

# On the Hausdorff Voronoi Diagram of Point Clusters in the Plane

Evanthia Papadopoulou

IBM TJ Watson Research Center Yorktown Heights, NY 10598, USA  
evanthia@watson.ibm.com

**Abstract.** We study the *Hausdorff Voronoi diagram* of point clusters in the plane and derive a tight combinatorial bound on its structural complexity. We present a plane sweep algorithm for the construction of this diagram improving upon previous results. Motivation for the investigation of this type of Voronoi diagram comes from the problem of computing the *critical area* of a VLSI Layout, a measure reflecting the sensitivity of the design to spot defects during manufacturing.

## 1 Introduction

Given a set  $S$  of point clusters in the plane their *Hausdorff Voronoi diagram* is a subdivision of the plane into regions such that the Hausdorff Voronoi region of a cluster  $P \in S$  is the locus of points closer to  $P$ , according to the Hausdorff distance<sup>1</sup>, than to any other cluster in  $S$ . As it was shown in [9], the Hausdorff Voronoi region of  $P$  can be defined equivalently as the locus of points  $t$  whose maximum distance from any point of  $P$  is less than the maximum distance of  $t$  from any other cluster in  $S$ . The Hausdorff Voronoi region of  $P$  is subdivided into finer regions by the farthest-point Voronoi diagram of  $P$ . This structure generalizes both the ordinary Voronoi diagram of points and the farthest-point Voronoi diagram. It is equivalent to the ordinary Voronoi diagram of points if clusters degenerate to single points and to the farthest-point Voronoi diagram if  $S$  consists of a single cluster.

The Hausdorff Voronoi diagram has appeared in the literature under different names, defined in terms of the maximum distance and not in terms of the Hausdorff distance, and motivated by independent problems. In [3] it was termed the *Voronoi diagram of point clusters*, in [1] the *closest covered set diagram*, and in [7,9] the *min-max Voronoi diagram*. The equivalence to the Voronoi diagram under the Hausdorff metric was shown in [9]. In [3] combinatorial bounds regarding this diagram were derived by means of envelopes in three dimensions. It was shown that the size of this diagram is  $O(n^2\alpha(n))$  for arbitrary clusters of points, and  $O(n)$  for clusters of points with disjoint convex hulls, where  $n$  is the total number of points on the convex hulls of individual clusters in  $S$ . The latter

---

<sup>1</sup> The (directed) Hausdorff distance from set  $A$  to  $B$  is  $h(A, B) = \{\max_{a \in A} \min_{b \in B} d(a, b)\}$ . The Hausdorff distance between  $A$  and  $B$  is  $d_h(A, B) = \max\{h(A, B), h(B, A)\}$ .

was also shown in [1] for disjoint convex shapes and arbitrary convex distance functions. An  $\Omega(n^2)$  example of  $n$  intersecting segments was given in [3]. In [9] a tighter combinatorial bound on the size of the diagram was given and the linearity property was extended to the more general class of *non-crossing* clusters (see Def. 1). An  $O(n^2\alpha(n))$ -time algorithm for the construction of this diagram was given in [3] by applying a divide and conquer technique for envelopes of piece-wise linear functions in three dimensions. This time complexity was automatically improved to  $O(n^2)$  by the tighter combinatorial bound given in [9]. In [1] the problem for disjoint convex sets was reduced to abstract Voronoi diagrams and the randomized incremental construction of [5] was proposed for the computation of the diagram. This approach resulted in an  $O(kn \log n)$ -time algorithm, where  $k$  is the time to construct the Hausdorff bisector of two convex polygons. In [9] a direct divide and conquer algorithm was given for the construction of the Hausdorff Voronoi diagram of time complexity  $O((n + M + N + K) \log n)$ , where  $M$  was the number of pairs of points on the convex hull of *crossing* clusters,  $N$  was  $O(n \log n)$ , and  $K$  was the total number of points of clusters entirely enclosed in the minimum enclosing circle of some  $P \in S$ . In [7] the simpler  $L_\infty$  version of the problem was investigated and a simple plane sweep algorithm of time complexity  $O((n + K) \log n)$  was given for the non-crossing case.

In this paper we derive a tight combinatorial bound on the structural complexity of the Hausdorff Voronoi diagram. In particular we show that the size of the diagram is  $O(n + m)$ , where  $m$  is the number of *crucial* supporting segments among pairs of crossing clusters (see Def. 2). Crucial supporting segments are entirely enclosed in the minimum enclosing circle of one of the clusters. We also present a simple plane sweep algorithm for the construction of the Hausdorff Voronoi diagram of time complexity  $O(M + (n + m + K) \log n)$ , where  $K = \sum_{P \in S} K(P)$ ,  $K(P)$  is the number of clusters entirely enclosed in the *anchor circle* of  $P$ ,  $M = \sum_{P \in S} M(P)$  and  $M(P)$  is the number of points  $q \in Q$  enclosed in the *anchor circle* of  $P$  such that either  $Q$  is entirely enclosed in the *anchor circle* of  $P$  or  $Q$  is crossing with  $P$ . The *anchor circle* of  $P$  is a specially defined enclosing circle (see Def. 4), generally larger than the minimum enclosing circle of  $P$ . This algorithm improves the time complexity of previous results, generalizes the plane sweep construction for Voronoi diagrams, and remains simple.

Our motivation for studying the Hausdorff Voronoi diagram comes from an application in VLSI manufacturing as explained in [7,9], in particular *critical area* extraction for predicting the *yield* of a VLSI chip. The *critical area* is a measure reflecting the sensitivity of a VLSI design to manufacturing defects due to dust particles and other contaminants on materials and equipment. In [7,9] the critical area computation problem for *via-blocks* was shown to be reducible to the Hausdorff Voronoi diagram (termed the min-max Voronoi diagram). Via-blocks represent the 2nd most important defect mechanism (after shorts) for yield loss. For more details on the critical area computation problem and its connection to the Hausdorff Voronoi diagram see e.g. [6,7,8,9,11]. Plane sweep is our method of choice for this problem because of the very large data volume of VLSI designs. The advantage of plane sweep is that we never need to keep the entire Voronoi diagram in memory. Instead we only keep the *wavefront*, the portion of the

Voronoi diagram bounding the sweep-line. As soon as a Voronoi cell is computed, critical area computation can be performed independently within that cell, and the cell can be immediately discarded.

