Map of Geometric Minimal Cuts for General Planar Embedding*

Lei Xu[1], Evanthia Papadopoulou[2], and Jinhui Xu[1]

[1] Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260, USA
{lxu,jinhui}@buffalo.edu
[2] Faculty of Informatics
Università della Svizzera italiana
Via Giuseppe Buffi 13
CH 6904 Lugano, Switzerland
evanthia.papadopoulou@unisi.ch

Abstract. In this paper, we consider the problem of computing the map of geometric minimal cuts (MGMC) induced by a general planar embedding (i.e., the edge orientation is either rectilinear or diagonal) of a subgraph $H = (V_H, E_H)$ of an input graph $G = (V, E)$. The MGMC problem is motivated by the critical area extraction problem in VLSI layout and finds applications in several other areas. In this paper, we extend an earlier result for planar rectilinear embedding to its more general case. The increased freedom on edge orientation in the embedding imposes new challenges, mainly due to the fact that the inducing region of a geometric minimal cut is no longer unique. We show that the MGMC problem can be solved by computing the L_∞ Hausdorff Voronoi diagram of a set of rectangle families, each containing an infinite number of axis-aligned rectangles. By exploiting the geometric properties of these rectangle families, we present an output-sensitive algorithm for computing the Hausdorff Voronoi diagram in this general case which runs in $O((N + K) \log^2 N \log \log N)$ time, where K is the complexity of the Hausdorff Voronoi diagram and N is the number of geometric minimal cuts.

1 Introduction

In this paper, we consider the following problem, called *Map of Geometric Minimal Cuts or MGMC* problem: Given a graph $G = (V, E)$ and an planar embedding of a subgraph $H = (V_H, E_H)$ of G with rectilinear or diagonal edges, compute a map $\mathcal{M}$ of the embedding plane P of H so that for every point $p \in P$, the cell in $\mathcal{M}$ containing p is associated with the "closest" *geometric cut* (in G) to p, where the distance between a point p and a cut C is defined as the maximum distance between p and any individual element of C. A geometric cut C

* The research of the third author was supported in part by NSF under grant IIS-1115220.

P. Widmayer, Y. Xu, and B. Zhu (Eds.): COCOA 2013, LNCS 8287, pp. 238–249, 2013.

of G is a set of edges and vertices in H that overlap a given geometric shape S in P and whose removal from G disconnects G. In this paper we consider the case where geometric cuts are induced by axis-aligned rectangles and distances are measured by the L_∞ metric. The main objective of the MGMC problem is to compute the map $\mathcal{M}$ of all geometric minimal (or canonical) cuts (the exact definition of geometric minimal cuts will be given in next section) of the planar embedding of H.

The MGMC problem was introduced in [5] motivated by the VLSI critical area computation problem. The critical area problem for various types of faults can be reduced to different variants of Voronoi diagrams that lead to accurate critical area extraction (see e.g., [4,6] and references therein). A VLSI net can be modeled as a graph $G = (V, E)$ with a subgraph embedded on every conducting layer. A subgraph $H = (V_H, E_H)$ on a layer X is vulnerable to random defects associated with layer X. Defects on layer X may create *cuts* on graph G that result in disconnecting the net N. The Voronoi framework for critical area extraction asks for a subdivision of layer X into regions that reveal for every point p the radius of the smallest disk centered at p inducing a cut of G.

The MGMC problem was first addressed in [5,6], based on higher order Voronoi diagrams and an iterative process to determine min-cuts that resulted in the L_∞ Hausdorff Voronoi diagram of all geometric min-cuts. In [7] the rectilinear version of the problem was considered and an output sensitive approach was proposed that first computed all possible geometric min-cuts and then directly computed the L_∞ Hausdorff Voronoi diagram, where each geometric min-cut induced an axis-aligned rectangle representing its minimum inducing region. The MGMC problem is also closely related to the farthest line-segment Voronoi diagram which has constant complexity for non-crossing line segments [2].

2 Geometric Cuts

Let $G = (V, E)$ be the undirected graph in an MGMC problem and $H = (V_H, E_H)$ be its planar subgraph embedded in the plane P with $|V| = N_G$, $|E| = M_G$, $|V_H| = n$, and $|E_H| = m$. Due to the planarity of H, $m = O(n)$. Edges in H are straight line segments with rectilinear or diagonal orientation. A pair of vertices u and v in a graph is connected if there is a path in this graph from u to v, and disconnected otherwise. A graph is connected if every pair of its distinct vertices is connected. Without loss of generality (WLOG), we assume that G is connected. A cut C of G is a subset of edges in G whose removal disconnects G. A cut C is minimal if removing any edge from C no longer forms a cut.

