Algorithmica (2004) 40: 63–82
DOI: 10.1007/s00453-004-1095-0

Algorithmica

The Hausdorff Voronoi Diagram of Point Clusters in the Plane[1]

Evanthia Papadopoulou[2]

Abstract. We study the *Hausdorff Voronoi diagram* of point clusters in the plane, a generalization of Voronoi diagrams based on the Hausdorff distance function. We derive a tight combinatorial bound on the structural complexity of this diagram and present a plane sweep algorithm for its construction. In particular, we show that the size of the Hausdorff Voronoi diagram is $\Theta(n + m)$, where n is the number of points on the convex hulls of the given clusters, and m is the number of *crucial supporting segments*[3] between pairs of *crossing clusters*.[4] The plane sweep algorithm generalizes the standard plane sweep paradigm for the construction of Voronoi diagrams with the ability to handle disconnected Hausdorff Voronoi regions. The Hausdorff Voronoi diagram finds direct application in the problem of computing the *critical area* of a VLSI layout, a measure reflecting the sensitivity of the VLSI design to spot defects during manufacturing.

Key Words. Voronoi diagram, Hausdorff distance, Plane sweep, VLSI yield prediction, VLSI critical area, Manufacturing defects, Via-blocks.

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 than to any other cluster in S, according to the *Hausdorff distance*.[5] 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 (see [14] for a proof of equivalence). 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 in the plane 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. It can also be regarded as the reverse of the *farthest color Voronoi diagram* presented in [2].

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 [4] it was termed the *Voronoi diagram of*

[1] A preliminary version of this paper appeared in the *Proceedings of the Workshop on Algorithms and Data Structures* 2003 (WADS 2003), LNCS 2748, pp. 439–450.

[2] IBM TJ Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, USA. evanthia@watson.ibm.com.

[3] See Definition 4. A crucial supporting segment between clusters P and Q must appear on the convex hull of $P \cup Q$ and must be enclosed in the minimum enclosing circle of P or Q.

[4] See Definition 1. Two clusters P, Q are called crossing if the convex hull of $P \cup Q$ admits more than two supporting segments between the convex hulls of P and Q.

[5] The (directed) Hausdorff distance from a set A to a set B is $h(A, B) = \{\max_{a \in A} \min_{b \in B} d(a, b)\}$. The (undirected) Hausdorff distance between A and B is $d_{\mathrm{h}}(A, B) = \max\{h(A, B), h(B, A)\}$.

Received September 11, 2003; revised February 7, 2004. Communicated by B. Chazelle.
Online publication May 14, 2004.

point clusters, in [1] the *closest covered set diagram*, and in [12] and [14] the *min-max Voronoi diagram*. The equivalence to the Voronoi diagram under the Hausdorff metric was shown in [14]. In [4] several combinatorial bounds regarding this diagram were derived using a powerful geometric transformation in three dimensions. Under this transformation the diagram can be simply regarded as the vertical projection of the upper envelope of a set of convex surfaces, where each surface is the lower envelope of a family of planes in three dimensions [4]. Each plane is the unique transformation of a point $p \in S$ derived by lifting p onto the unit paraboloid at point p' and considering the plane touching the unit paraboloid at p'. Using this transformation the size of the diagram was shown in [4] to be $O(n^2\alpha(n))$ 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. It was also shown that in the case of n intersecting line segments, the size of the diagram can be $\Omega(n^2)$. By applying a divide and conquer technique for constructing the upper envelope of piecewise linear functions in three dimensions, an $O(n^2\alpha(n))$-time algorithm for the construction of the diagram was also given in [4]. In [1] the linear size of the Hausdorff Voronoi diagram for clusters of points with disjoint convex hulls was generalized to arbitrary convex shapes under arbitrary convex distance functions. The problem for disjoint convex sets was reduced to abstract Voronoi diagrams and the randomized incremental construction of [7] was proposed for its construction. This resulted in an $O(kn \log n)$-expected-time algorithm, where k is the time to construct the Hausdorff bisector between two convex polygons. In our companion paper [14], the size of the Hausdorff Voronoi diagram was shown to be linear in the case of *non-crossing*[6] clusters, i.e., clusters of points with *non-crossing* convex hulls, not necessarily disjoint. In the general case of arbitrary clusters of points, the size of the diagram was shown to be $O(n^2)$ in the worst case. This tighter combinatorial bound automatically improved the time complexity of [4] to $O(n^2)$. A divide and conquer construction was given in [15], for both the general and the non-crossing case, of time complexity $O(M' + n \log^2 n + (m + K') \log n)$, where m is the size of the diagram, $K' = \sum_{P \in S} K'(P)$, where $K'(P)$ is the number of clusters enclosed in the minimum enclosing circle of P, and $M' = \sum_{P \in S} M'(P)$, where $M'(P)$ is the total number of points $q \in Q$ enclosed in the minimum enclosing circle of P such that Q is entirely enclosed in the minimum enclosing circle of P or Q is crossing with P. In [12] the simpler L_∞ 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. The reverse type of Voronoi diagram, i.e., the *farthest color Voronoi diagram* of [2], had also been considered in [6], by means of a geometric transformation that reduces the problem to computing the upper envelope of a set of Voronoi surfaces in three dimensions.

In this paper we continue the study on the Hausdorff Voronoi diagram initiated in our companion paper [14]. We derive a tight combinatorial bound on the structural complexity of this diagram and present a simple plane sweep algorithm for its construction. In particular, we show that the size of the Hausdorff Voronoi diagram is $\Theta(n + m)$,

[6] See Definition 1. Two clusters P, Q are called non-crossing if the convex hull of $P \cup Q$ admits at most two supporting segments between P and Q.

where m is the number of *crucial supporting segments*[7] among pairs of crossing clusters. In the worst case, m is $O(n^2)$. The time complexity of the plane sweep algorithm is $O(M + (n + m + K)\log n)$, where $K = \sum_{P \in S} K(P)$, $K(P)$ is the number of clusters entirely enclosed in the *anchor circle* of P, and $M = \sum_{P \in S} M(P)$, where $M(P)$ is the number of convex hull points $q \in Q$ that are 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 Definition 7) in general different than the minimum enclosing circle of P. The algorithm improves upon previous results especially when the number of crossings is small, and its time complexity is comparable with the one obtained by divide and conquer in [15]. The plane sweep construction, however, is much simpler. Plane sweep has been the method of choice for our application because of the very large size of VLSI designs as explained below. Our plane sweep construction generalizes the standard plane sweep paradigm for Voronoi diagrams [5], [3] with the ability to handle disconnected Hausdorff Voronoi regions. To the best of our knowledge disconnected Voronoi regions have not been considered by plane sweep methods so far.

This paper is organized as follows. In the remainder of this section we describe a direct application of the Hausdorff Voronoi diagram in computing the *critical area* of a VLSI layout. In Section 2 we give preliminary definitions for the Hausdorff Voronoi diagram and we review concepts and terminology introduced in our companion paper [14]. Section 3 derives a tight combinatorial bound on the structural complexity of the Hausdorff Voronoi diagram. In Section 4 we present a simple plane sweep algorithm for the construction of this diagram. Section 5 provides concluding remarks.

1.1. *Application to VLSI Critical Area Computation.* Our motivation for investigating the Hausdorff Voronoi diagram comes from an application in VLSI manufacturing as explained in [12], in particular *critical area extraction* for predicting the *yield* of a VLSI chip. The *yield* of a VLSI design is the percentage of functional chips over all chips manufactured. The *critical area* is a measure reflecting the sensitivity of a VLSI design to random defects during manufacturing and it is widely used to predict yield. One of the most serious defect mechanisms for yield loss are the blockages of *vias*, called *via-blocks*, between conducting regions in two different layers (see, e.g., [9] and [11]). A *via-block* is a defect that entirely covers a polygon representing a contact region between two layers [10]. Defects are widely modeled as circles following the $1/r^3$ distribution, where r denotes the size of a defect (see, e.g., [8] and [17]). In [12] the critical area computation problem for *via-blocks* was shown to be reducible to the Hausdorff Voronoi diagram (termed the min-max Voronoi diagram). A VLSI via-layer, after preprocessing to identify redundant vias, consists typically of rectilinear shapes, in their majority non-crossing. A relatively small number of crossings may be present due to redundant vias between portions of the same net that are located further apart. Figure 1 illustrates two groups of redundant vias and their corresponding unified contacts. Given a point t on a via layer, the *critical radius* of t is the smallest defect centered at t causing a via block. Figure 2 illustrates the critical radius of points t and t' on the via layer of Figure 1. It

[7] See Definition 4. A crucial supporting segment between clusters P and Q must appear on the convex hull as $P \cup Q$ and must be enclosed in the minimum enclosing circle of P or Q.

