Critical Area Computation – A New Approach

Evanthia Papadopoulou
IBM TJ Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598
E-mail: evanthia@watson.ibm.com
Fax: (914) 945-4469

and

D. T. Lee
Department of Electrical and Computer Engineering
Northwestern University
Evanston, IL 20208
E-mail: dtlee@ece.nwu.edu
Fax: (847) 467-4144

Abstract

In this paper we present a new approach for computing the critical area for shorts in a circuit layout. The critical area calculation is the main computational problem in VLSI yield prediction. The method is based on the concept of Voronoi diagrams and computes the critical area for shorts (for all possible defect radii, assuming square defects) accurately in $O(n \log n)$ time, where n is the size of the input. The method is presented for rectilinear layouts and layouts containing edges of slope ± 1. As a byproduct we briefly sketch how to speed up the grid method of Wagner and Koren [18].

1 Introduction

The critical area of a VLSI layout is a measure that reflects the sensitivity of the layout to defects occurring during the manufacturing process and it is widely used to predict the yield of a VLSI chip. Yield prediction is essential in today's VLSI manufacturing due to the growing need to control cost. Models for yield estimation are based on the concept of critical area which represents the main computational problem in the analysis of yield loss due to spot defects during fabrication. Spot defects are caused by particles such as dust and other contaminants in materials and equipment and are classified into two types: "extra material" defects causing shorts between different conducting regions and "missing material" defects causing open circuits. Extra material defects are the ones that appear most frequently in a typical manufacturing process and are the main reason for yield loss. For more information on yield estimation and spot defects see for example [4, 5, 8, 11, 12, 14, 16, 17, 18].

Quoting from [18], the yield of a chip, denoted by Y, is computed as $Y = \Pi_{i=1}^{m} Y_i$ where Y_i is the yield associated with the ith step of the manufacturing process. The yield of a single processing step is modeled as

$$Y_i = (1 + \frac{dA_c}{\alpha})^{-\alpha}$$

where d denotes the average number of defects per unit of area, α the clustering parameter, and A_c the critical area.

For a circuit layout C, the critical area is defined as

$$A_c = \int_0^\infty A(r)D(r)dr$$

where $A(r)$ denotes the area in which the center of a defect of radius r must fall in order to cause circuit failure and $D(r)$ is the density function of the defect size. The defect density function has been estimated as follows[5, 8, 17, 18]:

$$D(r) = \begin{cases} cr^q/r_0^{q+1}, & 0 \le r_0 \\ cr_0^{p-1}/r^p, & r_0 \le r \le \infty \end{cases} \tag{1}$$

where p, q are real numbers (typically $p = 3, q = 1$), $c = (q+1)(p-1)/(q+p)$, and r_0 is some minimum optically resolvable size.

Existing methods for yield prediction and critical area computation can be classified into the following types:

1. Geometric methods: Compute $A(r)$ for several different values of r independently. Use the results to approximate A_c. The methods to compute $A(r)$ are usually based on *shape-expansion* followed by *shape-intersection* (see e.g. [6]) and have a quadratic flavor for medium or large defect sizes. For rectilinear layouts there is a more efficient scan-line method which can compute $A(r)$ in multiple layers [14].
2. Virtual artwork approach: Build a virtual artwork having the same statistical features as the nominal IC layout. The virtual artwork is arranged in a form allowing easy calculation of the critical area as a function of the defect radius [10]. Accuracy is limited by differences in the detail of the nominal and virtual artworks.
3. Monte Carlo approach: Draw a large number of defects with their radii distributed according to $D(r)$, check for each defect if it causes a fault, and divide the number of defects causing faults by the total number of defects [20].
4. Grid approach: Assume an integer grid over the layout. Compute the critical radius (the radius of the smallest defect causing a fault at this point) for every grid point [18]. The method works in $O(I^{1.5})$ time, where I is the number of grid points. The accuracy depends on the density of the grid.

In this paper we present a new approach for computing the critical area for shorts in a single layer. The method is based on the concept of *Voronoi diagrams* and can accurately compute the total critical area A_c for square defects in $O(n \log n)$ time where n is the size of the input. In this paper we focus on rectilinear layouts but we also consider layouts containing edges of slope ± 1. This is the type of layout appearing most often in practice. Our method is extendible to layouts with shapes in arbitrary orientations (see [13]). To the best of our knowledge this is the first low polynomial algorithm to compute critical area for shorts accurately in irregular layouts.

This paper is organized as follows. In Section 2, we show the connection between the L_∞ metric and square defects. In Section 3, we review Voronoi diagrams of polygons in the L_∞ metric and in Section 4 we show the relevance of the 2nd order Voronoi diagram to critical area computation. In Section 5, we show that the critical area for shorts can be computed analytically once the 2nd order Voronoi diagram is available. In Section 6, we give a simple plane sweep algorithm to compute the Voronoi diagram of rectangles. Finally, in Section 7, we show that the implicit mechanical construction of the 2nd order Voronoi diagram using a grid can speed up the method of [18] to $O(I)$ where I is the number of grid points.