Definition 1 (see [7]). *Let R be a connected region in P, and $C = R \cap H$ be the set of edges in H intersected by R. C is called a geometric cut induced by R if the removal of C from G disconnects G. A geometric cut C is called a 1-D geometric cut (or a 1-D cut) if $R(C)$ is a segment. If $R(C)$ is an axis-aligned rectangle, then C is called a 2-D geometric cut (or a 2-D cut). A geometric cut*

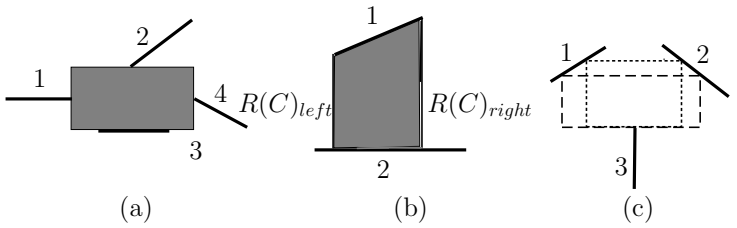

Fig. 1. (a) R(C) of a 2-D cut C is bounded by 4 edges. (b) Each vertical line segment between $R(C)_{left}$ and $R(C)_{right}$ (gray region) forms a 1-D cut $\{1, 2\}$. (c) Two minimum reducing regions (dashed and dotted rectangles) of geometric cut {1,2,3}.

C is a geometric minimal cut if the set of edges intersected by any rectangle shrinking from $R(C)$ is no longer a cut.

When there is no ambiguity of the region R, we often call the cut induced by R as a geometric cut for simplicity. For a given cut C, its *minimum inducing region* $R(C)$ is the minimum axis-aligned rectangle which intersects every edge of C. For some geometric cut C, its $R(C)$ could be degenerated into a horizontal or vertical line segment, or even a single point. If $R(C)$ is not a point, it may not be fixed for a given geometric minimal cut C (see Figure 1).

Let $B(C)$ denote the set of edges bounding a geometric minimal cut C (i.e., the set of edges in C intersecting $R(C)$). Due to the minimality nature of C, removing any edge in $B(C)$ will lead to a non-cut. This means that any edge in $B(C)$ is necessary for forming the cut. However, this is not necessarily true for edges in $C \setminus B(C)$. Thus, a geometric minimal cut may not be a minimal cut. This also explains why the number of geometric minimal cuts is polynomial and the number of minimal cuts is exponential [7].

Clearly edges in $B(C)$ define the boundary position of $R(C)$. However, it is not true that all edges in $B(C)$ are needed to define $R(C)$.

Lemma 1. *For any 1-D (or 2-D) geometric minimal cut, the number of edges in $B(C)$ needed to define $R(C)$ is at most two (or four) (see Figure 1(a)).*

For simplicity, we assume thereafter that $B(C)$ contains only those edges which are barely sufficient to define $R(C)$.

For a 1-D cut C, the location of $R(C)$ may not be fixed, since there may be an infinite number of 1-D cuts cutting the same set of edges (see Figure 1(b)). For a 2-D cut C, it is also possible that $R(C)$ is not fixed due to the appearance of diagonal oriented edge(s) in $B(C)$. For example, if the vertex of $R(C)$ incident to a diagonal oriented edge $e \in B(C)$, moving the vertex along e continuously could generate an infinite number of different $R(C)$. In Figure 1(c), two minimum inducing regions represented by dotted and dashed rectangles are induced by the geometric cut $\{1, 2, 3\}$.

Lemma 2. *For a given geometric minima cut C, if $R(C)$ is not fixed, the number of $R(C)$ is infinite.*

Note that in the presence of non-fixed inducing region, the computation of map $\mathcal{M}$ is quite different. In this case, if point $p \in P$ falls in the cell of a geometric minimal cut C, p is closer to one of its $R(C)$ than to all $R(C')$ of any other cut C'. Thus the MGMC problem is to construct a Hausdorff Voronoi diagram of which each cell corresponding to a geometric minimal cut C is a union of its $R(C)$. The main challenge is to efficiently deal with those Voronoi cell owned by an infinite number of rectangles corresponding to the same geometric minimal cut.

3 Identifying Geometric Minimal Cuts and Minimum Inducing Regions

To compute the map $\mathcal{M}$ of geometric minimal cuts, we first identify all possible geometric minimal cuts and then construct the Hausdorff Voronoi diagram of their infinite number of minimum inducing regions.

3.1 Computing Geometric Minimal Cuts

To identify all 1-D and 2-D geometric minimal cuts, we adopt the algorithm proposed in [7]. In [7], it has shown that all geometric minimal cuts induced by a planar rectilinear embedding of H can be identified in a worst case $O(n^3 \log n(\log\log n)^3)$ time and in $O(n \log n(\log\log n)^3)$ time if the maximum size of the cut is bounded by a constant.