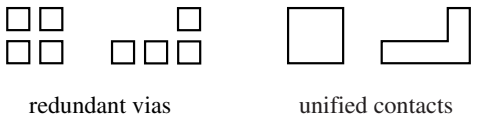

Fig. 1. Contacts as groups of redundant vias (borrowed from [12]).

is now easy to see that the Hausdorff Voronoi diagram of unified contact shapes on a via layer reveals the critical radius for via-blocks for every point on that layer. Once the Hausdorff Voronoi diagram of a via layer is available, the critical area integral can be computed easily, either in an analytical manner as in the L_∞ case (see [12] and [13]), or by applying standard integration techniques in general (see, e.g., [18]). For more details on the critical area computation problem see, e.g., [9], [12]–[14], [18], and [19].

Plane sweep has been our method of choice for the construction of Voronoi diagrams for critical area computation because of memory considerations associated with the very large size of modern VLSI designs. For the computation of critical area there is no need to keep the whole 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 within that cell can be performed independently, and the Voronoi cell can be immediately discarded. For this reason, our CAD tool [20] for the extraction of VLSI critical area for *shorts*, *opens*, *via-blocks*, and *combination faults* has been based on plane sweep. The Hausdorff Voronoi diagram of arbitrary point clusters in the plane provides the framework for general critical area computation for via-blocks under the general circular defect model even in the presence of crossings among contact shapes. Plane sweep has been dictated by the framework of our CAD tool and results in a simple algorithm.

2. Preliminaries and Definitions. The *farthest distance* between two sets of points P, Q is $d_{\mathrm{f}}(P, Q) = \max\{d(p, q),\ p \in P, q \in Q\}$, where $d(p, q)$ denotes the ordinary distance between two points p, q. The (directed) Hausdorff distance from P to Q is $h(P, Q) = \{\max_{p\in P} \min_{q\in Q} d(p, q)\}$. The (undirected) Hausdorff distance between P and Q is $d_{\mathrm{h}}(P, Q) = \max\{h(P, Q), h(Q, P)\}$. The Hausdorff bisector between P and Q is $b_{\mathrm{h}}(P, Q) = \{y \mid d_{\mathrm{h}}(y, P) = d_{\mathrm{h}}(y, Q)\}$. In [14] the Hausdorff bisector was shown to be a subgraph of *f-Vor*$(P \cup Q)$, equivalent to the bisector between P and Q under the farthest distance function, i.e., $b_{\mathrm{h}}(P, Q) = \{y \mid d_{\mathrm{f}}(y, P) = d_{\mathrm{f}}(y, Q)\}$. The ordinary bisector between any two points p, q is denoted as $b(p, q)$.

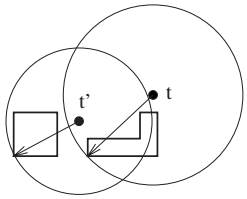

Fig. 2. The critical radius for via-blocks for points t, t'.

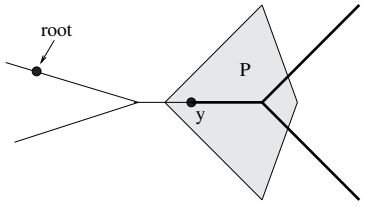

Fig. 3. The intra-bisector tree of P.

The Voronoi diagram of a set of points P under the farthest distance function is called the *farthest Voronoi diagram* and it is denoted as *f-Vor*(P). It is well known (see, e.g., [16]) that *f-Vor*(P) consists of unbounded convex regions, one for each point on the convex hull of P. More precisely, the *farthest Voronoi region* of point $p_i \in P$ is $freg(p_i) = \{x \mid d(x, p_i) > d(x, p_j), p_j \in P\}$. The bisectors of *f-Vor*(P) are portions of ordinary bisectors between points on the convex hull of P. To distinguish between the bisectors of the farthest Voronoi diagram for a single cluster P and the Hausdorff bisector between a pair of different clusters, we use the terms *intra-bisector* and *inter-bisector*, respectively. That is, any ordinary bisector of *f-Vor*(P) is called an *intra-bisector* and any portion of a Hausdorff bisector $b_{\mathrm{h}}(P, Q)$ is called an *inter-bisector*. The convex hull of P is denoted as $CH(P)$. A segment $\overline{p_i p_j}$ connecting any two points on $\mathrm{CH}(P)$ is called a *chord*.

The tree structure of *f-Vor*(P), rooted at an arbitrary point y_0, is termed the *intra-bisector tree* of P and it is denoted by $\mathcal{T}(P)$. Every point $y \in \mathcal{T}(P)$ is weighted by $d_{\mathrm{f}}(y, P)$, the radius of the smallest circle centered at y entirely enclosing P. The circle centered at y of radius $d_{\mathrm{f}}(y, P)$ is called a *P-circle* and it is denoted as $\mathcal{K}_y$. The P-circle $\mathcal{K}_y$ must pass through points $p_i, p_j \in P$, where $y \in b(p_i, p_j)$. Depending on the choice of the root, point y partitions $\mathcal{T}(P)$ in two parts: $\mathcal{T}(y)$ and $\mathcal{T}_{\mathrm{c}}(y)$, where $\mathcal{T}(y)$ consists of the descendents of y in the rooted $\mathcal{T}(P)$, and $\mathcal{T}_{\mathrm{c}}(y)$ is the complement of $\mathcal{T}(y)$. In Figure 3 $\mathcal{T}(y)$ is depicted in bold. If y is a vertex of $\mathcal{T}(P)$ and an incident intra-bisector segment $\overline{yy_j}$ is explicitly specified to contain y, then $\mathcal{T}(y)$ consists of the subtree rooted at y containing segment $\overline{yy_j}$; $\mathcal{T}_{\mathrm{c}}(y)$ is still the complement of $\mathcal{T}(y)$. The P-circle $\mathcal{K}_y$ is also partitioned in two parts by chord $\overline{p_i p_j}$: the portion enclosing the points of $CH(P)$ inducing $\mathcal{T}(y)$, called the *rear* portion of $\mathcal{K}_y$, denoted as $\mathcal{K}_y^{\mathrm{r}}$, and the portion enclosing the points of $CH(P)$ inducing $\mathcal{T}_{\mathrm{c}}(y)$, called the *forward* portion of $\mathcal{K}_y$, denoted as $\mathcal{K}_y^{\mathrm{f}}$. Figure 4 depicts $\mathcal{K}_y^{\mathrm{f}}$ shaded and $\mathcal{K}_y^{\mathrm{r}}$ transparent. The characterization of the portions of $\mathcal{K}_y$ as *rear* or *forward* depends on the root of $\mathcal{T}(P)$ and it can be reversed for a different choice of the root. The following observation is a generalization of one given in [14] and is used throughout this paper.

LEMMA 1. *For any point $y \in \mathcal{T}(P)$, such that $y \in b(p_i, p_j)$, the following holds: For any point $x \in \mathcal{T}(y)$, $\mathcal{K}_y^{\mathrm{f}} \subset \mathcal{K}_x^{\mathrm{f}}$ and $\mathcal{K}_x^{\mathrm{r}} \subset \mathcal{K}_y^{\mathrm{r}}$. For any point $x \in \mathcal{T}_{\mathrm{c}}(y)$, $\mathcal{K}_y^{\mathrm{r}} \subset \mathcal{K}_x$, in particular, $\mathcal{K}_y^{r} \subset \mathcal{K}_x^{f}$, for any $x \in \mathcal{T}_c(y)$ that is not an ancestor of y, and $\mathcal{K}_y^{\mathrm{r}} \subset \mathcal{K}_x^{\mathrm{r}}$ for any $x \in \mathcal{T}_c(y)$ ancestor of y. The observation is valid for any root of $\mathcal{T}(P)$.*

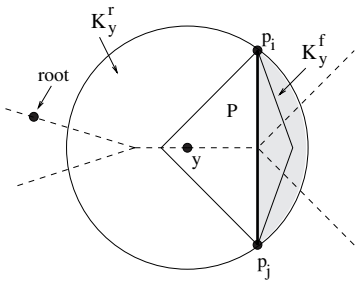

Fig. 4. $\mathcal{K}_y$ is partitioned into $\mathcal{K}_y^{\mathrm{r}}$ and $\mathcal{K}_y^{\mathrm{f}}$ shown shaded.