In the following, due to lack of space, we skip proofs that are easy to derive.

## 2 Preliminaries and Definitions

The *farthest distance* between two sets of points  $A, B$  is  $d_f(A, B) = \max\{d(a, b), a \in A, b \in B\}$ , where  $d(a, b)$  denotes the ordinary distance between two points  $a, b$ . The (directed) Hausdorff distance from  $A$  to  $B$ ,  $h(A, B) = \{\max_{a \in A} \min_{b \in B} d(a, b)\}$ . The (undirected) Hausdorff distance between  $A$  and  $B$  is  $d_h(A, B) = \max\{h(A, B), h(B, A)\}$ . The Hausdorff bisector between  $A$  and  $B$  is  $b_h(A, B) = \{y \mid d_h(y, A) = d_h(y, B)\}$  and the farthest bisector is  $b_f(A, B) = \{y \mid d_f(y, A) = d_f(y, B)\}$ . As it was shown in [9] the Hausdorff bisector and the farthest bisector are equivalent. In the following we simply use the generic term *inter-bisector* to denote both.

The *farthest* Voronoi region of point  $p_i \in P$  is  $freg(p_i) = \{x \mid d(x, p_i) > d(x, p_j), p_j \in P\}$  and the *farthest* Voronoi diagram of a set of points  $P$  is denoted as  $f-Vor(P)$ . It is well known (see e.g. [10]) that  $f-Vor(P)$  consists of unbounded convex regions, one for each point on the convex hull of  $P$ . The bisectors of  $f-Vor(P)$  are portions of ordinary bisectors, denoted  $b(p_i, p_j), p_i, p_j \in P$ , and they are called *intra-bisectors*. The convex hull of  $P$  is denoted as  $CH(P)$ . A segment  $\overline{p_i p_j}$  connecting any two points on  $CH(P)$  is called a *chord*.

The tree structure of  $f-Vor(P)$  is called the *intra-bisector tree* of  $P$  and it is denoted as  $T(P)$ .  $T(P)$  is assumed to be rooted at an arbitrary point  $y_0 \in T(P)$ . Every point  $y \in T(P)$  is weighted by  $d_f(y, P)$ , the radius of the smallest circle centered at  $y$  entirely enclosing  $P$ . The circle centered at  $y$  of radius  $d_f(y, P)$  is called a  $P$ -circle and it is denoted as  $\mathcal{K}_y$ .  $\mathcal{K}_y$  passes through  $p_i, p_j \in P$  such that  $y \in b(p_i, p_j)$ . Point  $y$  partitions  $T(P)$  in two parts:  $T(y)$  and  $T_c(y)$ , where  $T(y)$  consists of all descendents of  $y$  in the rooted  $T(P)$  including  $y$ , and  $T_c(y)$  is the complement of  $T(y)$ . If  $y$  is a vertex of  $T(P)$  and an incident intra-bisector segment  $\overline{y y_j}$  is explicitly specified to contain  $y$  then  $T(y)$  consists of the subtree rooted at  $y$  containing segment  $\overline{y y_j}$ ;  $T_c(y)$  is still the complement of  $T(y)$ . For figures and more details see [9]. Chord  $\overline{p_i p_j}$  ( $y \in b(p_i, p_j)$ ) partitions  $\mathcal{K}_y$  in two parts:  $\mathcal{K}_y^r$ , referred to as the *rear* portion of  $\mathcal{K}_y$ , enclosing the points of  $CH(P)$  that induce  $T(y)$ , and  $\mathcal{K}_y^f$ , referred to as the *forward* portion, enclosing the points of  $CH(P)$  inducing  $T_c(y)$ . The characterization of the portions of  $\mathcal{K}_y$  as *rear* or *forward* depends on the root of  $T(P)$  and can be reversed for a different choice of the root. Any point  $q$  in  $\mathcal{K}_y^r$  is called *rear* and any point in  $\mathcal{K}_y^f$  is called *forward*, with respect to intra-bisector point  $y, b(p_i, p_j)$ , and the root of  $T(P)$ . The following observation is a generalization of one given in [9] and is used throughout the paper.

**Lemma 1.** *For any point  $y \in T(P)$ , such that  $y \in b(p_i, p_j)$ , the following holds: For any  $y_j \in T(y)$ ,  $\mathcal{K}_{y_j}^f \subset \mathcal{K}_y^f$  and  $\mathcal{K}_{y_j}^r \subset \mathcal{K}_y^r$ . For any  $y_k \in T_c(y)$ ,  $\mathcal{K}_{y_k}^r \subset \mathcal{K}_y^r$ . The observation is valid for any root of  $T(P)$ .*

**Definition 1.** Two chords  $\overline{q_i q_j} \in Q$  and  $\overline{p_i p_j} \in P$  are called crossing iff they intersect and all  $p_i, p_j, q_i, q_j$  are points of the convex hull of  $P \cup Q$ ; otherwise they are called non-crossing. Cluster  $Q$  is said to be crossing chord  $\overline{p_i p_j} \in P$  iff there is a chord  $\overline{q_i q_j} \in Q$  that is crossing  $\overline{p_i p_j}$ ; otherwise  $Q$  is said to be non-crossing with  $\overline{p_i p_j}$ . Two clusters  $P, Q$  are called non-crossing iff their convex hulls admit at most two supporting segments.

Let  $S$  be a set of point clusters. The Hausdorff Voronoi region of cluster  $P \in S$  is  $hreg(P) = \{x \mid d_f(x, P) < d_f(x, Q), \forall Q \in S\}$  and it may be disconnected.  $hreg(P)$  is further partitioned into finer regions by  $f\text{-Vor}(P)$ . For any point  $p \in P$ ,  $hreg(p) = \{x \mid d(x, p) = d_f(x, P) < d_f(x, Q), \forall Q \in S\}$ . The collection of all (non-empty) Hausdorff Voronoi regions defined by  $S$ , together with their bounding edges and vertices, is called the Hausdorff Voronoi diagram of  $S$ . The bounding edges of  $hreg(P)$  consist of portions of *inter-bisectors* and the inner edges consist of portions of *intra-bisectors* among the points on the convex hull of  $P$ . The vertices are classified into three types: *inter-vertices* where at least three inter-bisectors meet, *intra-vertices* where at least three intra-bisectors meet, and *mixed-vertices* where at least one intra-bisector and two inter-bisectors meet.