3.2 Computing Minimum Inducing Regions

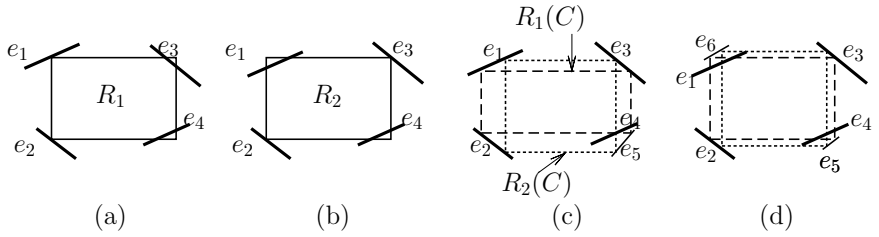

Fig. 2. (a) R_1 with 2 neighboring crossings $\{e_3, e_4\}$ is not an R(C). (b) R_2 with 2 opposite crossings $\{e_1, e_4\}$ is an $R(C)$. (c) Compute e_5 by $\{e_1, e_2, e_3\}$. (d) Compute $\{e_5, e_6\}$ by $\{e_1, e_3\}$.

First, we emphasize that the methods of computing minimum inducing regions described in this section actually can be applied to arbitrary planar embedding of H. Given a set $\mathcal{C}$ of geometric minimal cuts of H, we need to first identify their minimum inducing regions before computing the Hausdorff Voronoi diagram. For a given cut $C \in \mathcal{C}$, from our previous discussion we know that $R(C)$ may

not be unique. If $R(C)$ is fixed, $R(C)$ is bounded by the edges in $B(C)$ and can be computed in $O(1)$ time. If $R(C)$ is not fixed, we know (from Lemma 2) that there are infinite number of $R(C)$s. Thus it is impossible to compute the Hausdorff Voronoi diagram for all such $R(C)$s. To overcome this difficulty, our main idea is to find a discrete representation to capture the behaviors of all possible $R(C)$s. In other words, we need to find a small set of extreme $R(C)$s to represent the infinite number of $R(C)$s. To achieve this goal, our idea is to analyze the geometric properties of all $R(C)$s. For instance, if $R(C)$ is not fixed for a given 1-D cut C, it is easy to see that each $R(C)$ is bounded by the two extreme 1-D cuts $R(C)_{left}$ and $R(C)_{right}$ (or $R(C)_{top}$ and $R(C)_{bottom}$), and the two bounding edges in $B(C)$ (see Figure 1(b)). For a 2-D cut C, it is more complicated since (1) $B(C)$ contains up to 4 edges and (2) one or more edges could be arbitrarily orientated. From now on, we assume that $B(C)$ consists of 4 non-rectilinear edges. We focus on this case since all the other cases are simpler and can be handled similarly. Thus we omit the details for other cases in this extended abstract.

Definition 2. *$B(C)$ is general if it contains 4 non-rectilinear edges.*

Definition 3. *Given an edge e of a general $B(C)$, e is crossing (or tangent to) a rectangle R if e intersects R twice (or once). Each intersection is called a crossing (or tangency) between $B(C)$ and R. (see Figure 2(b))*

To better understand these concepts, consider the geometric minimal cut $C = \{e_1, e_2, e_3, e_4\}$ shown in Figure 2. R_1 is not a minimum inducing region of C since shrinking it a little bit still cuts C. e_2 (e_3) is tangent to R_2. R_2 is an $R(C)$ with 2 crossings $\{e_1, e_4\}$.

Lemma 3. *Given a geometric minimal cut C, if $R(C)$ is not unique, the number of crossing between $B(C)$ and any $R(C)$ is at most 2. If it is 2, the two crossings are not neighboring to each other.*

Proof. Clearly it is sufficient to prove that there is no pair of neighboring crossings. Suppose that this is not true. We can shrink $R(C)$ by moving the boundary edge of $R(C)$ connecting the two neighboring crossings toward its opposite edge by a small distance and $R(C)$ still cuts all edges in C. This is a contradiction. □