PROOF. Let $x \in \mathcal{T}(y)$ (see Figure 5(a)). By definition, the P-circle $\mathcal{K}_x$ must pass through points p_{k}, p_{r} on the convex hull of P such that p_{k}, $p_{\mathrm{r}} \in \mathcal{K}_{\mathrm{y}}^{\mathrm{r}}$. (In Figure 5(a) $p_{\mathrm{r}} = p_i$.) Then chord $\overline{p_{\mathrm{k}} p_{\mathrm{r}}}$ partitions $\mathcal{K}_x$ into two parts, $\mathcal{K}_x^1$ and $\mathcal{K}_x^2$, such that $\mathcal{K}_x^1 \subset \mathcal{K}_{\mathrm{y}}^{\mathrm{r}}$ and $\mathcal{K}_{\mathrm{y}}^{\mathrm{f}} \subset \mathcal{K}_x^2$. But by definition, $\mathcal{K}_x^1 = \mathcal{K}_x^{\mathrm{r}}$ and $\mathcal{K}_x^2 = \mathcal{K}_x^{\mathrm{f}}$. Let now $x \in \mathcal{T}_{\mathrm{c}}(y)$ (see Figure 5(b)). $\mathcal{K}_x$ must pass through points p_{s}, $p_{\mathrm{t}} \in \mathcal{K}_{\mathrm{y}}^{\mathrm{f}}$ and chord $\overline{p_{\mathrm{s}} p_{\mathrm{t}}}$ partitions $\mathcal{K}_x$ into two parts such that $\mathcal{K}_{\mathrm{y}}^{\mathrm{r}} \subset \mathcal{K}_x^1$ and $\mathcal{K}_x^2 \subset \mathcal{K}_{\mathrm{y}}^{\mathrm{f}}$. For $x \in \mathcal{T}_{\mathrm{c}}(y)$ such that x is not an ancestor of y, as shown in Figure 5(b), $\mathcal{K}_x^1 = \mathcal{K}_x^{\mathrm{f}}$ and $\mathcal{K}_x^2 = \mathcal{K}_x^{\mathrm{r}}$. In Figure 5(b), $p_{\mathrm{t}} = p_{\mathrm{j}}$, $\mathcal{K}_x^{\mathrm{r}}$ and $\mathcal{K}_{\mathrm{y}}^{\mathrm{r}}$ are shown shaded. For x being an ancestor of y, $\mathcal{K}_x^1 = \mathcal{K}_x^{\mathrm{r}}$ and $\mathcal{K}_x^2 = \mathcal{K}_x^{\mathrm{f}}$ (this can also be derived by the first statement). Thus, the second statement is also derived. □

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 points p_i, p_j, q_i, q_j are on 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 there is no pair of crossing chords between each other.

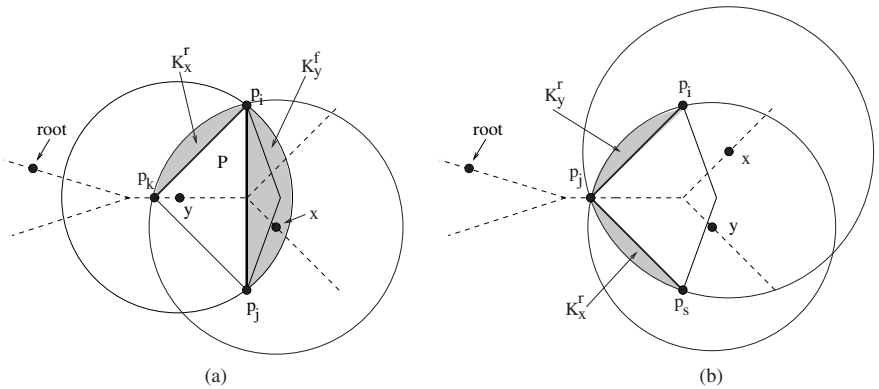

Fig. 5. Proof of Lemma 1. (a) $x \in \mathcal{T}(y)$. (b) $x \in \mathcal{T}_c(y)$.

It is easy to see that two clusters P and Q are non-crossing if and only if the convex hull of $P \cup Q$ contains at most two *supporting segments* between $CH(P)$ and $CH(Q)$. A segment $\overline{p_i q_j}$ such that $p_i \in P$ and $q_j \in Q$ is called a *supporting segment* between clusters P and Q if and only if $\overline{p_i q_j}$ is a segment of $CH(P \cup Q)$.

DEFINITION 2. Let S be a set of point clusters. The Hausdorff Voronoi region of cluster $P \in S$ is defined as

$$hreg(P) = \{x \mid d_f(x, P) < d_f(x, Q), \forall Q \in S\},$$

$hreg(P)$ may be disconnected and is further partitioned into finer regions by *f-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* between different clusters of S, and the inner edges consist of portions of *intra-bisectors* among 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. Figures 6 and 7, borrowed from [14], illustrate examples of Hausdorff Voronoi diagrams. Inter-bisectors are illustrated by solid lines and intra-bisectors are depicted by dashed lines. In Figure 6 the shaded regions depict $hreg(P_1)$ and $hreg(P_3)$. In Figure 7 the shaded region depicts the (disconnected) Hausdorff Voronoi region of segment Q.

Mixed Voronoi vertices encode the special properties of the Hausdorff Voronoi diagram and we consider them throughout this paper. 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 $q_i \in Q$, where $Q \neq P$, and $p_i, p_j, q_i \in CH(P \cup Q)$, such that $\mathcal{K}_v$ entirely encloses both P and Q but no other cluster in S. Vertex v is said to be *attributed* to the pair of supporting segments $(s_i, s_j) \in CH(P \cup Q)$ such that s_i (resp. s_j) is the first supporting segment between P and Q encountered as we traverse $CH(P \cup Q)$ from q_i to p_i (resp. p_j) (see Figure 8). For brevity, and since the vertices of $b_h(P, Q)$ are mixed Voronoi vertices of *H-Vor*$(\{P, Q\})$,

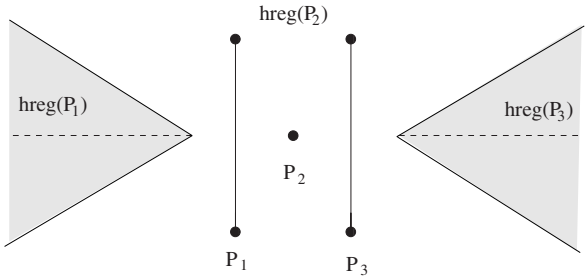

Fig. 6. The Hausdorff Voronoi diagram of non-crossing clusters P_1, P_2, P_3 [14].

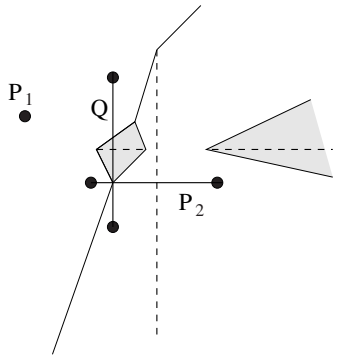

Fig. 7. The Hausdorff Voronoi diagram of clusters P_1, P_2, Q [14].

we refer to the vertices of $b_{\mathrm{h}}(P, Q)$ as *mixed vertices*. Mixed vertices are classified into *crossing*, *non-crossing*, *rear* and *forward* as given by the following definition.

DEFINITION 3. A mixed Voronoi vertex v induced by $p_i, p_j \in P$ and $q_i \in Q$ is called *crossing* (resp. *non-crossing*) iff cluster Q is crossing (resp. non-crossing) with $\overline{p_i p_j}$. Vertex v is characterized as *rear* (resp. *forward*) iff $q_i \in \mathcal{K}_v^r$ (resp. $q_i \in \mathcal{K}_v^f$) where $v \in b(p_i, p_j)$. In general, any point q enclosed in $\mathcal{K}_v^r$ (resp. $\mathcal{K}_v^f$) is called *rear* (resp. *forward*) with respect to v, intra-bisector $b(p_i, p_j)$, and the root of $\mathcal{T}(P)$.