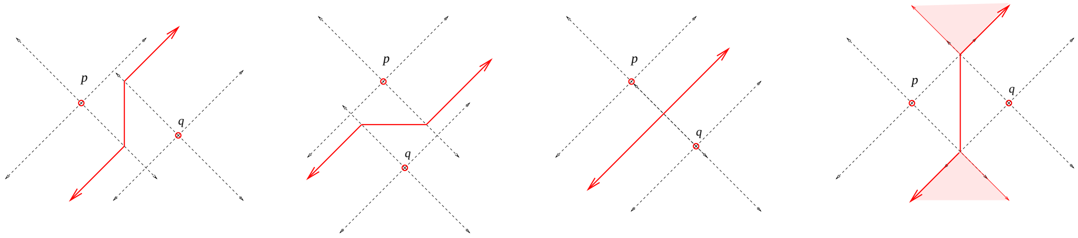

Figure 1: The L_∞ bisector of two points.

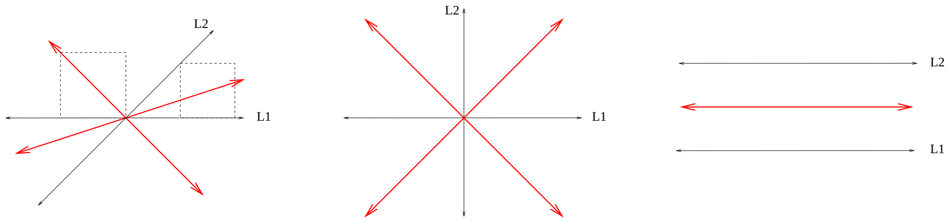

Figure 2: The L_∞ bisector of two lines.

2 Square Defects and the L_∞ Metric

In the critical area literature, defects have consistently been modeled as circles due to the common use of Euclidean geometry. To simplify the computation, in this paper, a spot defect of size r is modeled as a square of side $2r$ (i.e., a square of radius r). In reality spot defects have any kind of shape thus, modeling defects as squares is as good as the circle model. Moreover, in existing methods critical area is approximated in a way that would not make much difference if the defects were assumed squares or circles. (In [14] defects are also modeled as squares.) Modeling defects as squares corresponds to computing critical area in the L_∞ metric instead of the Euclidean. In L_∞, the distance between two points $p = (x_p, y_p)$ and $q = (x_q, y_q)$ is the maximum of the horizontal and the vertical distance between p and q i.e., $d(p,q) = \max\{|x_p - x_q|, |y_p - y_q|\}$. Intuitively, the L_∞ distance is the size of the smallest square touching p and q. In contrast, the Euclidean distance corresponds to the size of the smallest circle touching p and q. Note that the "unit circle" in the L_∞ metric is a square of side two (i.e., a square of radius 1). The L_∞ distance between a point p and a line l is $d(p,l) = \min\{d(p,q), \forall q \in l\}$. In the following, unless otherwise noted, we always imply the L_∞ distance and thus we skip the term for brevity when there is no confusion.

3 The Voronoi Diagram of Polygons

In L_∞, the bisector of two elements (lines or points) is the locus of points at equal L_∞ distance from the two elements. It can be regarded as the locus of points corresponding to centers of squares touching the two elements. Figures 1 and 2, illustrate the L_∞ bisector of two points and two lines respectively. Note that in case of two points along the same horizontal or vertical line the bisector consists of a line segment and two unbounded regions (shaded regions in Figure 1). Without creating any significant difference we assign one region to one point and consider only the outermost boundary of the bisecting region as the bisector (thick lines in Figure 1).

Consider a rectangle R and the four $45°$ rays[1] emanating away from the vertices of R (see Figure 3). They partition the plane into four quadrants. For any point q in the north or south

[1] A $45°$ ray is a ray of slope $+1$ or -1.

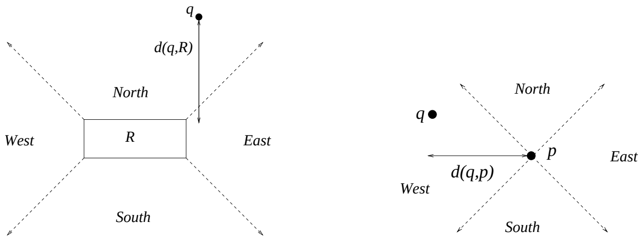

Figure 3: The 45° rays emanating from p and the corners of R partition the plane into four quadrants.