Thus, to find all $R(C)$s, we have two cases to consider, (1) the number of crossing is 1 and (2) the number of crossings is 2. For case (1), we explain our idea by an example. In Figure 2(c), a general $B(C)$ contains 4 edges $\{e_1, e_2, e_3, e_4\}$ with slopes $\{\kappa_1, \kappa_2, \kappa_3, \kappa_4\}$ respectively. A rectangle (shown by dashed line) tangent to $\{e_1, e_2, e_3\}$ respectively and crossed by e_4 is a minimum inducing region $R_1(C)$. Another rectangle (shown by dotted line) tangent to the same set of edges as $R_1(C)$ forms another minimum inducing region $R_2(C)$. $R_2(C)$ is also crossed by e_4. For all $R(C)$s tangent to e_1, e_2, e_3, their corner points which are not on any edge of $B(C)$ induce a new edge e_5 (i.e., union of all such corner points forms an edge), called *skating edge*. The skating edge e_5 can be easily computed from $\{e_1, e_2, e_3\}$. Given a set of 4 edges $\{e_1, e_2, e_3, e_5\}$, if we move a point $\mathtt{p}$ along

the valid interval of e_1, each p corresponds to exactly one minimum inducing region $R(C)$. Note that the valid interval in which $R(C)$ exists can be easily computed. Thus we call $\{e'_1, e'_2, e'_3, e_5\}$ a configuration of $R(C)$, where e'_i is the valid interval of e_i for $i \in \{1, 2, 3\}$. For case (2), as shown in Figure 2(d), the dotted and dashed rectangles are tangent to e_2 and e_3 respectively. Similar to case (1), we can also compute two skating edges e_5 and e_6 for all $R(C)$s tangent to e_2 and e_3. For any three edges of $B(C)$, the corresponding configuration can be computed in either case (1) or (2). Thus we have the following lemma.

Lemma 4. *Given a geometric minimal cut C, $R(C)$ has at most 4 configurations and each configuration is a set of 4 edges.*

For a general $B(C)$, we only need to find a set of 4 configurations to represent all its $R(C)$s.

Lemma 5. *Given a geometric minimal cut C, the representation of $R(C)$ (i.e., all configurations if $R(C)$ is not unique) can be computed in $O(|B(C)|)$ time.*

4 Generating Map of Geometric Minimal Cuts

Given a set $\mathcal{C}$ of geometric minimal cuts of H, the Hausdorff Voronoi diagram of $\mathcal{C}$ is a partition of the embedding plane P of H into regions (or cells) so that the Hausdorff Voronoi cell of a cut $C \in \mathcal{C}$ is the union of all points whose Hausdorff distance to some $R(C)$ is closer than to any minimum inducing region of other cuts in $\mathcal{C}$.

In our MGMC problem, we have four types of objects, the fixed and non-fixed minimum inducing regions of 1-D geometric minimal cuts and the fixed and non-fixed minimum inducing regions of 2-D geometric minimal cuts. As it is well known, the Hausdorff Voronoi diagram can be viewed as the intersections of wavefronts propagating from each object with unit speed. Thus our construction of the Hausdorff Voronoi diagram uses the wave propagation concept. We focus on the discussion of 2-D $R(C)$s since the same idea can be applied to the 1-D case. More specifically, for non-fixed $R(C)$, we assume that $B(C)$ is general. We have two types of objects to consider, the fixed rectangle $R(C)$ and the union of non-fixed rectangle $UR(C)$. To visualize the whole growing process, we can lift the waves to 3D with time being the third dimension and thus each object corresponds to a 3D cone. We will discuss the properties of 3D cones of $R(C)$ and $UR(C)$ in the next section.

Lemma 6 (see [7]). *The Hausdorff Voronoi diagram can be obtained by projecting the lower envelope of the 3D facet cones to the xy plane.*

4.1 Properties and Plane Sweep Approach

Lemma 7. *Let C be a 2-D geometric minimal cut with a fixed $R(C)$. At any moment, the wavefront of $R(C)$ is either empty or an axis-aligned rectangle. Furthermore, the wavefront in 3D is a facet cone apexed at a segment and with each facet forming a 45 degree angle with the xy plane. (see Figure 3)*

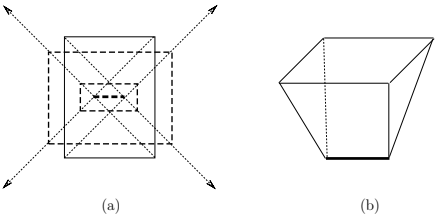

Fig. 3. (a) The wavefront (dashed line) of a fixed $R(C)$ (solid line). (b) Its corresponding 3D V-cone. Note that the bold dashed line in (a) is corresponding to the bold solid line in (b).