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 $\mathcal{T}(P)$ by $b_{\mathrm{h}}(P, Q)$. We have the following properties*:

- *For any rear non-crossing vertex v, $\mathcal{T}_c(v) \cap hreg(P) = \emptyset$. Thus, $\mathcal{T}(P)$ can contain at most one rear non-crossing mixed Voronoi vertex of H-Vor(S).*
- *For any forward non-crossing vertex v, $\mathcal{T}(v) \cap hreg(P) = \emptyset$. Thus, $\mathcal{T}(P)$ can contain at most $|\mathcal{T}(P)|$ forward non-crossing mixed Voronoi vertices, at most one for each unbounded segment of $\mathcal{T}(P)$.*

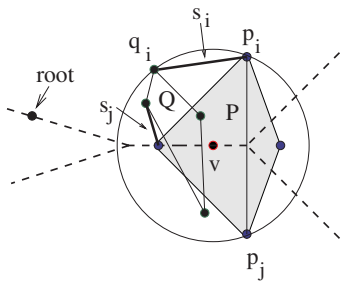

Fig. 8. Mixed vertex v is attributed to the pair of supporting segments (s_i, s_j).

- *Any crossing forward vertex on $\mathcal{T}(P)$ must be followed by a crossing rear vertex (considering only vertices of $b_{\mathrm{h}}(P, Q)$).*
- *Any rear (resp. forward) vertex delimits the beginning (resp. ending) of a component of $\mathcal{T}(P) \cap hreg(P)$ as we traverse $\mathcal{T}(P)$ from the root to the leaves.*

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

Consider a path from the root to a leaf of $\mathcal{T}(P)$ that contains vertices of $b_{\mathrm{f}}(P, Q)$. Let y_1 (resp. y_k) be the first (resp. last) vertex along this path bounding $hreg(P)$ in *H-Vor*$(\{P, Q\})$. Vertex y_1 may be the root of $\mathcal{T}(P)$ (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 induced by $p_i, p_j \in P$, $q_i \in Q$. If v is delimiting the ending of $hreg(P)$ and the begining of $hreg(Q)$ as we walk from y_1 to y_k, then q_i must be part of $\mathcal{K}_v^{\mathrm{f}}$, i.e., v must be forward. On the contrary, if v is delimiting the begining of $hreg(P)$ and the ending of $hreg(Q)$, v must be part of $\mathcal{K}_v^{\mathrm{r}}$, i.e., v must be rear. Thus, all the vertices between y_1 and y_k must be alternating between forward and rear. Since y_k must be forward, any crossing forward mixed vertex on the path must be followed by a rear crossing mixed vertex. This also derives the last statement. □

LEMMA 3. *Every rear vertex of $b_{\mathrm{h}}(P, Q)$ incident to $\mathcal{T}(P)$ is attributed to a unique pair of supporting segments between P and Q that are are entirely enclosed in the P-circle centered at the root of $\mathcal{T}(P)$ and appear consecutively on* $\mathrm{CH}(P \cup Q)$.

PROOF. Let v_i be a rear mixed vertex on $b_{\mathrm{h}}(P, Q)$ induced by $p_i, p_j \in P$ and $q_i \in Q$ and let (s_i, s_j) be the pair of supporting segments on $CH(P \cup Q)$ where v_i is attributed to. Recall that s_i (resp. s_j) is the first supporting segment between P and Q encountered as we traverse $CH(P \cup Q)$ from q_i to p_i (resp. p_j), and thus, s_i, s_j must appear consecutively on $CH(P \cup Q)$. Since v_i is rear, $q_i \in \mathcal{K}_{v_i}^{\mathrm{r}}$, and thus $s_i, s_j \in \mathcal{K}_{v_i}^{\mathrm{r}}$. However, by Lemma 1, $\mathcal{K}_{v_i}^{\mathrm{r}} \subset \mathcal{K}_{y_0}$, where y_0 is the root of $\mathcal{T}(P)$, and thus $s_i, s_j \in \mathcal{K}_{y_0}$. Suppose, for a contradiction, that there is another rear mixed vertex v_j, $v_j \neq v_i$, such that v_j is also attributed to (s_i, s_j). Then v_j must be induced by $q_i' \in Q$, $p_i', p_j' \in P$ such that q_i' (resp. p_i', p_j') is in the same component of $CH(P \cup Q)$ (as defined by supporting segments s_i and s_j) as q_i (resp. p_i, p_j). Triangle (q_i, p_i, p_j) partitions $\mathcal{K}_{v_i}$ into three slices where q_i', p_i', p_j' may belong. Since both v_i and v_j are rear, not all three q_i', p_i', p_j' can belong in the same slice. However, the circle through q_i', p_i', p_j' cannot enclose the entire triangle (q_i, p_i, p_j) and thus v_j cannot be a mixed vertex. □

DEFINITION 4. Let P, Q be a pair of crossing clusters and let (s, s') be a pair of supporting segments between P and Q such that there is a crossing vertex of $b_h(P, Q)$ that can be attributed to (s, s') and such that both s and s' are enclosed in the minimum enclosing circle of P. Segments s, s' are called *crucial supporting segments* for P and the pair (s, s') is called a *crucial pair* of supporting segments. The total number of crucial supporting segments between pairs of crossing clusters is denoted by m.

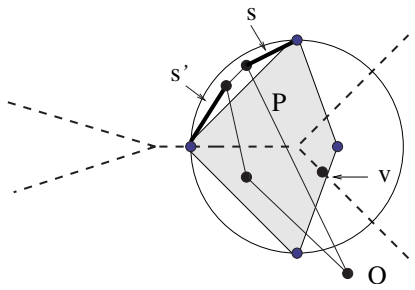

Fig. 9. A crucial pair of supporting segments (s, s').

Figure 9 illustrates the crucial pair of supporting segments attributed to a crossing mixed vertex v. We now and for the remaining of this section assume that the root of an intra-bisector tree $\mathcal{T}(P)$ is the center of the minimum enclosing circle of P. Then, by Lemma 3 and Definition 4, there is a 1–1 correspondence between rear crossing mixed vertices on $b_{\mathrm{h}}(P, Q)$ and crucial pairs of supporting segments between P and Q.

LEMMA 4. *The number of rear crossing mixed vertices on any inter-bisector $b_{\mathrm{h}}(P, Q)$ is exactly half the number of crucial supporting segments between P and Q. (The term rear assumes that the root of $\mathcal{T}(P)$, $P \in S$, is the center of the minimum enclosing circle of P.)*

PROOF. Let (s, s') be a crucial pair of supporting segments corresponding to crossing vertex v_i, induced by p_i, $p_j \in P$ and $q_i \in Q$. By definition, s and s' must be consecutive on $CH(P \cup Q)$. Let s'' be the supporting segment on $CH(P \cup Q)$ following s on the opposite direction from s'. Suppose for a contradiction that there is a rear mixed vertex v_j attributed to (s, s''). Then v_j would be induced by $q_j, q_k \in Q$ such that chord $\overline{q_j q_k}$ intersects $\overline{p_i p_j}$. However, then it is easy to see that there cannot be a P-circle through p_i, p_j and a Q-circle through q_j, q_k such that both circles enclose P and Q. Thus, no mixed vertex can be attributed to (s, s''), i.e., s can be part of exactly one crucial pair of supporting segments. The claim is now evident from Lemma 3. □

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

PROOF. As was shown in [14], the total number of vertices of *H-Vor*(*S*) is proportional to the number of mixed Voronoi vertices in *H-Vor*(*S*). By Lemma 2, the number of non-crossing mixed Voronoi vertices is $O(n)$. Also by Lemma 2 (third statement) the number of crossing mixed Voronoi vertices is proportional to the number of rear crossing mixed Voronoi vertices (for any convention of the root of intra-bisector trees). Let the root of $\mathcal{T}(P)$, $P \in S$, be the center of the minimum enclosing circle of P. Then, by Lemma 4, the number of rear crossing mixed Voronoi vertices, and therefore the total number of crossing mixed Voronoi vertices, is $O(m)$.

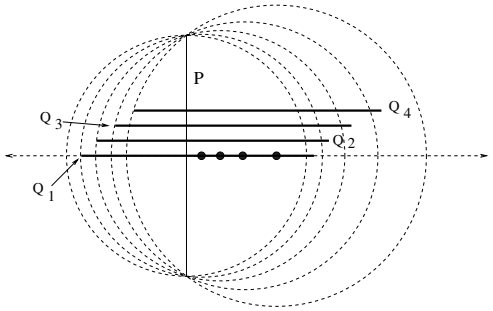

Fig. 10. The $\Omega(m)$ construction for the complexity of *H-Vor*(S).

