, s_3, s_4, and s_5) and containing three sites (s_6, s_7, and s_8). (**b**) A 5-NN Delaunay cycle associated with $H_1 = \{s_1, s_2, s_6, s_7, s_8\}$, $H_2 = \{s_2, s_3, s_6, s_7, s_8\}$, $H_3 = \{s_3, s_4, s_6, s_7, s_8\}$, $H_4 = \{s_4, s_5, s_6, s_7, s_8\}$, 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)

Lemma 6 *Given a set S of point sites $\in R^2$, l k-NN Delaunay nodes, $H_1, H_2, \ldots,$ and H_l, form a k-NN Delaunay cycle $(H_1, H_2, \ldots, H_l)$ 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 $c_1 \in H_1 \setminus H_2$, $c_2 \in H_2 \setminus H_3, \ldots,$ 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.*

Figure 4 illustrates Lemma 6. Figure 4(a) shows a circle passing through five sites and containing three sites. According to Lemma 6, the circle corresponds to a k-NN Delaunay cycle, $4 \leq k \leq 7$. Figure 4(b) shows a 5-NN Delaunay cycle. As illustrated in Fig. 4(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.

5 Planar Algorithm in the L_∞ Metric

We implement our paradigm in the L_∞ metric such that the hull construction takes $O(n \log n)$ time (Sect. 5.1) and each traversal operation between two triangles takes $O(\log n)$ time (Sect. 5.2). 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.

5.1 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

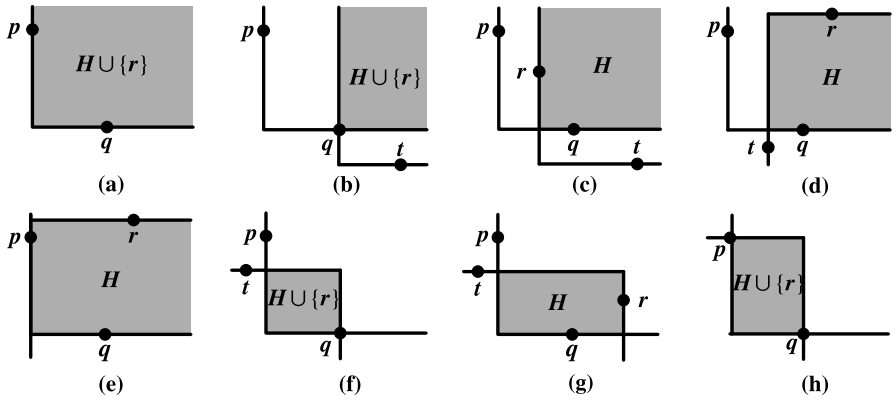

Fig. 5 $e_1 = (H_1, H_2)$ and $e_2 = (H_2, H_3)$, where e_1 is NE. (**a**) $H_1 = H \cup \{p, r\}$, and $H_2 = H \cup \{q, r\}$. (**b**)–(**c**) e_2 is NE. (**b**) $H_3 = H \cup \{r, t\}$. (**c**) $H_3 = H \cup \{q, t\}$. (**d**)–(**e**) e_2 is SE. (**d**) $H_3 = H \cup \{q, t\}$. (**e**) $H_3 = H \cup \{p, q\}$. (**f**)–(**h**) e_2 is SW. (**f**) $H_3 = H \cup \{r, t\}$. (**g**) $H_3 = H \cup \{q, t\}$. (**h**) $e_1 = e_2$

two portions, where one portion is a *quarter-plane*, illustrated shaded in Fig. 5. Thus, an extreme k-NN Delaunay edge (H_1, H_2) corresponds to an L-shaped curve, which passes through exactly two sites, $p, q \in H_1 \oplus H_2$, and whose quarter-plane exactly contains in its interior the $k-1$ sites in $H_1 \cap H_2$. All the extreme k-NN Delaunay edges can be classified into four categories, {NE, SE, SW, NW}, according to the orientation of their corresponding quarter-plane.

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 Figs. 5(b) and 5(c). Actually, these two cases correspond to $|H_1 \cap H_2 \cap H_3| = k-1$ and $|H_1 \cap H_2 \cap H_3| = k-2$, respectively, and r is implicit for the former case. 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 small positive value, less than the minimum distance among the given sites. For each case of e_2, we first analyze the situation, and then explain the operation.