By definition, a mixed Voronoi vertex  $v$  is the center of a  $P$ -circle  $\mathcal{K}_v$  passing through  $p_i, p_j \in P$  and  $q_i \in Q$ , entirely enclosing both clusters  $P$  and  $Q$  and not containing any other cluster in  $S$ . Clearly  $v$  is a vertex of the intra-bisector  $b_h(P, Q)$  that is incident to  $T(P)$ . Since  $b_h(P, Q)$  is a subgraph of  $f\text{-Vor}(P \cup Q)$  [9], any vertex of the intra-bisector  $b_h(P, Q)$  is a mixed Voronoi vertex of  $H\text{-Vor}(\{P, Q\})$  and a candidate for a mixed Voronoi vertex of  $H\text{-Vor}(S)$ . Vertex  $v$  is characterized as *crossing* (resp. *non-crossing*) iff  $Q$  is crossing (resp. *non-crossing*) with  $\overline{p_i p_j}$ . Furthermore,  $v$  is characterized as *rear* (resp. *forward*) iff  $q_i \in \mathcal{K}_v^r$  (resp.  $q_i \in \mathcal{K}_v^f$ ). The characterization of a mixed vertex as rear or forward depends on the choice of the root for the intra-bisector tree.

### 3 Structural Complexity

In this section we give a tight bound on the structural complexity of the Hausdorff Voronoi diagram.

**Lemma 2.** Consider the mixed vertices induced on  $T(P)$  by  $b_h(P, Q)$ . We have the following properties:

- For any rear non-crossing vertex  $v$ ,  $T_c(v) \cap hreg(P) = \emptyset$ . Thus,  $T(P)$  can contain at most one rear non-crossing mixed Voronoi vertex.
- For any forward non-crossing vertex  $v$ ,  $T(v) \cap hreg(P) = \emptyset$ . Thus,  $T(P)$  can contain at most  $|T(P)|$  forward non-crossing mixed Voronoi vertices, at most one for each unbounded segment of  $T(P)$ .
- Any crossing forward vertex on  $T(P)$  must be followed by a crossing rear vertex (considering only vertices of  $b_h(P, Q)$ ).
- Any rear (resp. forward) vertex delimits the beginning (resp. ending) of a component of  $T(P) \cap hreg(P)$  as we traverse  $T(P)$  from the root to the leaves.

*Proof.* For a rear non-crossing  $v$ ,  $Q \in \mathcal{K}_v^r \cup CH(P)$  and for a forward non-crossing  $v$ ,  $Q \in \mathcal{K}_v^f \cup CH(P)$ . The first two statements are easy to derive by Lemma 1.

Consider a path from the root to a leaf of  $T(P)$  that contains vertices of  $b_f(P, Q)$ . Let  $y_1$  (resp.  $y_k$ ) be the first (resp. last) vertex along this path that is contained in  $hreg(P)$ .  $y_1$  may be the root of  $T(P)$  (if  $y_0 \in hreg(P)$ ) and  $y_k$  may extend to infinity. Because of the first two statements of this Lemma, any vertex between  $y_1$  and  $y_k$ , except  $y_1, y_k$ , must be crossing. Let  $v$  be such a vertex between  $y_1$  and  $y_k$  and let  $p_i, p_j \in P$ ,  $q_i \in Q$  be the points inducing  $v$ . If  $v$  is delimiting the ending of  $hreg(P)$  and the beginning of  $hreg(Q)$ , as we walk from  $y_1$  to  $y_k$ , then  $q_i$  must be part of  $\mathcal{K}_v^f$ , that is  $v$  must be forward. On the contrary, if  $v$  is delimiting the beginning of  $hreg(P)$  after  $hreg(Q)$ ,  $v$  must be in  $\mathcal{K}_v^r$  i.e.,  $v$  must be rear. Thus, all the vertices between  $y_1$  and  $y_k$  must be crossing and must be alternating between forward and rear starting with a forward. Since  $y_k$  cannot be rear, any crossing forward mixed vertex on the path must be followed by a rear crossing mixed vertex. This also derives the last statement.  $\square$

Let  $v$  be a mixed vertex of  $b_h(P, Q)$  induced by  $(q_i, p_i, p_j)$ ,  $q_i \in Q, p_i, p_j \in P$ . Clearly  $q_i, p_i, p_j$  are points of  $CH(P \cup Q)$ . We say that vertex  $v$  can be attributed to the first pair of supporting segments encountered as we traverse  $CH(P \cup Q)$  from  $q_i$  to  $p_i$  and  $p_j$ . We have the following property.

**Lemma 3.** *Every rear vertex of  $b_h(P, Q)$  on  $T(P)$  can be attributed to a unique pair of supporting segments between  $CH(P)$  and  $CH(Q)$  that are entirely enclosed in the  $P$ -circle centered at the root of  $T(P)$ .*

*Proof.* Let  $v_i$  be a rear vertex of  $b_h(P, Q)$  induced by  $(q_i, p_i, p_j)$ ,  $q_i \in Q, p_i, p_j \in P$ , that can be attributed to the pair of supporting segments  $(s_1, s_2)$ . Since  $v_i$  is rear,  $s_1, s_2 \in \mathcal{K}_{v_i}^r$ . By Lemma 1,  $\mathcal{K}_{v_i}^r \subset \mathcal{K}_{y_0}$ , where  $y_0$  is the root of  $T(P)$ , and thus,  $s_1, s_2 \in \mathcal{K}_{y_0}$ . Let  $v_j$  be another vertex of  $b_h(P, Q)$  induced by a triplet of points  $(q_j, p'_i, p'_j)$ ,  $q_j \in Q, p'_i, p'_j \in P$ . The only way for  $\mathcal{K}_{v_j}$  to enclose both  $P$  and  $Q$  is for all three  $q_j, p'_i, p'_j$  be enclosed in the same slice of  $\mathcal{K}_{v_i}$  as defined by triangle  $(q_i, p_i, p_j)$ . But then either  $q_j \in \mathcal{K}_{v_j}^f$  i.e.,  $v_j$  is forward, or  $q_j$  must be part of a different component of  $CH(P \cup Q)$  than  $q_i$ , i.e.,  $v_j$  is attributed to a pair of supporting segments other than  $(s_1, s_2)$ .  $\square$