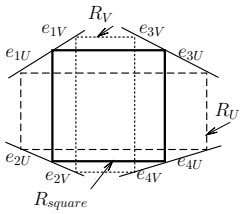

Fig. 4. R_{square} (bold line), R_U (dashed line) and R_V (dotted line)

Definition 4. *Given a fixed $R(C)$, a 3D facet cone $\partial W(C)$ is a U-cone (or V-cone) if its apex segment s_C is parallel to the y (or x) axis. (see Figure 3(b))*

Next we discuss the properties of the wavefront of $UR(C)$ of a general $B(C)$. By Lemma 4, we know that $UR(C)$ is represented by at most 4 configurations $UR_1(C)$, $UR_2(C)$, $UR_3(C)$ and $UR_4(C)$, with each corresponding to a 3D wavefront. The 3D wavefront of $UR(C)$ is simply the lower envelope of the wavefronts of the four configurations (looking from $-\infty$ of the z axis). Since the property of each wavefront is the same, we only need to focus on one configuration $UR_1(C) = \{e_1, e_2, e_3, e_4\}$.

$UR_1(C)$ is the union of an infinite number of $R(C)$s. It is possible that some of them have U-cones as their 3D wavefronts and the others have V-cones as their 3D wavefronts. To distinguish these $R(C)$s, we further classify $UR_1(C)$ into two sub-configurations $UR_{1U}(C) = \{e_{1U}, e_{2U}, e_{3U}, e_{4U}\}$ and $UR_{1V}(C) = \{e_{1V}, e_{2V}, e_{3V}, e_{4V}\}$ such that any rectangle from $UR_{1U}(C)$ (or $UR_{1V}(C)$) generates only U-cone (or V-cone). The computation of $UR_{1U}(C)$ and $UR_{1V}(C)$ can be done in $O(1)$ time since we only need to check the position of the square (denoted by R_{square} in Figure 4) of $UR_1(C)$ if it exists. All the rectangles of $UR_1(C)$ with length bigger (or smaller) than the width form U-cones (or V-cones). In Figure 4, R_U (or R_V) is a minimum inducing region corresponding to an U- (or V-) cone in 3D. Now we analyze the property of the wavefront of $UR_{1U}(C)$ and $UR_{1V}(C)$ respectively.

To better illustrate the whole growing process of the wavefront of $UR_{1U}(C) = \{e_{1U}, e_{2U}, e_{3U}, e_{4U}\}$, we first choose 3 rectangles $\{R_{min}, R_{mid}, R_{max}\}$ such that R_{min} (or R_{max}) is an extreme rectangle of $UR_{1U}(C)$ in which the difference between the length and width is minimum (or maximum). R_{mid} is any rectangle in between. We first analyze the wavefront of these three rectangles and then generalize the idea to all rectangles in $UR_{1U}(C)$.

Since the Hausdorff distance to a rectangle R_{min} is determined by the four corner points, an equivalent view is to propagate 4 separated waves from the 4

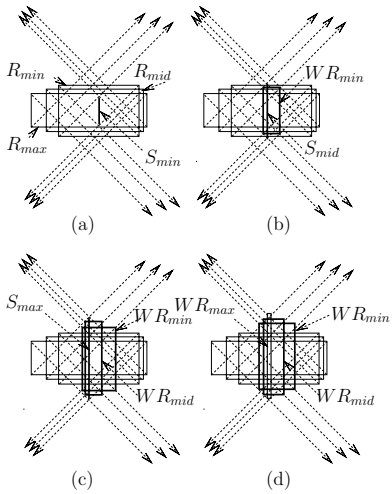

Fig. 5. The growing process of the wavefront for 3 rectangles

corner points of R_{min} with each being an L_∞ ball. Let $B1_{min}$, $B2_{min}$, $B3_{min}$ and $B4_{min}$ be the 4 L_∞ balls of R_{min}. The common intersection of the 4 balls are the wave WR_{min} of R_{min}. We grow the 4 corner balls of R_{mid} and R_{max} in the same way and denote their waves by WR_{mid} and WR_{max} respectively. Initially, both of them are empty. We call it stage 1. Once the size of the 4 balls of R_{min} reaches the minimum Hausdorff distance to R_{min}, their common intersection forms a segment s_{min} located at the center of R_{min} and parallel to the shorter side of R_{min} (see Figure 5(a)). As Bi_{min} grows, WR_{min} becomes a rectangle. Later, s_{mid} appears when the size of the 4 balls of R_{mid} reaches the minimum Hausdorff distance to R_{mid} and intersects WR_{min} (see Figure 5(b)). After that, WR_{mid} grows in the same way as WR_{min} does. Finally, s_{max} appears and intersects the above two rectangles (see Figure 5(c)). We call the above procedure stage 2. In stage 3, all 3 rectangles grows simultaneously (see Figure 5(d)).