1. e_2 is **NE** (Fig. 5(b)–(c)):

   If H_3 is $H \cup \{r, t\}$, the corresponding NE L-shaped curve of e_2 passes through q in its north ray and t in its east ray. Under these circumstances, for each site $v \in H_1 \cap H_2$, $x_v > x_q$, and $y_v > y_t$. If H_3 is $H \cup \{q, t\}$, the corresponding NE L-shaped curve of e_2 passes through r in its north ray and t in its east ray, where $x_r \leq x_v$, $v \in H_1 \cap H_2$. Under these circumstances, for each site $v \in H_2 \setminus \{r\}$, $x_v > x_r$, and $y_v > y_t$.

   In order to compute e_2, 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. 5(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 $H \cup \{r\}$. If H_3 is $H \cup \{q, t\}$, the corresponding SE L-shaped curve of e_2 passes through r in its east ray and t in its south ray, where $y_r \geq y_v$, $v \in H_1 \cap H_2$. Under these circumstances, for each site $v \in H_1 \cap H_2$, $x_v > x_t$, and $y_v < y_r$.

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.

At 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. Figure 5(d)–(e) shows the two cases $t \neq p$ and $t = p$.

3. e_2 is **SW** (Fig. 5(f)–(h)):

   If H_3 is $H \cup \{r, t\}$, the corresponding SW L-shaped curve of e_2 passes through q in its south ray and t in its west ray. Under these circumstances, for each site $v \in H_1 \cap H_2$, $x_v < x_q$, and $y_v < y_t$. If H_3 is $H \cup \{q, t\}$, the corresponding NE L-shaped curve of e_2 passes through r in its south ray and t in its west ray, where $x_r \geq x_v$, $v \in H_1 \cap H_2$. Under these circumstances, for each site $v \in H_2 \setminus \{r\}$, $x_v < x_r$, and $y_v < y_t$.

   As a result, 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 [12] 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 {NE, SE, SW, 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.

5.2 L_∞ k-NN Traversal Operation among Triangles

As mentioned in Sect. 4.4, 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. 6. Therefore, a square wave propagation along an L_∞ bisector would consist of three stages: square contraction, square

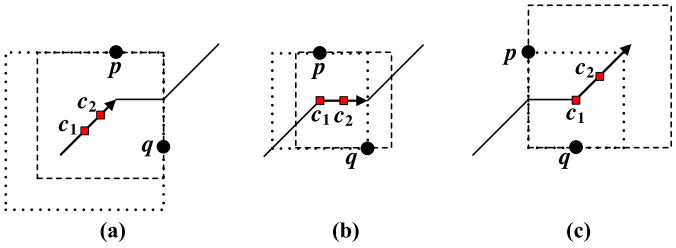

Fig. 6 Square wave propagation along $B(p,q)$. *Solid segments* are $B(p,q)$, *arrow heads* are moving directions, and c_1 and c_2 are the centers of *dotted squares* and *dashed squares*, respectively. (**a**) square contraction. (**b**) square movement. (**c**) square expansion

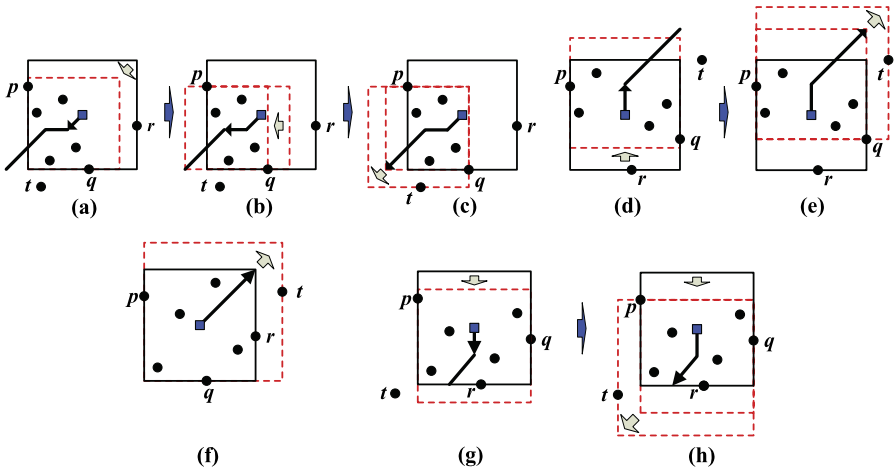

Fig. 7 Square wave propagation for $T(p,q,r)$. (**a**)–(**e**) $T(p,q,r)$ is new. (**a**)–(**c**) p and q are on adjacent sides. (**d**)–(**e**) p and q are on opposite sides. (**f**)–(**h**) $T(p,q,r)$ is old. (**f**) p and q are on adjacent sides. (**g**)–(**h**) p and q are on opposite sides