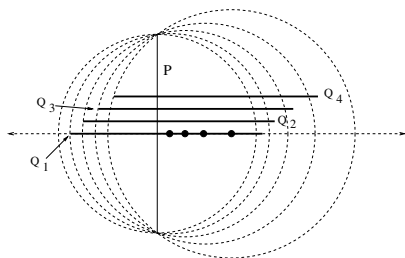
**Lemma 4.** *If there is a  $P$ -circle through  $p_i, p_j \in P$  that encloses  $Q$  then there can be no  $Q$ -circle through  $q_i, q_j \in Q$  that encloses  $P$  such that  $\overline{q_i q_j}$  and  $\overline{p_i p_j}$  are crossing.*

**Definition 2.** *Let  $P, Q$  be a pair of crossing clusters and let  $(s_1, s_2)$  be a pair of supporting segments between  $CH(P)$  and  $CH(Q)$  enclosed in the minimum enclosing circle of  $P$  such that there is a vertex of  $b_h(P, Q)$  that can be attributed to  $(s_1, s_2)$ . Segments  $s_1, s_2$  are called crucial supporting segments for  $P$ . The number of crucial supporting segments of  $P$  is denoted by  $m(P)$ .*

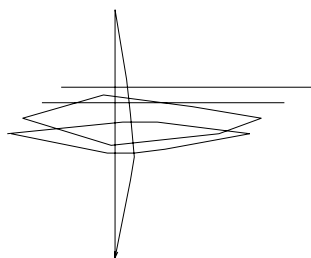
**Theorem 1.** *The structural complexity of  $H\text{-Vor}(S)$  is  $O(n+m)$ , where  $n$  is the total number of points on convex hulls of clusters in  $S$ , and  $m = \sum_{P \in S} m(P)$  is the total number crucial supporting segments between pairs of crossing shapes. The bound is tight in the worst case.*

*Proof.* As it was shown in [9] the structural complexity of  $H\text{-Vor}(S)$  is proportional to the number of mixed Voronoi vertices of  $H\text{-Vor}(S)$ . By Lemma 2, the number of non-crossing mixed Voronoi vertices is  $O(n)$ . (This was also shown in [9]). Also by Lemma 2 (3rd statement) the number of crossing mixed Voronoi vertices is proportional to the number of rear crossing mixed Voronoi vertices. But by Lemma 3, any rear mixed Voronoi vertex can be attributed to a unique pair of supporting segments enclosed in  $\mathcal{K}_{y_0}$ . By considering the center of the minimum enclosing circle of  $P$  as the root of  $T(P)$ , the pair of supporting segments associated with a rear crossing mixed Voronoi vertex must be crucial. Therefore, the number of crossing mixed Voronoi vertices is  $O(m)$ .

To obtain the lower bound it is enough to construct a set  $S$  such that every rear crossing vertex of  $b_h(P, Q)$ ,  $P, Q \in S$ , remains a Voronoi vertex in  $H\text{-Vor}(S)$ . Consider a vertical segment  $P = \overline{p_1 p_2}$  and its minimum enclosing circle,  $\mathcal{K}_P$ . Let  $\epsilon > 0$  be a small constant and let  $l_i, 1 < i \leq k$ , be a set of horizontal lines, each  $\epsilon > 0$  above the other, where  $l_1$  is the horizontal line through the midpoint of  $P$ . Let  $Q_i, 1 \leq i \leq k$  be a set of horizontal line segments each located on line  $l_i$  (see Figure 1). Let  $\mathcal{K}_i$  (resp.  $R_i$ )  $1 \leq i \leq k$  be the  $P$ -circle passing through the leftmost (resp. rightmost) point of  $Q_i$ . The left endpoint of  $Q_i$  is chosen in the interior of  $\mathcal{K}_{i-1}$ ,  $\epsilon$  away from the boundary of  $\mathcal{K}_{i-1}$ . The right endpoint of  $Q_i$  is in the exterior of  $\mathcal{K}_{i-1}$ ,  $\epsilon/2$  away from the boundary. By construction,  $R_i$  and  $\mathcal{K}_i$  enclose exactly  $P$  and  $Q_i$  in their interior and the same holds for any  $P$ -circle centered on  $l_i$  between the centers of  $R_i$  and  $\mathcal{K}_i$ . Thus, the center of  $\mathcal{K}_i, 1 \leq i \leq k$ , remains a vertex in  $H\text{-Vor}(S)$ . By Lemma 4 there can be no  $Q_i$ -circle enclosing  $P$  i.e., all vertices of  $b_h(P, Q_i)$  must be incident to  $T(P)$ . Thus, the only crucial supporting segments are the  $O(k)$  supporting segments between  $P$  and  $Q_i$ , each pair inducing a rear mixed Voronoi vertex in  $H\text{-Vor}(S)$ . The construction is shown using segments for clarity only. Each  $Q_i$  can be substituted by a cluster  $Q'_i$  of arbitrarily many points forming a thin convex shape around segment  $Q_i$  such that no endpoint of the original set of segments is enclosed in  $CH(Q'_i)$  as shown in Figure 2. Segment  $P$  can also be substituted by a cluster  $P'$  forming a thin convex shape to the right of segment  $P$  with no change in the arguments of the proof. □



**Fig. 1.** The  $\Omega(m)$  construction for the complexity of  $H\text{-Vor}(S)$ .



**Fig. 2.** Segments can be substituted by convex polygons.

## 4 A Plane Sweep Algorithm

In this section we give a plane sweep algorithm to construct the Hausdorff Voronoi diagram of  $S$ . The algorithm is based on the standard plane sweep paradigm of [2,4] but requires special *events* to handle mixed Voronoi vertices and disconnected Voronoi regions. The farthest Voronoi diagram of each individual cluster  $P$  is assumed to be available and it can be constructed by divide and conquer in  $O(|P| \log |P|)$  time. The plane sweep basically stitches together the farthest Voronoi diagrams of the individual clusters of  $S$  into the Hausdorff Voronoi diagram of  $S$ .