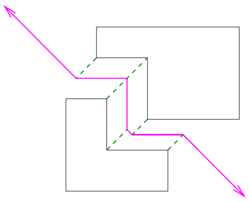

Figure 4: L_∞ bisector of two rectilinear polygons.

(resp. east or west) quadrant the L_∞ distance to R simplifies to the vertical (resp. horizontal) distance between q and the line through the respective horizontal (resp. vertical) edges of R. The four 45° rays through the corners of R are regarded as bisectors between the edges of R. Similarly for a point p; the four 45° rays emanating from p partition the plane into four quadrants where distance simplifies into vertical or horizontal (see Figure 3). The rays can be regarded as bisectors between the vertical and horizontal distance from p.

In Figure 4, the bisector of two rectilinear polygons is illustrated. It consists of portions of bisectors between edges of the two polygons. The border of each portion is induced by the 45° rays emanating from the corners of the polygons (illustrated by dashed lines in Figure 4.) In this paper we focus on the rectilinear case and thus for simplicity, we usually consider rectilinear shapes.

The Voronoi diagram of a set C of polygons is a partition of the plane into regions, called *Voronoi cells*, each of which is associated with a polygon, called the *owner*, of the cell. The Voronoi cell of a polygon P, denoted as $reg(P)$, is the locus of points closer to P than to any other polygon (see for example Figure 5). The Voronoi cell of P is further partitioned into finer regions each of which is associated with an element (edge or vertex) of P. The Voronoi cell of an element $e \in P$ is the locus of points closer to e than to any other element. Clearly, $reg(e) \subset reg(P)$. The boundary that borders two Voronoi cells is called a *Voronoi edge*, and consists of portions of *bisectors* between the owners of the cells. The point where three or more Voronoi edges meet is called a *Voronoi vertex*. For more information on Voronoi diagrams in general see [1, 2]. In the Euclidean metric the boundary of a Voronoi cell contains parabolic curves which make the computation difficult in practice. For this reason we perform computations in the L_∞ metric.

The L_∞ Voronoi diagram of a set of polygons is a skeleton of straight line segments of linear combinatorial complexity [13]. In the special case of rectilinear polygons, the L_∞ Voronoi diagram is a particularly simple skeleton. It consists of line segments in only four orientations, vertical, horizontal, and lines of slope $(+1)$ and (-1) [13]. Figures 9 and 5, illustrate the Voronoi diagram of a set of rectangles and a set of rectilinear polygons respectively. Solid edges illustrate the Voronoi edges separating the cells of different polygons and dashed edges illustrate Voronoi edges induced by the edges of the same polygon. Note that a Voronoi vertex is the meeting point of at least three Voronoi edges (bisectors) and thus it is equidistant from at least three elements. An upper bound

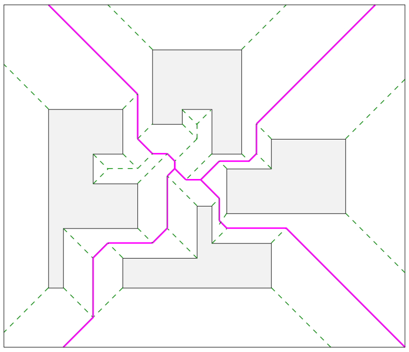

Figure 5: The L_∞ Voronoi diagram of four polygons.

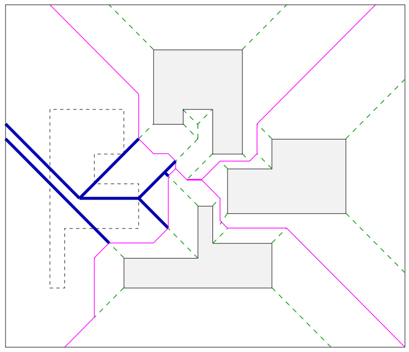

Figure 6: The 2nd order Voronoi diagram in $reg(P)$.

on the size of the L_∞ Voronoi diagram of a set of rectilinear polygons is $(2n-2)$ where n is the number of vertices [13].

The Voronoi diagram encodes nearest neighbor information for every point in the plane. For any point $t \in reg(e)$, where e is an element of polygon P, t is closer to P than to any other polygon. Moreover, t is closer to e than to any other polygonal element (in the L_∞ sense). We say that polygon P and in particular element e is the *nearest neighbor* of t. Several algorithms are available in the literature for computing the Euclidean Voronoi diagram of line segments in time $O(n \log n)$ in expected or worst case (see [2] for a survey). Most of them could be modified to compute the simpler L_∞ diagram, however not all are simple to implement. In Section 6 we give a simple plane sweep algorithm to compute the L_∞ Voronoi diagram of a set of rectangles which can be easily generalized to rectilinear polygons. For a generalization of this algorithm to polygons in arbitrary orientations see [13].