To obtain the lower bound it is enough to construct a set S such that every rear crossing vertex of $b_h(P, Q)$, for any pair of crossing clusters $P, Q \in S$, remains a Voronoi vertex in *H-Vor*(S), as by Lemma 4, this number is $\Theta(m)$. Throughout this proof, any intra-bisector tree $\mathcal{T}(P)$ is assumed to be rooted the center of the minimum enclosing circle of P. The $\Omega(n)$ lower bound is trivial.

Consider a vertical line segment P and its minimum enclosing circle, $\mathcal{K}_0$. Let $\varepsilon > 0$ be a small constant and let l_i, $1 < i \leq k$, be a set of horizontal lines, each $\varepsilon > 0$ 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 10). 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 $\mathcal{K}_{i-1}$, ε away from the boundary of $\mathcal{K}_{i-1}$. The right endpoint of Q_i is in the exterior of $\mathcal{K}_{i-1}$, $\varepsilon/2$ 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 centers of R_i and $\mathcal{K}_i$ remain mixed Voronoi vertices in *H-Vor*(S) for any i, $1 \leq i \leq k$. In particular, the centers of R_i, $1 \leq i \leq k$, correspond to forward crossing mixed Voronoi vertices delimiting the ending of a component of *hreg*(P) on $\mathcal{T}(P)$, and the centers of K_i, $1 \leq i \leq k$, correspond to rear crossing mixed Voronoi vertices delimiting the begining of a component of *hreg*(P). Since every Q_i is crossing with P and there is a P-circle enclosing every Q_i, it is easy to see that there can be no Q_i-circle enclosing P (see also Lemma 4). Hence, the centers of P-circles K_i, $1 \leq i \leq k$, are the only rear crossing vertices for all inter-bisectors between any two crossing clusters in S, and they all remain as mixed Voronoi vertices in *H-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 11. 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. □

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 [3] and [5] but requires special *events* to handle mixed Voronoi

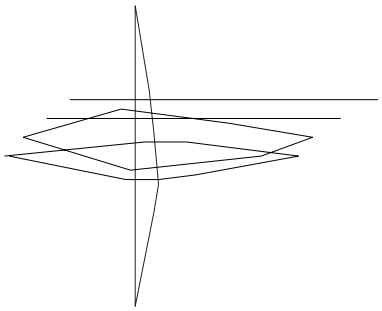

Fig. 11. Segments can be substituted by convex polygons of arbitrarily many points.

vertices and disconnected Voronoi regions. Note that disconnected Voronoi regions have not been considered by plane sweep approaches so far. 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 construction 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 $d_{\mathrm{h}}(p, l_t) = d(p, l_t)$. Following the terminology of [3], 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 *waves*, corresponding to ordinary bisectors between points $p \in S_t$ and the sweep line l_t. In the L_2 metric the waves are parabolic arcs. 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 at specific *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 from the ordinary plane sweep and their handling is illustrated in the remander of this section.

DEFINITION 5. The *priority* of any point $v \in H\text{-}Vor(S)$ or $v \in f\text{-}Vor(P)$, $P \in S$, is $priority(v) = x(v) + d(v, p_i)$, 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, $priority(v)$ equals the rightmost x-coordinate of the circle centered at v passing through p_i.

The point of minimum priority for any cluster P, denoted by $\tilde{y}_0(P)$ (for brevity $\tilde{y}_0$), is clearly the intra-bisector point derived by shooting a horizontal ray backwards from the rightmost point p_{r} of P, until it hits the boundary of $freg(p_{\mathrm{r}})$ in $f\text{-}Vor(P)$. Clearly,

priority$(\tilde{y}_0) = x(p_r)$. The P-circle centered at $\tilde{y}_0$ is referred to as the *minimum priority* circle of P. If P has more than one rightmost point, i.e., if p_r is incident to a vertical segment $\overline{p_r p'_r}$ in $CH(P)$, then the horizontal ray is identical to intra-bisector $b(p_r, p'_r)$, and $\tilde{y}_0$ is a point at infinity. Throughout this section (unless explicitly noted otherwise) we assume that the root of $\mathcal{T}(P)$ is $\tilde{y}_0$, the point of minimum priority of P. As a result, the definition of a rear/forward mixed Voronoi vertex or a rear/forward point always assumes that the corresponding intra-bisector tree is rooted at $\tilde{y}_0(P)$. The following lemma shows that the priority of $\mathcal{T}(P)$ is monotonically increasing as we move from the root to the leaves.

LEMMA 5. *Let $y_i \in \mathcal{T}(P)$. Then priority$(y_i) <$ priority(y) for any $y \in \mathcal{T}(y_i)$ (assuming that the root of $\mathcal{T}(P)$ is the point of minimum priority $\tilde{y}_0(P)$).*

PROOF. The priority of any $y_i \in \mathcal{T}(P)$ is given by the rightmost vertical line l tangent to $\mathcal{K}_{y_i}$. Since *priority*$(\tilde{y}_0) \leq$ *priority*(y_i), and by Lemma 1 $\mathcal{K}^r_{y_i} \subset \mathcal{K}_{\tilde{y}_0}$, l must be tangent to $\mathcal{K}^f_{y_i}$. However by Lemma 1, $\mathcal{K}^f_{y_i} \subset \mathcal{K}^f_y$ for every $y \in \mathcal{T}(y_i)$. Thus, *priority*$(y_i) <$ *priority*(y). □

By the definition of the wavefront, any Voronoi point $v \in$*H-Vor*(S) must enter the wavefront at time $t =$ *priority*(v). Thus, appropriate events need to be generated so that at least one event exists (site or spike event) for every vertex of *H-Vor*(S). We define two types of site events: *ordinary-vertex events*, one for every vertex of $\mathcal{T}(P)$, $P \in S$, and *mixed-vertex events* (for brevity *mixed events*) whose purpose is to predict the rear mixed Voronoi vertices of *H-Vor*(S). Mixed events are also used to identify disconnected Hausdorff Voronoi regions. Ordinary events are readily available from *f-Vor*(P), $P \in S$, and mixed events get generated throughout the algorithm. An event is called *valid* if it falls on or ahead of the wavefront at the time of its priority. An event falling behind the wavefront at the time of its priority is called *invalid*. The main difficulty encountered here is that site events need not always be valid, furthermore, invalid site events may still correspond to non-empty Voronoi regions. For example, even in the non-crossing case, when the priority of a cluster P is reached at time $t =$ *priority*$(\tilde{y}_0)$, point $\tilde{y}_0$ may fall behind the wavefront although *hreg*$(P) \neq \emptyset$. To predict the actual time when *hreg*(P) appears on the wavefront for the first time, a mixed-vertex event gets generated. Similarly, in the general crossing case where Hausdorff Voronoi regions may be disconnected, mixed events get generated to predict the time when a connected component of *hreg*(P) may appear on the wavefront for the first time. The following lemma shows that the minimum priority of any connected component of *hreg*(P) must appear at a rear mixed Voronoi vertex, unless $\tilde{y}_0 \in$ *hreg*(P). As a result, if there exists a mixed-vertex event for every rear mixed Voronoi vertex in *H-Vor*(S), no component of *hreg*(P) can be missed even if *hreg*(P) is disconnected. Mixed events get generated at invalid site events. In the case of crossing clusters, mixed events are also generated at valid spike events. Throughout the plane sweep, any event is processed at the time of its priority.

LEMMA 6. *Let $hreg_i(P)$ be a connected component of hreg(P), $P \in S$. The point of minimum priority for $hreg_i(P)$ must always occur at an intra-bisector point of $\mathcal{T}(P) \cap$*

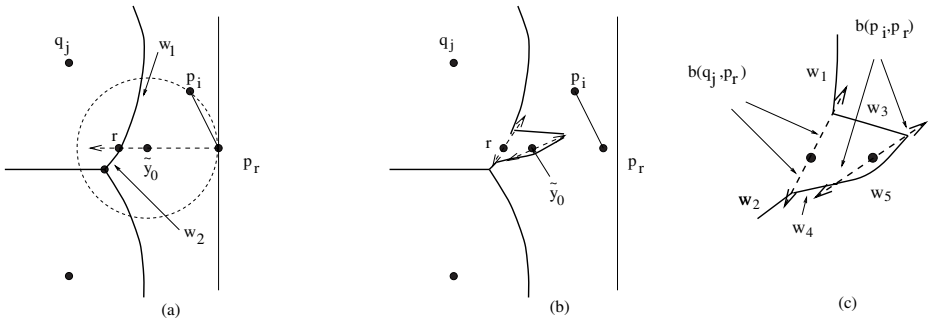

Fig. 12. The wavefront update at a valid minimum priority vertex event.