The plane sweep process assumes a vertical sweep-line  $l_t$  sweeping the entire plane from left to right. The distance to the sweeping line is measured in the ordinary way and not in the Hausdorff metric i.e.,  $d(p, l_t) = \min\{d(p, y), y \in l_t\}$  for any point  $p$ . At any instant  $t$  of the sweeping process we compute  $H\text{-Vor}(S_t \cup l_t)$  for  $S_t = \{P \in S \mid \max_{p \in P} x(p) < t\}$  where  $x(p)$  is the  $x$ -coordinate of point  $p \in P$ , and  $d_h(p, l_t) = d(p, l_t)$ . Following the terminology of [2] the boundary of the Voronoi region of  $l_t$  is called the *wavefront* at time  $t$ . The bisectors incident to the wavefront are called *spike bisectors* and consist of inter- and intra- spike bisectors. The wavefront consists of parabolic arcs, called *waves*, corresponding to ordinary bisectors between points  $p \in S_t$  and the sweep line  $l_t$ . As the sweep line moves to the right, the wavefront and the endpoints of spike bisectors move continuously to the right. The combinatorial structure of the wavefront changes only at certain *events* organized in an *event queue*. We have two types of events, *site events* when new waves appear in the wavefront, and *spike events* when old waves disappear. Spike events correspond to the intersection of two neighboring spike bisectors and their treatment remains similar to the ordinary plane sweep paradigm. Site events are different and their handling is illustrated in the following.

**Definition 3.** *The priority of any point  $v \in H\text{-Vor}(S)$  or  $v \in f\text{-Vor}(P)$ ,  $P \in S$ , is  $\text{priority}(v) = x(v) + d(v, p_i)$ , where  $p_i$  is the owner of region bounded by  $v$  in  $H\text{-Vor}(S)$  or  $f\text{-Vor}(P)$  respectively. In other words,  $\text{priority}(v)$  equals the rightmost  $x$ -coordinate of the circle centered at  $v$  passing through  $p_i$ .*

The point of minimum priority for any cluster  $P$  is the intra-bisector point  $Y_0(P)$  (for brevity  $Y_0$ ) derived by shooting a horizontal ray backwards from the rightmost point  $p_r$  of  $P$ , until it hits the boundary of  $freg(p_r)$  in  $f\text{-Vor}(P)$ . Throughout this section (unless explicitly noted otherwise) we assume that the root of  $T(P)$  is  $Y_0(P)$ . As a result, the definition of a rear/forward mixed Voronoi vertex or rear/forward cluster always assumes that the corresponding intra-bisector tree is rooted at the point of minimum priority. The  $P$ -circle  $\mathcal{K}_0$  centered at  $Y_0(P)$  is referred to as the *minimum priority circle* of  $P$ . Clearly,  $\text{priority}(Y_0) = x(p_r)$ .

**Lemma 5.** *Let  $y_i \in T(P)$ . Then  $\text{priority}(y_i) < \text{priority}(y_j)$  for any  $y_j \in T(y_i)$  (assuming that the the root of  $T(P)$  is  $Y_0(P)$ ).*

*Proof.* The priority of any  $y_i \in T(P)$  is given by the rightmost vertical line  $l$  tangent to  $\mathcal{K}_{y_i}$ . Since  $priority(Y_0) \leq priority(y_i)$ , and  $\mathcal{K}_{y_i}^r \subset \mathcal{K}_0$  (Lemma 1),  $l$  must be tangent to  $\mathcal{K}_{y_i}^f$ . But by Lemma 1,  $\mathcal{K}_{y_i}^f \subset \mathcal{K}_{y_j}^f$  for every  $y_j \in T(y_i)$ . Thus,  $priority(y_i) < priority(y_j)$ .  $\square$

The following Lemma can be easily derived from Lemma 2 and Lemma 5.

**Lemma 6.** *Let  $hreg_i(P)$  be a connected component of  $hreg(P)$ ,  $P \in S$ , that does not contain  $Y_0(P)$ . Then  $hreg_i(P)$  must have exactly one rear mixed Voronoi vertex on  $T(P)$  and this is the point of minimum priority of  $hreg_i(P)$ . If  $Y_0(P) \in hreg_i(P)$  all mixed Voronoi vertices of  $hreg_i(P) \cap T(P)$  are forward.*

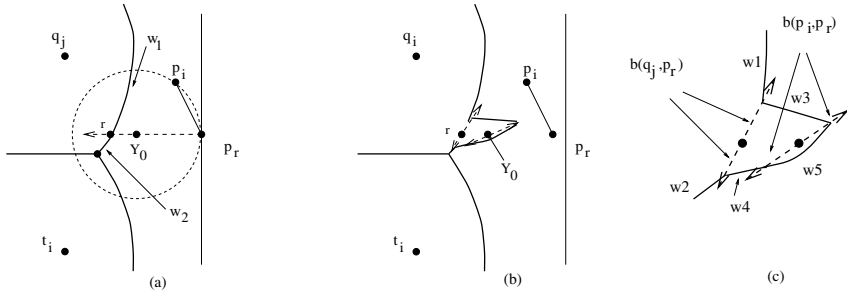
By the definition of the wavefront, any Voronoi point  $v \in H\text{-Vor}(S)$  must enter the wavefront at time  $t = priority(v)$ . Thus, appropriate site events need to be generated so that at least one event exists (site or spike event) for every vertex of  $H\text{-Vor}(S)$ . We define two types of site events: *ordinary vertex events*, one for every vertex of  $T(P)$ ,  $P \in S$ , and *mixed vertex events* (for brevity *mixed events*) whose purpose is to predict the rear mixed Voronoi vertices of  $H\text{-Vor}(S)$ . Ordinary vertex events are readily available from  $f\text{-Vor}(P)$ ,  $P \in S$ . Mixed events get generated throughout the algorithm. An event is called *valid* if it falls on or ahead the wavefront at the time of its priority. An event falling behind the wavefront at the time of its priority is called *invalid*. Any event is processed at the time of its priority.