4 Voronoi Diagrams for Critical Area

Let's assume the set of polygons C represents a layer of a VLSI layout. The *critical radius* of a point t is the radius of the smallest defect centered at t which overlaps with at least two different polygons (shapes) in different nets. The critical radius of t reflects the size of the smallest defect that would cause a short at point t. To determine the critical radius of a point t whose nearest polygon is P we need the second nearest polygon to t, say Q (assuming that P and Q belong to different nets). If w is the element of Q nearest to t, the critical radius of t is $d(w,t) = d(Q,t)$.

Consider now the partitioning of the plane obtained as follows: For every polygon $P \in C$

partition the interior of $reg(P)$ by the Voronoi diagram of $C - P$. The subdivision derived in this manner is called the *2nd order Voronoi diagram* of C and provides 2nd nearest neighbor information. In Figure 6, the thick lines illustrate the *2nd order Voronoi diagram* restricted in the interior of $reg(P)$ where P is shown in dotted lines. More formally, the 2nd order Voronoi region of an element $s \in C - P$ within the Voronoi cell of P is defined as $reg_P(s) = \{x \mid d(s, x) \leq d(t, x), \forall t \in C - P\}$. Region $reg_P(s)$ is *owned* by the unique pair (P, s). The critical radius of every point $x \in reg_P(s)$ is the distance between x and s. Note that a square centered at x overlaping with s must also overlap with P (since P is closer to x than s) and thus it causes a short. In other words, s is the element responsible for shorts within $reg_P(s)$. Because s is the element inducing the critical radius for every point $x \in reg_P(s)$, we drop P and say that s is the *owner* of $reg_P(s)$.

This 2nd order Voronoi diagram of polygons is based on the concept of *kth order Voronoi diagram* of line segments [9] which is defined as a planar subdivision such that each region is closest to k line segments. As described in [9], we can visualize the kth order Voronoi diagram as a finer subdivision of the cells in the $(k-1)$ order diagram, where the first order diagram is the ordinary Voronoi diagram. The size of the *2nd order Voronoi diagram* cannot be more than twice the size of ordinary Voronoi diagram i.e., it is linear in the size of the input. To derive the 2nd order diagram within the Voronoi cell of each polygon P we only need to consider the Voronoi neighbors of $reg(P)$. In fact, as shown in Figure 6, the 2nd order Voronoi edges in $reg(P)$ are obtained by extending the 1st order Voronoi edges incident to the boundary of $reg(P)$ in the interior of the cell. More details on how to compute the 2nd order diagram in the rectilinear case are given in Section 6. In the following we assume that the 2nd order Voronoi diagram is available.

A detail worth pointing out in the computation of critical area for shorts is that a pair of distinct polygons may be part of the same net via a connection through a different layer. In this case a defect overlaping with these polygons does not cause a short and thus should not be contributing to critical area. In the above approach it is easy to accommodate this point. Let P, Q be two disjoint polygons that belong to the same net. If the Voronoi cells of P and Q are not neighboring then any defect overlaping P and Q must also overlap with some other polygon. If on the other hand the regions are neighboring we can simply unite $reg(P)$ and $reg(Q)$ into $reg(P \cup Q)$ by removing the adjacent Voronoi edges and treat the resulting region as any other when computing the 2nd order diagram.

5 Computing Critical Area

We have a layer in a circuit layout consisting of a collection of rectilinear polygons C. We assume that overlaping polygons have been unified into single shapes and thus we can assume that all polygons are disjoint. Moreover we can assume that all polygons are *simple* i.e., they do not contain holes. Note that holes are irrelevant to shorts and thus they can be ignored. The boundary of the layout is assumed to be a rectangle B. A defect overlaping a shape and the boundary is not considered to create a short. Our goal is to compute the critical area for shorts i.e., to evaluate the integral $A_c = \int_0^\infty A(r)D(r)dr$, where the defect density function is given by Eq.(1). Using typical values for p, q and c we derive the widely used defect size distribution $D(r) = r_0^2/r^3$. (Since r_0 is typically smaller than the minimum feature size we ignore $D(r)$ for $r < r_0$). Recall that $A(r)$ denotes the area of the *critical region* for square defects of radius r. The critical region for radius r is the locus of points where if the center of a square defect of radius r is placed it causes a short i.e., the defect overlaps with two polygons in different nets.

Let's assume that we are given the 2nd order L_∞ Voronoi diagram of C (with the modification described in the previous section to accommodate polygons of the same net). This diagram essen-

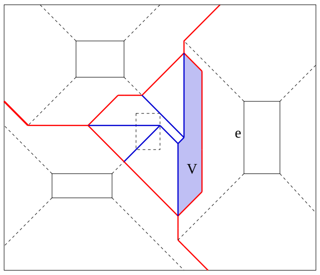

Figure 7: 2nd order Voronoi cell of vertical edge e

tially provides a partitioning of the plane into regions where critical area is easy to compute. The points of intersection of the boundary B with the Voronoi diagram are regarded as Voronoi vertices. The portion of the boundary between two consecutive Voronoi edges can also be regarded as a Voronoi edge and is referred to as a *boundary edge*. Note that boundary edges are only bounding Voronoi cells and do not have regions of their own.