$hreg_i(P)$. This is either $\tilde{y}_0$, if $\tilde{y}_0 \in hreg_i(P)$, or a rear mixed Voronoi vertex, otherwise. All remaining mixed Voronoi vertices of $\mathcal{T}(P) \cap hreg_i(P)$, except the one realizing the minimum priority of $hreg(P)$, must be forward.

PROOF. Let u be any point in $hreg_i(p)$, $p \in P$, and let y be the intersection point of the boundary of $freg(p)$ and $\overline{pu}$. By definition, $\mathcal{K}_y$ must be entirely enclosed in $\mathcal{K}_u$ and thus $priority(y) < priority(u)$. Thus, the minimum priority of $hreg_i(P)$ must occur on $\mathcal{T}(P)$. By Lemma 2, this must be a rear mixed vertex on $\mathcal{T}(P)$, unless $\tilde{y}_0 \in hreg_i(P)$. Since, by Property 3 of [14], $\mathcal{T}(P) \cap hreg_i(P)$ is a single connected component, by Lemma 2, any other mixed vertex of $\mathcal{T}(P) \cap hreg_i(P)$ must be forward. □

We now discuss in detail the handling of the various kinds of site events. We first consider the vertex event corresponding to the minimum priority point of P, $\tilde{y}_0$, and assume that the event is valid. Let $\overline{p_i p_r}$ be the chord inducing $\tilde{y}_0$ (i.e., $\tilde{y}_0 \in b(p_i, p_r)$), 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 spike 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 12). If $\overline{p_i p_r}$ is vertical, let the horizontal ray be $b(p_i, p_r)$. 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 and w_4 for p_r, and wave w_5 for p_i gliding along the two rays of intra-bisector $b(p_i, p_r)$ emanating from $\tilde{y}_0$. The ordering of the waves from top to bottom is w_1, w_3, w_5, w_4, w_2. Figure 12(a) and (b) depicts the wavefront before and after the update, respectively, and Figure 12(c) shows the topological arrangement of the new waves. Spike events corresponding to intersections of new neighboring spike bisectors are generated as in the ordinary Voronoi diagram construction.

The update of the wavefront at any other valid ordinary-vertex event $y_i \in \mathcal{T}(P)$, $y_i \neq \tilde{y}_0$, is similar and is depicted in Figure 13. In detail, let $p_i, p_j, p_k \in P$ be the points inducing vertex y_i in *f-Vor*(P). 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 $b(p_i, p_j)$. At time t, a new wave for point p_k must enter the wavefront gliding between the spike intra-

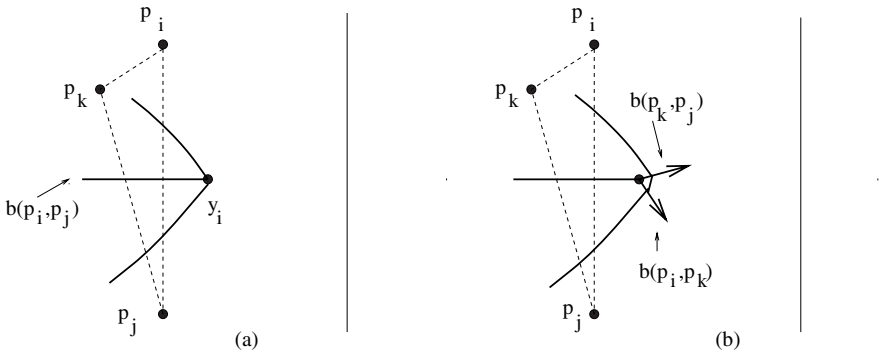

Fig. 13. The wavefront update at an ordinary valid vertex event.

bisectors $b(p_k, p_i)$ and $b(p_k, p_j)$ as shown in Figure 13. Note that there is no need for the new spike bisectors $b(p_k, p_i)$ and $b(p_k, p_j)$ to point toward the right of the vertical line through y_i as shown in Figure 13, but the priority of their endpoint must be increasing as the sweep line moves to the right.

We now consider the handling of an invalid vertex event $y_i \in \mathcal{T}(P)$ and the generation of a mixed vertex event. Note that y_i may be an ordinary- or a mixed-vertex event including $\tilde{y}_0$. At time $t = priority(y_i)$, point y_i is behind the wavefront (see Figure 14). Let y_j be any immediate descendent of y_i in $\mathcal{T}(P)$ 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_t. The following process repeats for all immediate descendents of y_i that are not yet covered by the wavefront. Let $q_j \in Q$ be the owner of y_t in $H\text{-}Vor(S_t)$ and let $\overline{y_i y_j} \in b(p_i, p_j)$. As will be shown in Lemma 7, there is a rear mixed vertex m_i, induced by Q on $\overline{y_t y_j}$, if and only if $d_f(y_j, P) < d_f(y_j, Q)$. Vertex m_i (if it exists) can be easily determined in $O(|CH(Q)|)$ time by considering intersections of $\overline{y_t y_j}$ with $f\text{-}Vor(Q)$. If Q is crossing $\overline{p_i p_j}$, then it is essential to start the search at y_j and not at y_t to maintain the time complexity bound, as

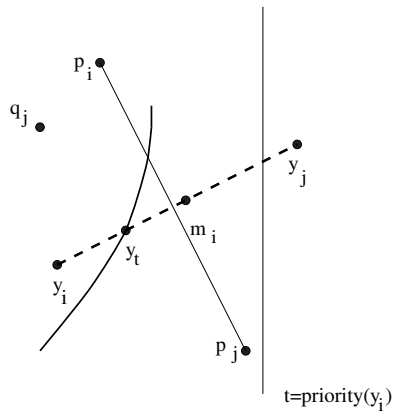

Fig. 14. The generation of a mixed-vertex event.

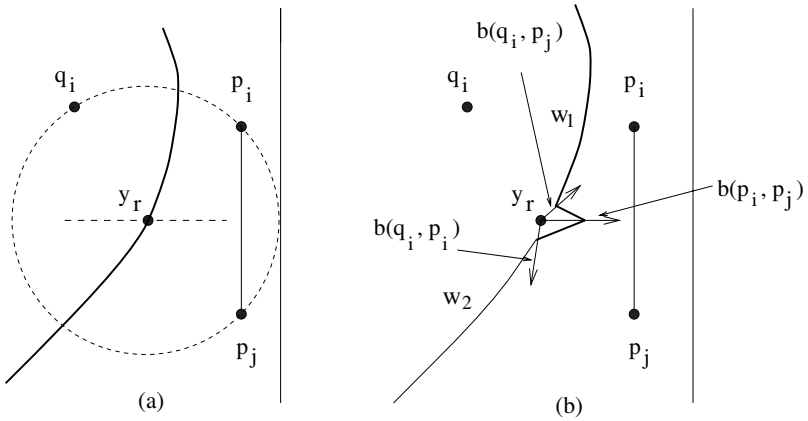

Fig. 15. The update of the wavefront at a valid mixed-vertex event.

will be shown in Lemma 9. If m_i exists, a mixed-vertex event gets generated for m_i. In addition, if Q is non-crossing with $\overline{p_i p_j}$, we can eliminate from further consideration the entire $\mathcal{T}(y_t)$ or the entire $\mathcal{T}_c(y_t)$ depending on whether q_j is forward or rear respectively with respect to y_t. To eliminate $\mathcal{T}(y_t)$ or $\mathcal{T}_c(y_t)$ we simply eliminate from the event queue all vertex events associated with that subtree.

The handling of a mixed-vertex event is similar to the handling of an ordinary-vertex event. In particular, let $y_r \in \mathcal{T}(P)$ be a mixed-vertex event induced by $p_i, p_j \in P, q_i \in Q$. Event y_r is valid if and only if at time $t = priority(y_r)$, 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 $b(p_i, p_j)$, as shown in Figure 15. If y_r is invalid, then a new mixed-vertex event may be generated on $b(p_i, p_j)$, exactly as described in the preceding paragraph.