To analyze the wavefront $WUR_{1U}(C)$ of $UR_{1U}(C)$, we generalize the above discrete process by replacing R_{mid} with the union of all rectangles between R_{mid} and R_{max}. It is easy to see the shape of $WUR_{1U}(C)$ at each stage. In stage 1, $WUR_{1U}(C)$ is empty. In stage 2, $WUR_{1U}(C)$ is a segment s_{min}. $WUR_{1U}(C)$ keeps the shape as a hexagon (see Figure 6(a)) until s_{max} appears. Each non-rectilinear edge of the hexagon is parallel to $\{e_{1U}, e_{2U}, e_{3U}, e_{4U}\}$ respectively. In stage 3, $WUR_{1U}(C)$ is an octagon, since each endpoint of s_{max} grows to an segment paralleled to the x-axis. We denote the two endpoints of s_{min} (or s_{max}) as pa_{min} and pb_{min} (or pa_{max} and pb_{max}). e_a (or e_b) is the straight line segment between pa_{min} and pa_{max} (or pb_{min} and pb_{max}) (see Figure 6(b)). The property of wavefront of $UR_{1V}(C)$ is the same except that everything in each stage is rotated counterclockwise by 90-degree.

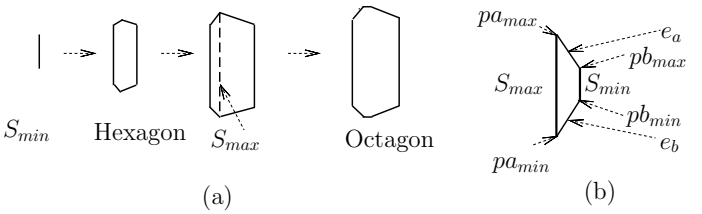

Fig. 6. (a) The growing process of wavefront $WUR_{1U}(C)$. (b)e_a and e_b.

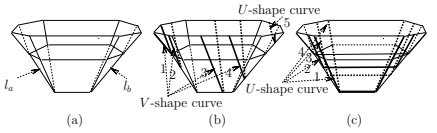

Fig. 7. (a) CV-cone. (b) 5 stages of sweeping CV-cone. (c) 4 stages of sweeping CU-cone (view the process by rotated 90-degree).

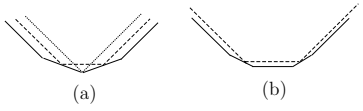

Fig. 8. (a) stage 4 (b) stage 5

To better understand the whole growing process, we lift the wavefront $WUR_{1U}(C)$ to 3D, with time being the third dimension (see Figure 7(a)). The following lemma summarizes the main properties of the growing process.

Lemma 8. *Let C be a 2-D geometric minimal cut with non-fixed $R(C)$. At stage 1, the wavefront $WUR_{1U}(C)$ (or $WUR_{1V}(C)$) is empty. At stage 2 (or 3), $WUR_{1U}(C)$ (or $WUR_{1V}(C)$) is a hexagon (or an octagon) with the property discussed above. Furthermore, the wavefront in 3D is a facet cone apexed at a segment and with each facet forming a 45 degree angle with the xy plane. It is a 6-sided (or 8-sided) facet cone in stage 2 (or 3). Let l_a and l_b be the top and bottom (or left and right) edge of the 6-sided facet cone of $WUR_{1U}(C)$ ($WUR_{1V}(C)$) at stage 2. Then e_a (or e_b) is the projection of l_a (or l_b) on the xy plane.*

Definition 5. *Given a non-fixed $R(C)$ and one of its sub-configuration I, a 3D facet cone $\partial WI(C)$ is a CU-cone (or CV-cone) if its apex segment s_{min} is parallel to the y (or x) axis.*

By the above lemma, CU-cone (or CV-cone) has 6 facets at stage 2 and 8 facets at stage 3. Thus we totally have 4 types of objects in 3D, U-cone, V-cone, CU-cone and CV-cone.

Lemma 9. *The wavefront of $UR(C)$ can be represented by at most 4 CU-cones and 4 CV-cones.*

To efficiently construct the Hausdorff Voronoi diagram $HVD(\mathcal{C})$, we follow the spirit of Fortune's plane sweep algorithm for points [3], and sweep along the

x axis direction a tilted plane Q in 3D which is parallel to the y axis and forms a 45 degree with the xy plane. Q intersects the xy plane at a sweep line L parallel to the y axis.

Since every facet of a 3D facet cone forms a 45 or 135-degree angle with the xy plane and apexed at either a horizontal or vertical segment, at each moment, the intersection of Q and a cone $\partial WI(C)$ is either a V-shape curve (i.e., consisting of a 45-degree ray and a 135-degree ray on Q) or a U-shape curve (i.e., consisting of a 45-degree ray, a segment parallel to L, and a 135-degree ray). When the cone is first encountered, it introduces either a V-shape curve or a U-shape curve to Q. When L (or Q) moves, the curve grows and its shape may change from a V-shape to a U-shape. In addition, the height of the apex of a V-shape curve could change due to the existence of CV-cones. For both CU-cone and CV-cone, the 45-degree and 135-degree ray of a U-shape curve could move along the y direction on Q. Next, we discuss the intersection between each type of cones and Q in details.