It is not hard to see that the total critical area of the layout is the sum of the critical areas within each (2nd order) Voronoi cell. That is, $A_c = \sum_V A_c(V)$ for all (2nd order) Voronoi cells V where $A_c(V)$ denotes the critical area within V. Note that $A_c(V) = \int_0^\infty A(r,V)D(r)dr$ where $A(r,V)$ denotes the area of the critical region for defect radius r within V. Let's first concentrate on computing critical area within a single (2nd order) Voronoi cell V having as owner an edge e. Let's assume without loss of generality that e is a vertical edge. Since the edges of V are bisectors induced by e they must be vertical or 45°. If V is incident to the boundary then it may be bounded by a horizontal boundary edge which however can be ignored as it will be evident in the sequel. In Figure 7, the shaded region illustrates a 2nd order Voronoi cell of the vertical edge e.

Consider a decomposition of V into trapezoids by drawing the horizontal lines emanating from the Voronoi vertices of V (see Figure 8a). Each trapezoid $\mathcal{T}$ can be further decomposed into triangles and at most one rectangle $\mathcal{R}$ by drawing the vertical lines through its vertices. (In case $\mathcal{T}$ is a slanted parallelogram continue the decomposition recursively, see Figure 8b). Note that the hypotenuse of the triangles (if any) is portion of a 45° Voronoi edge. Note that the (2nd order) Voronoi cell of a horizontal edge e' would be decomposed into trapezoids by drawing vertical lines through its vertices. We distinguish between two kinds of triangles in the above decomposition depending on the relevant position with respect to e of their vertical edge and opposite apex. If the apex lies between e and the vertical edge (comparing abscissae) the triangle is called *diverging*; otherwise it is called *converging*. In Figure 8, the lightly shaded triangle is diverging and the darker shaded one is converging. Given two vertices v_j and v_k such that v_j is closer to e than v_k, let r_j, r_k denote the corresponding critical defect radii i.e., $r_j = x_e - x_j$ and $r_k = x_e - x_k$ where x_j, x_k and x_e denote the x-coordinates of v_j, v_k and e respectively.

Consider a rectangle $\mathcal{R}$, a diverging triangle T_{div}, and a converging triangle T_{cnv} in the above decomposition of $\mathcal{T}$. Using $D(r) = r_0^2/r^3$ and algebraic manipulation we derive the following formulas for the critical area within $\mathcal{R}, T_{div}$ and T_{cvn}.

Lemma 1 *The critical area within a rectangle $\mathcal{R}$, a diverging triangle T_{div} and a converging triangle T_{cnv} is given by the following formulas, using the "r_0^2/r^3" defect density distribution:*

$$A_c(\mathcal{R}) = \frac{r_0^2}{2}\left(\frac{l}{r_j} - \frac{l}{r_k}\right) \tag{2}$$

$$A_c(T_{div}) = \frac{r_0^2}{2}\left(\ln\left(\frac{r_k}{r_j}\right) - \frac{l}{r_k}\right) \tag{3}$$

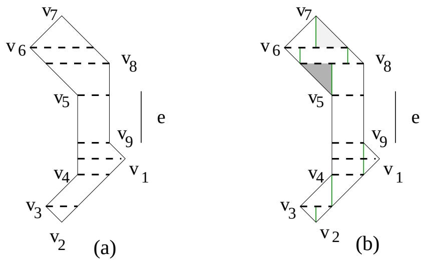

Figure 8: The decomposition of V into trapezoids

$$A_c(T_{cnv}) = \frac{r_0^2}{2}\left(\frac{l}{r_j} - \ln\left(\frac{r_k}{r_j}\right)\right) \tag{4}$$

where l is the size of the vertical side of $\mathcal{R}, T_{div}$ and T_{cnv}, and $r_k, r_j, r_k > r_j$ are the maximum and the minimum critical radius of their vertices.