Let's first consider the vertex event corresponding to  $Y_0(P)$  and let's assume that the event is valid. Let  $\overline{p_i p_r}$  be the chord inducing  $Y_0$  (i.e.,  $Y_0 \in b(p_i, p_r)$ ) where  $p_r$  is the rightmost point of  $P$ . Then at time  $t$ , the waves of  $p_i$  and  $p_r$  enter the wavefront for the first time, separated by the intra-bisector  $b(p_i, p_r)$ . In more detail, let  $q_j$  be the owner of the intersection point  $r$  where the horizontal ray from  $p_r$  hits the wavefront (see Figure 3). The wave of  $q_j$  is split at point  $r$  into two waves  $w_1, w_2$  (say  $w_1$  is above  $w_2$ ), and the two rays of the spike bisector  $b(q_j, p_r)$  emanating from  $r$  enter the wavefront, serving as new gliding tracks for waves  $w_1$  and  $w_2$ . Furthermore, three new waves enter the wavefront: waves  $w_3, w_4$  for  $p_r$  and wave  $w_5$  for  $p_j$  gliding along the two rays of  $b(p_j, p_r)$  emanating from  $Y_0$ . The ordering of the waves from top to bottom is  $w_1, w_3, w_5, w_4, w_2$ . Figures 3a and 3b depict the wavefront before and after the update, and Figure 3c shows the topological arrangement of the new waves. Spike events are generated as in the ordinary Voronoi diagram construction.

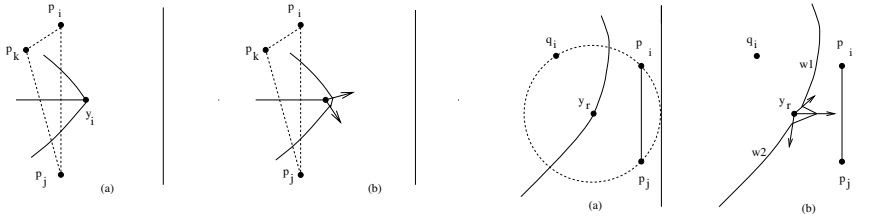
The update of the wavefront at any other valid ordinary vertex event  $y_i \in T(P)$ ,  $y_i \neq Y_0$ , is similar and is depicted in Figure 4. In detail, let  $p_i, p_j, p_k \in P$  be the points inducing  $y_i$ . Since  $y_i$  is valid, at time  $t = priority(y_i)$ , point  $y_i$  must be a point of the wavefront incident to exactly one spike intra-bisector, say  $b(p_i, p_j)$ . Note that by Lemma 5 only one of the intra-bisectors incident to  $y_i$  can contain points of lower priority than  $y_i$ . At time  $t$ , a new wave for point  $p_k$  must enter the wavefront gliding between the spike intra-bisectors  $b(p_k, p_i)$  and  $b(p_k, p_j)$  as shown in Figure 4.

Let's now consider the handling of an invalid vertex event  $y_i \in T(P)$  ( $y_i$  may be  $Y_0$ ) and the generation of a mixed vertex event. At time  $t = priority(y_i)$  point





**Fig. 3.** The wavefront update at a valid minimum priority vertex event.



**Fig. 4.** The wavefront update at an ordinary valid vertex event.

**Fig. 5.** The update of the wavefront at a valid mixed vertex event.

$y_i$  is behind the wavefront. Let  $y_j$  be any immediate descendent of  $y_i$  in  $T(P)$  that is not yet covered by the wavefront (if any). Then segment  $\overline{y_i y_j}$  must intersect the wavefront at a point  $Y$ . The following process repeats for all immediate descendents of  $y_i$  that are not yet covered by the wavefront. Let  $q_j \in Q_j$  be the owner of  $Y$  in  $H\text{-Vor}(S_t)$  and let  $\overline{y_i y_j} \in b(p_i, p_j)$ . Point  $q_j$  may be rear or forward with respect to  $\overline{p_i p_j}$ . Let's first consider the case where  $Q_j$  is non-crossing with  $\overline{p_i p_j}$ . If  $q_j$  is forward then by Lemma 2,  $T(Y) \cap h\text{reg}(P) = \emptyset$ . Thus, we eliminate all vertex events of  $T(y_j)$  from the event queue. If  $q_j$  is rear, we eliminate from the event queue any vertex event associated with  $T_c(Y)$ . If  $q_j$  is rear and  $d(y_j, p_j) < d_f(y_j, Q_j)$  then  $b_f(Q_j, P)$  must intersect  $\overline{y_i y_j}$  at a mixed vertex  $m_i$ . Since vertex  $m_i$  may or may not appear in  $H\text{-Vor}(S)$ , a mixed-vertex event needs to get generated for  $m_i$ . Vertex  $m_i$  can be easily determined in  $O(|CH(Q_j)|)$  time by considering intersections of  $\overline{y_i y_j}$  with  $f\text{-Vor}(Q_j)$  (see also Lemma 9). Note that if  $d(y_j, p_j) \geq d_f(y_j, Q_j)$ , no mixed event gets generated as there can be no portion of  $h\text{reg}(P)$  on  $\overline{y_i y_j}$ . Let's now assume that  $Q_j$  is crossing with  $\overline{p_i p_j}$ . Similarly to the non-crossing case, if  $d(y_j, p_j) < d_f(y_j, Q_j)$  then a mixed vertex event  $m_i$  needs to get generated for  $\overline{y_i y_j}$  at the point where  $b_f(Q_j, P)$  intersects  $\overline{y_i y_j}$ . Although  $b_f(Q_j, P)$  may induce several rear mixed vertices on  $T(P)$ , it can induce at most one on a single segment  $\overline{y_i y_j}$ . To determine  $m_i$  we walk on  $\overline{y_i y_j}$  backwards starting at  $y_j$ , considering intersections with  $f\text{-Vor}(Q_j)$ , until  $m_i$  is determined. The search starts at  $y_j$  and not at  $Y$  to maintain time complexity as it will be evident in Lemma 9.