First we consider U cones. Let $\partial W(C)$ be any U cone with apex segment s_C, and v_1 and v_2 be the two endpoints of s_C. When the sweep plane Q first encounters $\partial W(C)$, it introduces a U-shape curve C_u to Q. Let r_l, r_r, and s_m be the left and right rays and the middle segment of C_u respectively. Initially s_m is the apex segment s_C, and r_l and r_r are the two edges of facet cone. When Q (or L) moves, C_u grows and always maintains its U-shape.

Lemma 10 (see [7]). *Let $\partial W(C)$, C_u, r_l, r_r and s_m be defined as above. When Q moves in the direction of the x axis, C_u is always a U-shape curve. The supporting lines of r_l and r_r remain the same on Q, and the two endpoints of s_m (the fixed points of r_l and r_r) moves upwards in unit speed along the two supporting lines.*

Let $\partial WI(C)$ be any CU-cone with apex segment s_{min}, and pa_{min} and pb_{min} be the two endpoints of s_{min}. When the sweep plane Q first encounters $\partial WI(C)$, it introduces a U-shape curve C_{cu} to Q. Let r_l, r_r, and s_m be the left and right rays and the middle segment of C_{cu} respectively.

Lemma 11. *Let $\partial WI(C)$, C_{cu}, r_l, r_r and s_m be defined as above. When Q moves in the direction of the x axis, C_{cu} is always a U-shape curve (i.e., at each moment, r_l and r_r remain 45-degree and 135-degree respectively.). The whole process can be divided into 4 stages (see Figure 7(c)). At stage 1, s_m is the apex segment s_{MM} which is the edge of rectangle RI parallel to y-axis with smaller x coordinate. RI is the rectangle growing from s_{min} at the moment when s_{max} appears. At stage 2, s_m moves in unit speed downwards from s_{MM} to s_{min} with its two endpoints staying on l_a and l_b respectively. At stage 3, s_m moves in unit speed upwards from s_{min} to S_{NN} with its two endpoints staying on l_a and l_b respectively. S_{NN} is the edge of rectangle RI parallel to s_{MM}. At stage 4, two endpoints of s_m moves upwards in unit speed along r_l and r_r.*

For an arbitrary V cone $\partial W(C')$, let $s_{C'}$ be its apex segment, and v'_1 and v'_2 be its two endpoints (or left and right endpoints). When Q first touches $\partial W(C')$

at v'_1, it generates a V-shape curve C'_v. C'_v remains a V-shape curve before encountering v'_2. After that, C'_v becomes a U-shape curve.

Lemma 12 (see [7]). *Let r_l and r_r be the two rays of C'_v, and s_m be the middle segment of the U-shape curve C'_v after Q visiting v'_2. During the whole sweeping process, the supporting lines of r_l and r_r are fixed lines on Q. C'_v remains the same V-shape curve on Q before encountering v'_2. s_m moves upwards in unit speed along the supporting lines of r_l and r_r after Q encounters v'_2.*

For an arbitrary CV-cone $\partial WI(C')$, let s'_{min} be its apex segment, and pa'_{min} and pb'_{min} be its two endpoints (or left and right endpoints).

Lemma 13. *The whole sweeping process of a CV-cone can be divided into 5 stages (see Figure 7(b)). At stage 1, when Q first touches $\partial WI(C')$ at pa'_{max}, it generates a V-shape curve C_{cv}. At stage 2, C_{cv} is still a V-shape curve moving in unit speed downwards from pa'_{max} to pa'_{min} with its apex staying on l'_a. At stage 3, C_{cv} remains the same V-shape curve on Q before encountering pb'_{min}. At stage 4, C_{cv} is a 4-edge V-shape curve (see solid lines in Figure 8 (a)) moving in unit speed upwards from pb'_{min} to pb'_{max} with its apex staying on l'_b. At stage 5, after Q encounters pb'_{max}, C_{cv} becomes an 5-edge U-shape curve (see solid lines in Figure 8 (b)) with the middle segment s'_m moves upwards in unit speed along the fixed supporting lines of r'_l and r'_r on Q.*

The 4-edge V-shape curve (5-edge U-shape curve) is essentially a union of U-shape curves if a V-shape is considered as a degenerated case of U-shape curve (See dotted lines in Figure 8 (a)). Two of those U-shape curves are shown at dotted lines in Figure 8. For each 4-edge V-shape curve (5-edge U-shape curve), we divide it to two curves, one curve with 45 and 135 degrees lines and another curve with the rest. Instead of working on the 4-edge V-shape curve and 5-edge U-shape curve directly, we convert each of them to the normal U or V-shape curve by computing the intersections with other curves on the beach line. Since any two curves may intersect at most twice, the same complexity will be kept of the beach line as [7]. Thus, even though there are 4 types of cones, at each moment, we only have U and V-shape curves on Q. We have two cases in which a hidden U or V-shape curve could appear in the beach line instead of only one case as in [7].