Proof. Consider the rectangle $\mathcal{R}$ in the above decomposition. The area of the critical region within $\mathcal{R}$ for defect radius r where $r_j \leq r \leq r_k$ is $A(r, \mathcal{R}) = l(r - r_j)$, where l is the length of vertical side of $\mathcal{R}$. For a defect radius $r \geq r_k$ the whole rectangle $\mathcal{R}$ is critical and thus the area is $A(r, \mathcal{R}) = l(r_k - r_j)$. Hence, the critical area within $\mathcal{R}$ is $A_c(\mathcal{R}) = \int_0^\infty A(r, \mathcal{R})D(r)dr = r_0^2 l(\int_{r_j}^{r_k} \frac{r-r_j}{r^3}dr + \int_{r_k}^{\infty} \frac{(r_k-r_j)}{r^3}dr = r_0^2 l(\int_{r_j}^{r_k}(\frac{1}{r^2} - \frac{r_j}{r^3})dr + \int_{r_k}^{\infty} \frac{(r_k-r_j)}{r^3}dr) = r_0^2 l(-\frac{1}{r}\mid_{r_j}^{r_k} + \frac{r_j}{2r^2}\mid_{r_j}^{r_k} - \frac{r_k-r_j}{2r^2}\mid_{r_k}^{\infty} = r_0^2 l(\frac{1}{r_j} - \frac{1}{r_k} + \frac{r_j}{2r_k^2} - \frac{r_j}{2r_j^2} + \frac{r_k-r_j}{2r_k^2}) = r_0^2 l(\frac{1}{2r_j} - \frac{1}{2r_k}) = \frac{r_0^2}{2}(\frac{l}{r_j} - \frac{l}{r_k})$.

Let T_{div} be a diverging triangle within the same decomposition. Let l, w be the length of the vertical and the horizontal edge respectively. Since T_{div} is orthogonal and isosceles, $l = w = r_k - r_j$. The area of critical region within T_{div} for defect radius r where $r_j \leq r \leq r_k$, is $A(r, T_{div}) = \frac{1}{2}l_r w_r$, where $l_r = w_r = r - r_j$. For defect radius $r \geq r_k$ the whole triangle is critical and thus, $A(r, T_{div}) = \frac{1}{2}lw$. Hence, the critical area within T_{div} is $A_c(T_{div}) = \int_0^\infty A(r, T_{div})D(r)dr = \frac{r_0^2}{2}(\int_{r_j}^{r_k}(r-r_j)^2/r^3dr + \int_{r_k}^{\infty}(r_k - r_j)^2/r^3dr) = \frac{r_0^2}{2}(\int_{r_j}^{r_k}(1/r - 2r_j/r^2 + r_j^2/r^3)dr + (r_k - r_j)^2\int_{r_k}^{\infty} 1/r^3dr) = \frac{r_0^2}{2}(\ln(\frac{r_k}{r_j}) + (\frac{2r_j}{r_k} - \frac{2r_j}{r_j}) + (-\frac{r_j^2}{2r_k^2} + \frac{r_j^2}{2r_j^2}) + \frac{(r_k-r_j)^2}{2r_k^2}) = \frac{r_0^2}{2}(\ln(\frac{r_k}{r_j}) + \frac{2r_j}{r_k} - 2 - \frac{r_j^2}{2r_k^2} + \frac{1}{2} + \frac{(r_k-r_j)^2}{2r_k^2}) = \frac{r_0^2}{2}(\ln(\frac{r_k}{r_j}) + \frac{2r_j}{r_k} - 2 + 1 - \frac{r_j}{r_k}) = \frac{r_0^2}{2}(\ln(\frac{r_k}{r_j}) + \frac{r_j}{r_k} - 1)$. But $l = r_k - r_j$ thus, $A_c(T_1) = \frac{r_0^2}{2}(\ln(\frac{r_k}{r_j}) - \frac{l}{r_k})$.

For a converging triangle T_{cnv} consider the diverging triangle T_{div} obtained by flipping T_{cnv} around the hypotenuse. T_{cnv} and T_{div} form a rectangle $\mathcal{R}$. For any defect radius $r \geq r_j$ the area of critical region within T_{cnv} is $A(r, T_{cnv}) = A(r, R) - A(r, T_{div})$. Note that for $r > r_k$ the whole triangle is critical. Thus, $A_c(T_{cnv}) = \int_0^\infty (A(r, R) - A(r, T_{div})D(r)dr = \int_0^\infty A(r, R)D(r)dr - \int_0^\infty A(r, T_{div})D(r)dr$. Thus, $A_c(T_{cnv}) = A_c(\mathcal{R}) - A_c(T_{div})$. The formulas is derived by arithmetic substitution. $\square$

We derive the critical area within V by adding up the critical areas within every rectangle and triangle in the above decomposition of V. Because of the summation, terms of the form l/r corresponding to internal decomposition edges cancel out. Similarly, for logarithmic terms involving endpoints of the decomposition other than Voronoi vertices. Let's classify the edges of V as *red* or *blue* as follows. A vertical Voronoi edge is colored red (resp. blue) if the interior V and the owner e lie at opposite sides (resp. same side) of the Voronoi edge. A 45° edge incident to a diverging triangle is colored red while a 45° edge incident to a converging triangle is colored blue. Similarly for boundary edges. A vertical boundary edge of V (if any) is clearly colored blue. A horizontal boundary edge is equivalent to a horizontal decomposition edge and does not get involved in the formulas of Lemma 1; in other words, it receives no classification and is irrelevant to the critical