movement, and square expansion. As shown in Fig. 6, square contraction occurs when the center of the square wave moves along the 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.

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. 7(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

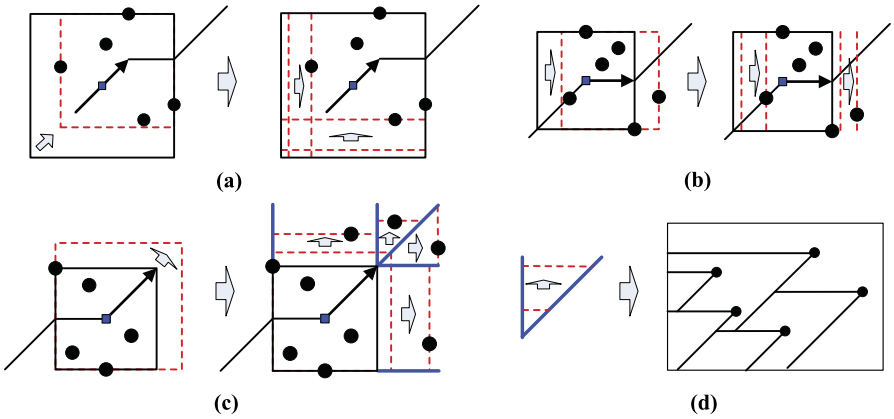

Fig. 8 Square wave propagation and segment-dragging queries. (**a**) square contraction. (**b**) square movement. (**c**) square expansion. (**d**) the second class of segment-dragging query and the point location query

located on parallel sides, as shown in Fig. 7(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. 7(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. 7(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. 8(a)). Square movement is similar to square contraction, but the two dragged segments are parallel to each other (see Fig. 8(b)). Since an orthogonal segment-dragging query can be computed in $O(\log n)$ time after $O(n \log n)$-time preprocessing [12], 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. 8(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 [23], 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. 8(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.

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.

6 Structural Complexity of the L_∞ k-NN Voronoi Diagram

Lee [21] derived a formula for the number N_k of Voronoi regions of $V_k(S)$ in the L_2 metric, which is valid in the general L_p metric, $1 \leq p \leq \infty$,

$$N_k = (2k-1)n - \left(k^2-1\right) - \sum_{i=1}^{k-1} \mathcal{S}_i,$$

where $\mathcal{S}_i$ is the number of unbounded regions of $V_i(S)$.

For $k < \lceil \frac{n}{2} \rceil$, this formula directly leads to an $O(k(n-k))$ upper bound on the structural complexity of $V_k(S)$, which is valid in the general L_p metric, $1 \leq p \leq \infty$. For $k \geq \lceil \frac{n}{2} \rceil$, however, the upper bound depends on $\sum_{i=1}^{k} \mathcal{S}_{i-1}$. In the Euclidean metric, $\sum_{i=1}^{k} \mathcal{S}_i \leq kn$ for $k < \lceil \frac{n}{2} \rceil$, as shown in [5], $\sum_{i=1}^{n-1} \mathcal{S}_i = n(n-1)$, as shown in [30], and $\mathcal{S}_i = \mathcal{S}_{n-i}$, as a straight line (an unbounded circle) which partitions S into i sites and $n-i$ sites corresponds to both a valid i-subset and a valid $(n-i)$-subset. These equations are identical in the L_p metric for $1 < p < \infty$, as an unbounded circle is always a straight line in this case [20]. Combining these equations, we derive $\sum_{i=1}^{k} \mathcal{S}_i \geq kn$ for $\lceil \frac{n}{2} \rceil \leq k \leq n-1$. Thus, the formula of [21] leads to an $O(k(n-k))$ upper bound for the k-NN Voronoi diagram in the L_p metric, $1 < p < \infty$.