The handling of a mixed vertex event is similar to the handling of an ordinary vertex event. In particular, let  $y_r \in T(P)$  be a mixed vertex event induced by  $p_i, p_j \in P, q_i \in Q$  and let  $y_j$  be the immediate descendent of  $y_r$  in  $T(P)$ .  $y_r$  is valid iff at time  $t = \text{priority}(y_r)$ ,  $y_r$  is a point of the wavefront, in particular a point on the wave of  $q_i$ . If  $y_r$  is valid then the wave of  $q_i$  is split into two waves  $w_1, w_2$ , and new waves for  $p_i$  and  $p_j$  enter the wavefront between  $w_1$  and  $w_2$ , separated by the spike intra-bisector  $\overline{y_r y_j} \in b(p_i, p_j)$ , as shown in Figure 5. If  $y_r$  is determined to be invalid then a new mixed-vertex event may be generated similarly to the case of an invalid vertex event.

The handling of spike events is identical to the ordinary Voronoi diagram construction with the exception of the need to generate mixed events for crossing clusters. A spike event between a spike intra-bisector  $b(p_i, p_j)$  of  $P$  and a spike inter-bisector  $b_h(P, Q)$  corresponds to a forward mixed Voronoi vertex  $v$  induced on  $T(P)$  by  $Q$ . If  $Q$  is crossing with  $\overline{p_i p_j}$  then  $v$  may be followed on  $T(P)$  by a rear mixed Voronoi vertex. Thus, in this case, a mixed event induced by  $Q$  on  $b(p_i, p_j)$  may need to be generated, exactly as for the case of an invalid ordinary vertex event. If  $Q$  is non crossing with  $\overline{p_i p_j}$  then  $T(v) \cap \text{hreg}(P) = \emptyset$  and thus, we can eliminate all the vertex events of  $T(v)$  from the event queue.

The correctness of the algorithm follows from the correctness of the plane sweep paradigm of [2] as long as we can show that no rear mixed Voronoi vertex can be missed. In other words, we need to show that a mixed vertex event gets generated for every rear mixed Voronoi vertex of  $H\text{-Vor}(S)$ . Note that by Lemma 3, the minimum priority of any connected component of  $\text{hreg}(P)$  occurs at a rear mixed Voronoi vertex of  $T(P)$  (if not  $Y_0(P)$ ), therefore a starting point for all connected Voronoi regions can be obtained if we have events for all rear mixed Voronoi vertices and the roots of intra-bisector trees. The following lemma shows that no mixed events can be missed and can be derived from the above discussion.

**Lemma 7.** *Let  $y_i \in T(P)$  be an invalid site event or a valid spike event involving a crossing cluster  $Q$ . Let  $y_j$  be an immediate descendent of  $y_i$  in  $T(P)$ . If a mixed-vertex event  $y_r$  gets generated on segment  $\overline{y_i y_j} \in T(P)$  during the handling of  $y_i$ , then  $\overline{y_i y_r} \cap \text{hreg}(P) = \emptyset$ . If no mixed-vertex event gets generated on  $\overline{y_i y_j}$  then  $\overline{y_i y_j} \cap \text{hreg}(P) = \emptyset$ .*

The time complexity depends on the number of mixed vertex events that get generated and the time to produce them. In the following we concentrate in formally bounding this number.

**Definition 4.** *Let  $r$  be the horizontal ray extending backwards from the right-most point  $p_r$  of  $P$ . Let  $C_r$  be the circle centered on  $r$  passing through  $p_r$  that contains exactly one cluster in its interior (in the worst case that cluster is  $P$ ). The cluster contained in  $C_r$  is called the anchor of  $P$ .*

**Definition 5.** *Let  $A$  be the anchor of  $P$ . If  $A \neq P$ , let  $Y_a$  be the nearest common ancestor of all rear vertices of  $b_f(P, A)$  on  $T(P)$ . The  $P$ -circle centered at  $Y_a$  is called the anchor circle of  $P$  and it is denoted as  $\mathcal{K}_a(P)$ . If  $A = P$ , the anchor*

circle is the minimum enclosing circle of  $P$ . The priority of the anchor circle is  $\text{priority}(Y_a)$ .

The anchor circle coincides with the minimum enclosing circle of  $P$  when  $Y_0(P) \in \text{hreg}(P)$ . In case of non-crossing clusters, the anchor circle of  $P$  is the minimum radius  $P$ -circle entirely enclosing both  $P$  and its anchor. In the worst case, the anchor circle of  $P$  coincides with the minimum priority circle of  $P$ , in which case  $Y_a = Y_0$  and  $T_c(Y_a) = \emptyset$ . Since any component of  $\text{hreg}(P) \cap T(P)$  must be bounded by a rear mixed vertex or  $Y_0$ ,  $T_c(Y_a) \cap \text{hreg}(P) = \emptyset$ . For the sake of formally bounding the number of mixed vertex events, we can add the following step when handling the invalid site event corresponding to  $Y_0(P)$ , in order to ensure that no mixed vertex events get generated for  $T_c(Y_a)$ : Determine the anchor of  $P$  and produce a mixed vertex event for  $Y_a$  (if  $Y_a$  does not coincide with a vertex of  $T(P)$ ); eliminate  $T_c(Y_a)$  i.e., delete all vertex events of  $T_c(Y_a)$  from the event queue. The correctness of the algorithm is not affected by the addition of this step.