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 in order to predict potential disconnected Voronoi regions. A spike event between a spike intra-bisector $b(p_i, p_j)$, $p_i, p_j \in P$, and a spike inter-bisector $b_h(P, Q)$ corresponds to a forward mixed vertex v on $\mathcal{T}(P)$. If Q is crossing with $\overline{p_i p_j}$, then v may be followed on $\mathcal{T}(P)$ by a rear mixed Voronoi vertex u. Thus, to predict u, a mixed event induced by Q on $b(p_i, p_j)$ may get generated, similarly to an invalid ordinary-vertex event. If Q is non-crossing with $\overline{p_i p_j}$, then $\mathcal{T}(v) \cap hreg(P) = \emptyset$, and thus all vertex events associated with $\mathcal{T}(v)$ can be eliminated in this case.

The correctness of the algorithm follows from the correctness of the plane sweep paradigm [3] as long as we can show that no rear mixed Voronoi vertex can be missed. By Lemma 5, a starting point for all connected components of Voronoi regions can be obtained if we have events for all rear mixed Voronoi vertices and all roots of intra-bisector trees. The following lemma shows that mixed events get correctly generated and therefore the correctness of the algorithm can be derived.

LEMMA 7. *Let $y_i \in \mathcal{T}(P)$ 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 $\mathcal{T}(P)$ and let y_t be the first*

intersection of $\overline{y_i y_j}$ with the wavefront at time $t = priority(y_i)$. If a mixed-vertex event m_i gets generated on segment $\overline{y_t y_j}$ during the handling of y_i, then $\overline{y_t m_i} \cap hreg(P) = \emptyset$. If no mixed-vertex event gets generated on $\overline{y_t y_j}$, then $\overline{y_t y_j} \cap hreg(P) = \emptyset$.

PROOF. Assume that the plane sweep construction is correct until time $t = priority(y_i)$. Let $\overline{y_i y_j}$ be induced by $p_i, p_j \in P$. By the definition of the wavefront and point y_t, $d_f(y_t, Q) < d_f(y_t, P)$. We first show that $b_h(P, Q)$ may intersect $\overline{y_t y_j}$ at most once and that the intersection point (if any) must be a rear mixed vertex. Let m_i be a mixed vertex of $b_h(P, Q)$ on $\overline{y_t y_j}$ induced by $q_i \in Q$. Suppose for a contradiction that m_i is forward. Then, by Lemma 1, for any $y \in \overline{y_i m_i}$, $q_i \notin \mathcal{K}_y$, and thus $d_f(y, P) < d_f(y, Q)$. However, $y_t \in \overline{y_i m_i}$ and $d_f(y_t, Q) < d_f(y_t, P)$. Thus, m_i must be rear, i.e., $q_i \in \mathcal{K}^r_{m_i}$. Nevertheless then, by Lemma 1, for any $y \in \overline{m_i y_j}$ other than m_i, $q_i \notin \mathcal{K}_y$, and thus $d_f(y, P) < d_f(y, Q)$. Hence, m_i must be unique and it exists if and only if $d_f(y_j, P) < d_f(y_j, Q)$, i.e., if and only if m_i is generated by the algorithm. By the above, $\overline{y_t m_i} \cap hreg(P) = \emptyset$, if m_i exists, and $\overline{y_t y_j} \cap hreg(P) = \emptyset$, otherwise. □

The time complexity of the plane sweep algorithm 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 6. Let r be the horizontal ray extending backwards from the rightmost point p_r of P. If p_r is not unique, let r be the bisector of the rightmost vertical segment of $CH(P)$. Let C_r be the circle centered on r, entirely to the left of the sweep line l_t, touching the sweep line at point $l_t \cap r$, such that there is exactly one cluster in the interior of C_r (in the worst case that cluster is P). The cluster contained in C_r is called the *anchor* of P.

DEFINITION 7. Let A be the anchor of P. Let $\tilde{y}_a$ be the point of minimum priority of P if $A = P$, and let $\tilde{y}_a$ be the nearest common ancestor on $\mathcal{T}(P)$ of all rear vertices of $b_f(P, A)$ incident to $\mathcal{T}(P)$, if $A \neq P$. The P-circle centered at $\tilde{y}_a$ is called the *anchor circle* of P. The anchor circle of P is denoted as $\mathcal{K}_a(P)$ and has priority $priority(\tilde{y}_a)$.

It is easy to see, by the definition of $\tilde{y}_a$, that $\mathcal{T}_c(\tilde{y}_a) \cap hreg(P) = \emptyset$. For the sake of formally bounding the number of mixed-vertex events, we can indeed identify A and $\tilde{y}_a$ during the handling of the minimum priority event for P, create a mixed event for $\tilde{y}_a$, and eliminate $\mathcal{T}_c(\tilde{y}_a)$ with no change in the behavior of the algorithm. To eliminate $\mathcal{T}_c(\tilde{y}_a)$ it is enough to delete from the event queue all vertex events associated with $\mathcal{T}_c(\tilde{y}_a)$. To determine A we first observe that the minimum priority point $\tilde{y}_0(P)$ is included in $hreg(P)$ if and only if $A = P$. Thus, if $\tilde{y}_0$ is valid, let $A = P$, otherwise, let A be the owner of the wave hit by the horizontal ray r of Definition 6. To determine $\tilde{y}_a$, if $A \neq P$, we determine all rear mixed vertices induced by A on $\mathcal{T}(P)$ by traversing $\mathcal{T}(P)$ though *f-Vor*(A), similarly to the generation of a mixed-vertex event during the handling of an invalid event described above (see also Lemma 9). If $A = P$, let $\tilde{y}_a = \tilde{y}_0$.

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 the bound on the size of H-Vor*(S) *as given 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 cluster Q on $\mathcal{T}(P)$. By Lemma 2, m_i is unique. Assume that during the handling of the minimum priority event for P, $\mathcal{T}_c(\tilde{y}_a)$ has been eliminated. Then $m_i \in \mathcal{T}(\tilde{y}_a)$ and thus, by Lemma 1, $\mathcal{K}^r_{m_i} \subset \mathcal{K}_a(P)$. Since m_i corresponds to a rear non-crossing mixed vertex, $Q \in \mathcal{K}^r_{m_i} \cup CH(P)$, and thus $Q \in \mathcal{K}_a(P)$. Hence, the total number of non-crossing mixed-events is $O(K)$. The total number of crossing mixed-vertex events is by construction $O(m)$. □

LEMMA 9. *The generation time for all mixed-vertex events induced on* $\mathcal{T}(P)$ *by a single cluster* Q *is* $O(|CH(Q) \cap \mathcal{K}_a(P)|)$, *where* $\mathcal{K}_a(P)$ *is the anchor circle of* P.

PROOF. We assume that the anchor of P is identified during the handling of the minimum priority event for P and that $\mathcal{T}_c(\tilde{y}_a)$ has been eliminated. The claim is easy to see for a cluster Q that is non-crossing with P, as Q can induce at most one mixed event on $\mathcal{T}(P)$, and Q must be entirely contained in $\mathcal{K}_a(P)$ (see the proof of Lemma 8). For a cluster Q crossing with P, Q may induce several crossing mixed-vertex events on $\mathcal{T}(P)$. However, we claim that for any $q_r \in Q$ that gets visited during the generation of a single mixed-vertex event m_i, q_r cannot be visited again during the generation of another mixed-vertex event m_j on $\mathcal{T}(P)$. Furthermore, $q_r \in \mathcal{K}_a(P)$.