Lemma 14. *Let $\partial W(C_1)$ be either a U or V cone and $\partial W(C_2)$ be a V cone with its left endpoint v_1 of s_{C_2} being inside of $\partial W(C_1)$ and its right endpoint v_2 being outside of $\partial W(C_1)$. If $\partial W(C_2)$ is not entirely contained by the union $\cup_{C_i \in \mathcal{C}; C_i \neq C_2} \partial W(C_i)$, the V-shape curve C_2 introduced by $\partial W(C_2)$ will be hidden by the beach line at the beginning and then becomes part of the beach line later. This is the first case in which a hidden U or V-shape curve could appear in the beach line.*

Lemma 15. *Let $\partial W(C'_1)$ be any type of cone and $\partial W(C'_2)$ be a CV-cone with pa'_{max} or pa'_{min} being inside of $\partial W(C'_1)$ and its right endpoint pb'_{min} being outside of $\partial W(C'_1)$. If $\partial W(C'_2)$ is not entirely contained by the union*

$\cup_{C_i \in \mathcal{C}; C_i \neq C_2'} \partial W(C_i)$, *the* V*-shape curve* C_2' *introduced by* $\partial W(C_2')$ *will be hidden by the beach line at the beginning and then becomes part of the beach line later. This is the second case in which a hidden* U *or* V*-shape curve could appear in the beach line.*

Lemma 16. *Let* $\mathcal{C}$ *be a set of* N *minimal geometrical cuts. The edges of* $HVD(\mathcal{C})$ *are either segments or rays, and the vertices of the* $HVD(\mathcal{C})$ *are either the vertices of bisectors or the intersections of bisectors.*

Lemma 17. *The size* K *of the* L_∞ *Hausdorff Voronoi diagram of* N *minimum geometrical cuts is* $O(N+M')$*, where* M' *is the number of intersecting minimum inducing region pairs. The bound is tight in the worst case.*

4.2 Events, Data Structures and Algorithm

To implement the plane sweep algorithm, we use similar data structures as in [7] with one modification for handling V events. To efficiently detect all possible V events, our idea is to process the apex points of all V-shape curves, including (1) the left endpoint of a V-cone's apex segment and (2) pa_{max} and pa_{min} of a CV-cone into the 3D dynamic range search tree data structure MD. Thus we are able to handle all events efficiently in a similar way as [7].

Theorem 1. *The* L_∞ *Hausdorff Voronoi diagram* $HVD(\mathcal{C})$ *of a set* $\mathcal{C}$ *of geometric minimal cuts can be constructed by a plane sweep algorithm in* $O((N + K)\log^2 N \log\log N)$ *time, where* $N = |\mathcal{C}|$ *and* K *is the complexity of the Hausdorff Voronoi diagram.*

References

1. Abellanas, M., Hernandez, G., Klein, R., Neumann-Lara, V., Urrutia, J.: A Combinatorial Property of Convex Sets. Discrete & Computational Geometry 17, 307–318 (1997)
2. Dey, S.K., Papadopoulou, E.: The $L_\infty(L_1)$ Farthest Line-Segment Voronoi diagram. In: The Ninth International Symposium on Voronoi Diagrams in Science and Engineering, pp. 49–55 (2012)
3. Fortune, S.: A sweepline algorithm for Voronoi diagrams. Algorithmica 2, 153–174 (1987)
4. Papadopoulou, E.: Critical area computation for missing material defects in VLSI circuits. IEEE Transactions on Computer-Aided Design 20(5), 583–597 (2001)
5. Papadopoulou, E.: Higher order Voronoi diagrams of segments for VLSI critical area extraction. In: The Eighteenth International Symposium on Algorithms and Computation, pp. 716–727 (2007)
6. Papadopoulou, E.: Net-aware critical area extraction for opens in VLSI circuits via high-order Voronoi diagram. IEEE Transactions on Computer-Aided Design 20(5), 583–597 (2011)
7. Xu, J., Xu, L., Papadopoulou, E.: Computing the Map of Geometric Minimal Cuts. In: The Twentieth International Symposium on Algorithms and Computation, pp. 244–254 (2009)
8. Xu, J., Xu, L., Papadopoulou, E.: Map of Geometric Minimal Cuts with Applications. In: Handbook of Combinatorial Optimization, 2nd edn. Springer (2013)