However, there is no existing result for the value of $\sum_{i=1}^{k} \mathcal{S}_i$ in the L_∞ or L_1 metric, and thus the structural complexity of the L_∞ (resp. L_1) k-NN Voronoi diagram is not directly derived by the formula in [21], except for $k < \lceil \frac{n}{2} \rceil$. Furthermore, the bound $O(k(n-k))$ is not tight for large k in these metrics, e.g., the complexity of the $(n-1)$-NN L_∞ Voronoi diagram is only $O(1)$. In this section, we use the Hanan grid [19] to bound 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, under the general position assumption, the L_∞ circumcircle of a k-NN Delaunay triangle is a unique square, called *k-NN Delaunay square*, that passes through three sites and contains either $k-1$ (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 7 *In the L_∞ metric, a k-NN Delaunay square must have at least two corners on the Hanan grid.*

Proof Given a k-NN Delaunay triangle, its k-NN Delaunay square may be derived in four possible ways as shown in Fig. 9. In Fig. 9(a), the bottom-left and bottom-right

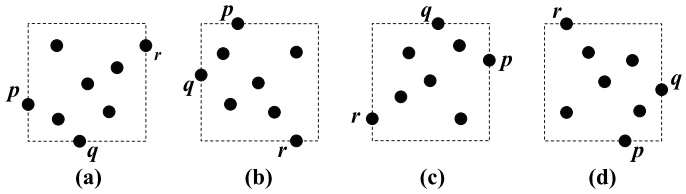

Fig. 9 Four classes of k-NN Delaunay square

corners are (x_p, y_q) and (x_r, y_q), respectively, which are both vertices of the Hanan gird for S. Similarly for each of the remaining three cases of Fig. 9. □

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 7, 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. □

Since Lee's formula [21] directly leads to an upper bound $O(k(n-k))$ in the general L_p metric, $1 \le p \le \infty$, for $k \le n/2$, and Theorem 2 leads to a bound $O((n-k)^2)$ in the L_∞ metric for $k > n/2$, we have

$$\begin{cases} O(k(n-k)), & \text{if } k \le n/2 \\ O((n-k)^2), & \text{if } k > n/2. \end{cases}$$

Since $(n-k)^2 < k(n-k)$ for $k > n/2$, we conclude.

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

7 Conclusion

We introduced a new traversal-based paradigm for the direct construction of the k-NN-Voronoi diagram in the plane, which led to an $O((n+m)\log n)$-time algorithm for the L_∞ k-NN Voronoi diagram, where m is the size of the diagram, using segment-dragging queries. This bound is output-sensitive and it considerably improves the time complexities of previous methods valid in the Euclidean plane, such as $O(nk^2 \log n)$ [21] and $O(n^2 \log n + k(n-k)\log^2 n)$ or $O(n^2 + k(n-k)\log^2 n)$ [13]. Besides, the efficient randomized approaches of [2, 11, 28] in the Euclidean metric are not directly applicable to the L_∞ metric. The efficient implementation of our paradigm in the L_2 metric remains an open problem.