Let $q_r \in Q$ be visited during the generation of $m_i \in \overline{y_i y_j} \in \mathcal{T}(P)$. Since q_r gets visited, segment $\overline{m_i y_j}$ must intersect $freg(q_r)$ in *f-Vor*(Q), as the traversal of $\overline{y_i y_j}$ starts at y_j and not at y_i, where y_i is the ancestor of y_j in $\mathcal{T}(P)$. By the proof of Lemma 7, $Q \in \mathcal{K}_{m_i}$ but $Q \notin \mathcal{K}_y$ for any $y \in \overline{m_i y_j}$, other than m_i. Thus, $q_r \in \mathcal{K}_{m_i}$ but $q_r \notin \mathcal{K}_y$ for any $y \in \overline{m_i y_j} \cap \text{freg}(q_r)$, other than m_i. Since, by Lemma 1, $\mathcal{K}^f_{m_i} \subset \mathcal{K}^f_y$ for any $y \in \overline{m_i y_j} \cap freg(q_r)$, $q_r \notin \mathcal{K}^f_{m_i}$ and thus $q_r \in \mathcal{K}^r_{m_i}$. However, by Lemma 1, $\mathcal{K}^r_v \subset \mathcal{K}_y$ for any $v \in \mathcal{T}(y)$, and thus $q_r \notin \mathcal{K}^r_v$ for any $v \in \mathcal{T}(y)$, where $y \in \overline{m_i y_j} \cap freg(q_r)$. In addition, by the proof of Lemma 1, for any $u \in \mathcal{T}_c(m_i)$ that is not an ancestor of m_i, $\mathcal{K}^r_{m_i} \subset \mathcal{K}^f_u$, and thus $q_r \in \mathcal{K}^f_u$, i.e., $q_r \notin \mathcal{K}^r_u$. Hence, q_r cannot be considered again during the generation of a mixed vertex event m_j on $T(m_i)$ or $T_c(m_i)$. Thus, q_r can be considered at most once during the generation of mixed vertex events induced by Q on $T(P)$. Since $m_i \in \mathcal{T}(\tilde{y}_a)$ and $q_r \in \mathcal{K}_{m_i}$, by Lemma 1, $q_r \in \mathcal{K}_a(P)$. □

THEOREM 2. *H-Vor*(S) *can be computed in* $O(M + (n+m+K)\log n)$ *time by plane sweep, where* $\Theta(n+m)$ *is the bound on the size of H-Vor*(S) *as defined in Theorem* 1, $K = \sum_{P \in S} K(P)$, *where* $K(P)$ *is the number of clusters entirely enclosed in the anchor circle of* P, *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 enclosed in the anchor circle of* P *or* Q *is crossing with* P.

PROOF. By the proof of Lemma 8, any cluster Q inducing a mixed-vertex event on $\mathcal{T}(P)$ is either entirely enclosed in the anchor circle of P or Q is crossing with P. Then,

by Lemma 9, the time attributed to the generation of all mixed-vertex events on $\mathcal{T}(P)$ is $O(M(P))$. Thus, the total time spent on the generation of mixed-vertex events is $O(M)$. To determine the anchor circle of P, it is enough to determine the rear mixed vertices of $b_{\mathrm{h}}(P, A)$ on $\mathcal{T}(P)$, where A is the anchor of P. However, by Lemma 9, this can be done in $O(|CH(P)| + |CH(A)|)$ time. Since $A \in \mathcal{K}_a(P)$, the anchor circles for all clusters in S can be determined in total $O(M)$ time. For each event (mixed or ordinary) $O(\log n)$ additional time is spent for the processing of the event. Determining whether a cluster Q is crossing a chord $\overline{p_i p_j} \in P$ can be done in $O(\log n)$ time as shown in [15]. Since the total number of mixed-vertex events is $O(K + m)$ (Lemma 8) and the total number of ordinary events is $O(n)$, the time complexity bound follows. Correctness follows from Lemma 7 and the standard arguments of plane sweep [3] as it is not hard to see that no rear mixed Voronoi vertex can be missed. □

5. Conclusion. We derived a tight combinatorial bound on the size of the Hausdorff Voronoi diagram of a set S of point clusters in the plane and presented a simple plane sweep algorithm for the construction of this diagram. The plane sweep construction has the ability to handle disconnected Hausdorff Voronoi regions, generalizing upon the standard plane sweep paradigm for the construction of Voronoi diagrams [5], [3]. More specifically, we showed that the size of the Hausdorff Voronoi diagram is $\Theta(n + m)$, where n is the number of points on the convex hulls of clusters in S, and m is the number of *crucial supporting segments* (see Definition 4) between pairs of crossing clusters. The plane sweep algorithm has 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, and $M = \sum_{P \in S} M(P)$, where $M(P)$ is the number of points $q \in CH(Q)$ that are enclosed in the *anchor circle* of P such that either Q is entirely contained in the *anchor circle* of P or Q is crossing with P. For a non-crossing set S the time complexity of the algorithm simplifies to $O((n + K)\log n)$. The *anchor circle* is a specially defined enclosing circle of P (see Definiton 7), in general different than the minimum enclosing circle of P. The plane sweep algorithm improves upon previous results, especially in the case of clusters with a small number of crossings, and it has comparable time complexity with the divide and conquer algorithm of our companion paper [15]. Furthermore, the plane sweep construction is simple. Whether the terms K and 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 and M remain small, most likely negligible in practice compared with n. Early experimental results reported in [12] on real VLSI data for the simpler L_∞ non-crossing case verified that K was indeed negligible compared with n [12].

References

[1] M. Abellanas, G. Hernandez, R. Klein,V. Neumann-Lara, and J. Urrutia, A combinatorial property of convex sets, *Discrete & Computational Geometry*, **17** (1997), 307–318.

[2] M. Abellanas, F. Hurtado, C. Icking, R. Klein, E. Langetepe, L. Ma, B. Palop, and V. Sacristán, The farthest color Voronoi diagram and related problems, *Abstracts*, 17*th European Workshop on Computational Geometry* (*CG* 2001), Freie Universität Berlin, Berlin, 2001, pp. 113–116.

[3] F. Dehne and R. Klein, The big sweep: on the power of the wavefront approach to Voronoi diagrams, *Algorithmica*, **17** (1997), 19–32.

[4] H. Edelsbrunner, L. J. Guibas, and M. Sharir, The upper envelope of piecewise linear functions: algorithms and applications, *Discrete & Computational Geometry*, **4** (1989), 311–336.

[5] S. J. Fortune, A sweepline algorithm for Voronoi diagrams, *Algorithmica*, **2** (1987), 153–174.

[6] D. P. Huttenlocher, K. Kedem, and M. Sharir, The upper envelope of Voronoi surfaces and its applications, *Discrete & Computational Geometry*, **9** (1993), 267–291.

[7] R. Klein, K. Mehlhorn, and S. Meiser, Randomized incremental construction of abstract Voronoi diagrams, *Computational Geometry*: *Theory and Applications*, **3** (1993), 157–184.

[8] I. Koren, The effect of scaling on the yield of VLSI circuits, in *Yield Modeling and Defect Tolerance in VLSI Circuits* (W. R. Moore, W. Maly, and A. Strojwas, eds.), Adam-Hilger, Bristol, 91–99, 1988.

[9] W. Maly, Computer Aided design for VLSI circuit manufacturability, *Proceedings IEEE*, **78**(2) (1990), 356–392.

[10] B. R. Mandava, Critical Area for Yield Models, IBM Technical Report TR22.2436, East Fishkill, NY, 12 Jan. 1982.

[11] C. H. Ouyang, W. A. Pleskacz, and W. Maly, Extraction of critical areas for opens in large VLSI circuits, *IEEE Transactions on Computer-Aided Design*, **18**(2) (1999), 151–162.

[12] E. Papadopoulou, Critical area computation for missing material defects in VLSI circuits, *IEEE Transactions on Computer-Aided Design*, **20**(5) (2001), 583–597.

[13] E. Papadopoulou and D. T. Lee, Critical area computation via Voronoi diagrams, *IEEE Transactions on Computer-Aided Design*, **18**(4) (1999), 463–474.

[14] E. Papadopoulou and D. T. Lee, The Min-Max Voronoi diagram of polygonal objects and applications in VLSI manufacturing, *Proceedings of the* 13*th International Symposium on Algorithms and Computation*, Nov. 2002, LNCS 2518, Springer-Verlag, Berlin, pp. 511–522.

[15] E. Papadopoulou and D. T. Lee, The Hausdorff Voronoi diagram of polygonal objects and applications in VLSI manufacturing, improved version of [14], submitted for publication.

[16] F. P. Preparata, and M. I. Shamos, *Computational Geometry*: *an Introduction*, Springer-Verlag, New York, 1985.

[17] C. H. Stapper, Modeling of defects in integrated circuits photolithographic patterns, *IBM Journal of Research and Development*, **28**(4) (1984), 461–475.

[18] I. A. Wagner and I. Koren, An interactive VLSI CAD tool for yield estimation, *IEEE Transaction on Semiconductor Manufacturing*, **8**(2) (1995), 130-138.

[19] H. Walker and S. W. Director, VLASIC: a yield simulator for integrated circuits, *IEEE Transactions on Computer-Aided Design*, **5**(4) (1986), 541–556.

[20] CAE: Critical Area Extraction, Internal IBM tool for the prediction of yield of VLSI chips, Department of Electronic Design Automation, IBM Micro-Electronics Division, Burlington, VT. Initial patents: US 6178539, US 6317859.