**Lemma 8.** *The number of mixed vertex events generated throughout the algorithm is  $O(K + m)$ , where  $K = \sum_{P \in S} K(P)$ ,  $K(P)$  is the number of clusters entirely enclosed in the anchor circle of  $P$ , and  $m$  is as defined in Theorem 1.*

*Proof.* We have two types of mixed vertex events: *crossing* and *non-crossing*. Let  $m_i$  be a non-crossing mixed vertex event induced by  $Q$  on  $T(P)$ . By Lemma 2,  $m_i$  is unique as  $Q$  can induce at most one rear non-crossing mixed vertex on  $T(P)$ . Since  $Q \in \mathcal{K}_{m_i}^r \cup CH(P)$  and  $m_i \in T(Y_a)$ ,  $Q$  must be entirely enclosed in  $\mathcal{K}_a(P)$  as  $\mathcal{K}_{m_i}^r \subset \mathcal{K}_a(P)$ . Thus,  $O(K)$  bounds the total number of non-crossing mixed events. By construction the total number of crossing mixed vertex events is upper bounded by the the total number of crossing vertices on  $b_h(P, Q)$  for any pair of crossing clusters  $(P, Q)$ . But this number is  $O(m)$  as it was shown in Theorem 1. □

**Lemma 9.** *The generation time for all mixed vertex events induced on  $T(P)$  by a single cluster  $Q$  is  $O(|CH(Q) \cap \mathcal{K}_a(P)|)$ .*

*Proof.* The claim is easy to see for a non-crossing  $Q$  as  $Q \in \mathcal{K}_a(P)$  and  $Q$  can induce at most one mixed event on  $T(P)$  (see Lemma 8). If  $Q$  is crossing with  $P$  then  $Q$  may induce several crossing mixed vertex events on  $T(P)$ . However, we claim that any  $q_r \in Q$  that gets visited during the generation of a single mixed vertex event  $m_i$ , cannot be visited again during the generation of another mixed vertex event on  $T(P)$ . Let  $q_r \in Q$  be visited during the generation of  $m_i \in \overline{y_i y_j} \in T(P)$ , that is,  $\text{freq}(q_r)$  is intersected by  $\overline{m_i y_j}$ . (Recall that the traversal of  $\overline{y_i y_j}$  starts at  $y_j$  and  $y_i$  is an ancestor of  $y_j$  in  $T(P)$ ). Then  $q_r \in \mathcal{K}_{y_i}^r$  but  $q_r \notin \mathcal{K}_y$  for any  $y \in \overline{m_i y_j} \cap \text{freq}(q_r)$ . Thus,  $q_r \notin \mathcal{K}_v^r$  for any  $v \in T(y_j)$  since  $\mathcal{K}_v^r \subset \mathcal{K}_{y_j}^r$  (Lemma 1). Furthermore,  $q_r \notin \mathcal{K}_u^r$  for any  $u \in T_c(y_i)$  that is not an ancestor of  $y_i$ , since  $\mathcal{K}_{y_i}^r \cap \mathcal{K}_u^r = \emptyset$ . Thus,  $q_r$  cannot be considered again during the generation of another mixed vertex event on  $T(P)$ . Since  $m_i \in T(Y_a)$ ,  $q_r \in \mathcal{K}_{y_i}^r \subset \mathcal{K}_a(P)$ . Thus, only points in  $CH(Q) \cap \mathcal{K}_a(P)$  can be visited. □

**Theorem 2.**  $H\text{-Vor}(S)$  can be computed in  $O(M + (n + m + K) \log n)$  time by plane sweep, where  $n, m$  are as defined in Theorem 1,  $K$  is as defined in Lemma 8, and  $M = \sum_{P \in S} M(P)$ , where  $M(P)$  is the total number of points  $q \in CH(Q)$  that are enclosed in the anchor circle of  $P$  such that either  $Q$  is entirely contained in  $\mathcal{K}_a(P)$  or  $Q$  is crossing with  $P$ .

*Proof.* By Lemma 9,  $O(M)$  time is attributed to the generation of mixed vertex events. The theorem follows from Lemma 8 and the fact that  $O(\log n)$  time is spent for each event (without counting time for generation of mixed events).  $\square$

Whether the terms  $K, M$  can be eliminated from the time complexity remains an open problem. In our VLSI application shapes are rectilinear in nature, well spaced, and in their majority non-crossing. A small number of crossings may be present due to non-neighboring redundant vias. In this setting both  $K, M$  remain small. In the simpler  $L_\infty$  non-crossing case, experimental results on VLSI vias-layers indicated that  $K$  was negligible compared to  $n$  [7].

## References

1. M. Abellanas, G. Hernandez, R. Klein, V. Neumann-Lara, and J. Urrutia, "A Combinatorial Property of Convex Sets", *Discrete Computat. Geometry* 17, 1997, 307-318.
2. F. Dehne and R. Klein, "The Big Sweep": On the power of the Wavefront Approach to Voronoi Diagrams", *Algorithmica*, 17, 19-32, 1997.
3. H. Edelsbrunner, L.J. Guibas, and M. Sharir, "The upper envelope of piecewise linear functions: algorithms and applications", *Discrete Computat. Geometry* 4, 1989, 311-336.
4. S. J. Fortune, "A sweepline algorithm for Voronoi diagrams", *Algorithmica*, 2, 1987, 153-174.
5. R. Klein, K. Mehlhorn, S. Meiser, "Randomized Incremental Construction of Abstract Voronoi diagrams", *Computational geometry: Theory and Applications* 3, 1993, 157-184
6. W. Maly, "Computer Aided Design for VLSI Circuit Manufacturability," *Proc. IEEE*, vol.78, no.2, 356-392, Feb. 90.
7. E. Papadopoulou, "Critical Area Computation for Missing Material Defects in VLSI Circuits", *IEEE Transactions on Computer-Aided Design*, vol. 20, no.5, May 2001, 583-597.
8. E. Papadopoulou and D.T. Lee, "Critical Area Computation via Voronoi Diagrams", *IEEE Trans. on Computer-Aided Design*, vol. 18, no.4, April 1999, 463-474.
9. E. Papadopoulou and D.T. Lee, "The Min-Max Voronoi diagram of polygonal objects and applications in VLSI manufacturing", *Proc. 13th Int. Symp. on Algorithms and Computation*, Nov 2002, LNCS 2518, 511-522.
10. Preparata, F. P. and M. I. Shamos, *Computational Geometry: an Introduction*, Springer-Verlag, New York, NY 1985.
11. H. Walker and S.W. Director, "VLASIC: A yield simulator for integrated circuits," *IEEE Trans. on Computer-Aided Design*, CAD-5,4, 541-556, Oct. 1986.