References

1. Abellanas, M., Bose, P., Garcia, J., Hurtado, F., Nicolas, C.M., Ramos, P.A.: On structural and graph theoretic properties of higher order Delaunay graphs. Int. J. Comput. Geom. Appl. **19**(6), 595–615 (2009)
2. Agarwal, P.K., de Berg, M., Matoušek, J., Schwarzkopf, I.: Constructing levels in arrangements and higher order Voronoi diagrams. SIAM J. Comput. **27**(3), 654–667 (1998)
3. Agarwal, P.K., Matoušek, J.: Dynamic half-space range reporting and its applications. Algorithmica **13**, 325–345 (1995)
4. Aggarwal, A., Guibas, L.J., Saxe, J., Shor, P.W.: A linear-time algorithm for computing Voronoi diagram of a convex polygon. Discrete Comput. Geom. **4**(1), 591–604 (1984)
5. Alon, N., Györi, E.: The number of small semispaces of a finite set of points in the plane. J. Comb. Theory, Ser. A **41**, 154–157 (1986)
6. Aurenhammer, F.: A new duality result concerning Voronoi diagrams. Discrete Comput. Geom. **5**(1), 243–254 (1990)
7. Aurenhammer, F., Klein, R.: Voronoi diagrams. In: Handbook of Computational Geometry, pp. 201–290. Elsevier, Amsterdam (2000)
8. Aurenhammer, F., Schwarzkopf, O.: A simple on-line randomized incremental algorithm for computing higher order Voronoi diagrams. Int. J. Comput. Geom. Appl. **2**(4), 363–381 (1992)
9. Boissonnat, J.D., Devillers, O., Teillaud, M.: A semidynamic construction of higher-order Voronoi diagrams and its randomized analysis. Algorithmica **9**, 329–356 (1993)
10. Brodal, G.S., Jacob, R.: Dynamic planar convex hull. In: Proceeding of IEEE Symposium on Foundations of Computer Science, pp. 617–626 (2002)
11. Chan, T.M.: Random sampling, halfspace range reporting, and construction of ($\leq k$)-levels in three dimensions. SIAM J. Comput. **30**(2), 561–572 (1998)
12. Chazelle, B.: An algorithm for segment dragging and its implementation. Algorithmica **3**, 205–221 (1988)
13. Chazelle, B., Edelsbrunner, H.: An improved algorithm for constructing kth-order Voronoi diagram. IEEE Trans. Comput. **36**(11), 1349–1454 (1987)
14. Clarkson, K.L.: New applications of random sampling in computational geometry. Discrete Comput. Geom. **2**(1), 195–222 (1987)
15. Clarkson, K.L., Shor, P.W.: Applications of random sampling in computational geometry, II. Discrete Comput. Geom. **4**(1), 387–421 (1989)
16. Edelsbrunner, H., O'Rourke, J., Seidel, R.: Constructing arrangements of lines and hyperplanes with applications. SIAM J. Comput. **15**(2), 341–363 (1986)
17. Edelsbrunner, H., Welzl, E.: Constructing belts of two-dimensional arrangements with applications. SIAM J. Comput. **15**(1), 271–284 (1986)
18. Gudmundsson, J., Hammar, M., van Kreveld, M.: Higher order Delaunay triangulations. Comput. Geom. Theory Appl. **23**(1), 85–98 (2002)
19. Hanan, M.: On Steiner's problem with rectilinear distance. SIAM J. Appl. Math. **14**, 255–265 (1966)
20. Lee, D.T.: Two-dimensional Voronoi diagrams in the L_p-metric. J. ACM **27**(4), 604–618 (1980)
21. Lee, D.T.: On k-nearest neighbor Voronoi diagrams in the plane. IEEE Trans. Comput. **31**(6), 478–487 (1982)
22. Liu, C.-H., Papadopoulou, E., Lee, D.T.: An output-sensitive approach for the L_1/L_∞ k-nearest-neighbor Voronoi diagram. In: Proceeding of European Symposium on Algorithms, pp. 70–81 (2011)
23. Mitchell, J.S.B.: L_1 shortest paths among polygonal obstacles in the plane. Algorithmica **8**, 55–88 (1992)
24. Mulmuley, K.: On levels in arrangements and Voronoi diagrams. Discrete Comput. Geom. **6**(1), 307–338 (1991)
25. Papadopoulou, E.: Critical area computation for missing material defects in VLSI circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **20**(5), 583–597 (2001)
26. Papadopoulou, E.: Net-aware critical area extraction for opens in VLSI circuits via higher-order Voronoi diagrams. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **30**(5), 704–716 (2011)
27. Papadopoulou, E., Lee, D.-T.: The L_∞ Voronoi diagram of segments and VLSI applications. Int. J. Comput. Geom. Appl. **11**(5), 503–528 (2001)

28. Ramos, E.: On range reporting, ray shooting, and k-level construction. In: Proceeding of ACM Symposium on Computational Geometry, pp. 390–399 (1999)
29. Schmitt, D., Spehner, J.C.: Order-k Voronoi diagrams, k-sections, and k-sets. In: Proceeding of Japanese Conference on Discrete and Computational Geometry. LNCS, vol. 1763, pp. 290–304 (1998)
30. Shamos, M., Hoey, D.: Closest point problems. In: Proceeding of IEEE Symposium on Foundations of Computer Science, pp. 151–162 (1975)