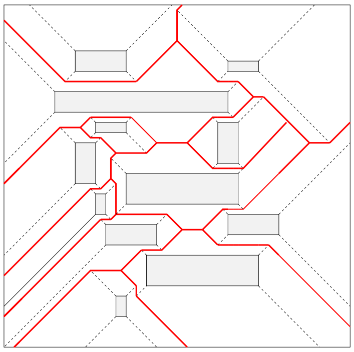

Figure 9: The L_∞ Voronoi diagram of rectangles.

area computation (assuming that the owner e is vertical). In the formulas of Lemma 1, terms corresponding to red edges get added while terms corresponding to blue edges get subtracted. The Voronoi edges bounding the cell of a horizontal edge are classified similarly.

To derive the total critical area for the layout we add the critical areas within every (2nd order) Voronoi cell. Every Voronoi edge bounds exactly two cells and receives the same coloring (red or blue) with respect to both cells. Boundary edges bound exactly one cell and their classification is blue or they are unclassified. We derive the following theorem.

Theorem 1 *Given the 2nd order L_∞ Voronoi diagram of rectilinear polygons of a layer in a circuit layout C and assuming that defects are squares following the "r_0^2/r^3" defect density distribution, the critical area for shorts in that layer is given by the following formula:*

$$A_c = r_0^2(\sum_{red\ e_i} \frac{l_i}{r_i} - \sum_{blue\ e_i} \frac{l_i}{r_i} + \sum_{red\ e_{45}} \ln \frac{r_k}{r_j} - \sum_{blue\ e_{45}} \ln \frac{r_k}{r_j} - \frac{1}{2} \sum_{blue\ b_i} \frac{l_b}{r_b})$$

where l_i and r_i denote the length and the critical radius of an orthogonal Voronoi edge, $r_k, r_j, r_k > r_j$ denote the maximum and the minimum critical radius of a $45°$ Voronoi edge, and l_b, r_b denote the length and the critical radius of a boundary edge. The first two summations are taken over all red and all blue orthogonal Voronoi edges respectively. The third and forth summations are taken over all red and all blue $45°$ Voronoi edges respectively. The last summation is taken over all blue boundary edges.

Thus, the critical area problem for shorts in a rectilinear layout is reduced to the 2nd order Voronoi diagram of polygons. The critical area can be written as a function of Voronoi edges and thus it can be computed analytically once the (2nd order) Voronoi diagram is available. As it will be shown in section 7 this result also holds for layouts containing $45°$ edges. For a generalization to arbitrary shapes see [13].

6 A Sweep-Line Algorithm to Compute the L_∞ Voronoi Diagram of rectilinear polygons

In this section we give a plane-sweep algorithm to compute the L_∞ Voronoi diagram of a rectangular layout consisting of n rectangles (see Figure 9). We consider rectangles instead of rectilinear polygons for simplicity of presentation; the same algorithm can be easily generalized to rectilinear layouts. The time complexity is $O(n \log n)$ and it is based on the wavefront paradigm introduced by Dehne and Klein [3] for the Voronoi diagram of points.

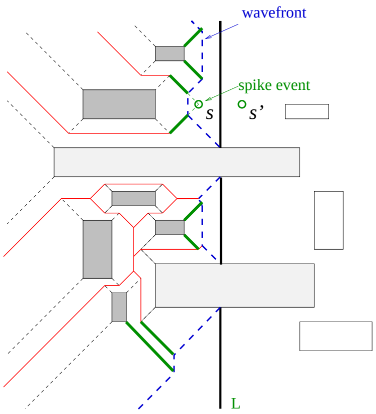

Figure 10: Construction of the Voronoi diagram by plane sweep method.

We shall apply a vertical sweep-line $\mathcal{L}$ sweeping across the entire plane from left to right. The set of rectangles at any instant of the sweeping process will be partitioned into three subsets, S_l, S_m and S_r, corresponding to those that lie totally to the left of $\mathcal{L}$, intersect $\mathcal{L}$, and lie totally to the right of $\mathcal{L}$, respectively. In Figure 10, S_l, S_m and S_r are shown darkly shaded, lightly shaded, and unshaded respectively. The sweep line, which is cut by the rectangles in S_m, is decomposed into $|S_m| + 1$ sweepline segments, two of which are unbounded. Each such sweepline segment is considered to be the left edge of a *moving* rectangle.

We shall maintain a partial Voronoi diagram computed so far for the rectangles in S_l, the portion of the rectangles in S_m that lie to the left of $\mathcal{L}$, and the sweepline segments. At any instant of the sweeping process the boundary of the Voronoi cell associated with a single sweepline segment is called a *wavefront curve* and consists of bisectors induced by the sweepline segment. The collection of the wavefront curves for all sweepline segments is referred to as the *wavefront*. In Figure 10, the wavefront is shown in dashed-lines. The Voronoi edges that have an endpoint common with the wavefront are called *spike bisectors* and are shown in solid and thicker lines in Figure 10. As the sweepline moves to the right, the wavefront will shift to the right and so do the spike bisectors.

Associated with the plane-sweep algorithm there are two major components: an *event list* and a *sweep-line status*. The event list, implemented as a *priority queue* Q, consists of abscissae, called *priority values*, at which the sweep-line status will change. The sweep-line status, implemented as a *height-balanced* binary tree $\mathcal{T}$, implicitly maintains the wavefront. In particular, the sweep-line status contains the spike bisectors and the north and south edges of the rectangles in S_m. The ordering follows the order of the wavefront.

To simulate the wavefront movement to the right as the plane sweep proceeds we parameterize the endpoints of spike bisectors with respect to the abscissa of the sweep line t. In other words we keep the abscissa of the endpoint of a spike bisector as a linear function of t. This function can be easily derived by the property that the endpoints must be equidistant (in L_∞) from the sweep-line and the owners of the spike bisector. The abscissa of the endpoint of a north or south edge in $\mathcal{T}$ is given by t. At any instant t, the wavefront is the polygonal line connecting the endpoints of the elements of $\mathcal{T}$.

In the event list we keep the *events* where the sweepline-status will change. The events are of two kinds: 1) events corresponding to vertical edges of rectangles referred to as *edge events* and 2) events corresponding to the intersection point of two neighboring spike bisectors referred to as *spike events*. The priority value of an edge event is the abscissa of the edge. A spike event corresponds to a potential Voronoi vertex that may or may not appear in the final Voronoi diagram. The priority

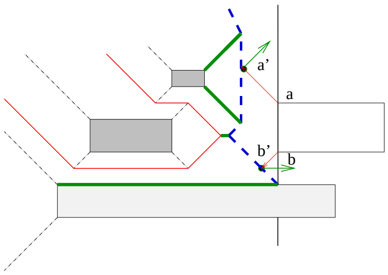

Figure 11: A left edge event.

value of a spike event s of abscissa x_s is $x_s + d(e_j, s)$, where e_j is the owner of s ($s \in reg(e_j)$). For example, in Figure 10, the spike event associated with the potential Voronoi vertex s has as priority value the abscissa of the point marked as s'. In other words, the spike event corresponding to s occurs when the sweepline reaches s'.

Initially the event list Q consists of *all* the abscissae of the west and east edges of the set of rectangles, and the sweep-line status T is empty. We sweep $\mathcal{L}$ across the plane in ascending order of the abscissae, stopping at each event point of a certain priority value. When there are more than one abscissa that corresponds to west or east edge of a rectangle, we adopt the following tie-breaking convention; an east edge always precedes a west edge, and if the edges are of the same type, the one with the smaller ordinate will be processed first.

Let us now discuss three cases to handle the event points. For brevity, let R_i^s, R_i^n, R_i^e and R_i^w, denote the south, north, east and west edges of a rectangle R_i. The bisector between two consecutive edges of R_i is a 45° ray emanating from the corner vertex outwards. The bisector of any two edges e_k, e_j is denoted as $B(e_k, e_j)$.

1. Left edge of rectangle R_i (Figure 11). Let the edge be denoted $\overline{a, b}$ where a and b are the northwest and southwest corners of R_i respectively. We construct bisectors $B(R_i^w, R_i^n)$ and $B(R_i^s, R_i^w)$.

   (a) Identify the intersection points a' and b' of $B(R_i^w, R_i^n)$ and $B(R_i^s, R_i^w)$ with the wavefront respectively. This can be done by binary search in $\mathcal{T}$ for each bisector. The binary search proceeds by discriminating the endpoints of the elements of $\mathcal{T}$ against the bisector. The goal is to identify the pair of consecutive elements in $\mathcal{T}$ whose endpoints lie at opposite sides of the bisector. Then a' and b' can be easily identified as the intersection points of the respective bisector and the segment joining the corresponding pairs of endpoints. In Figure 11, the elements of $\mathcal{T}$ are depicted thickened. Let e_a and e_b denote the edges owning a' and b'.

   (b) Update the Voronoi diagram constructed so far by explicitly constructing the portion of the wavefront between a' and b'. Recall that this corresponds to the line segments joining the endpoints of the elements of $\mathcal{T}$ between a' and b'.

   (c) Update $\mathcal{T}$ by removing those spike bisectors between a' and b'. The spike events associated with these spike bisectors can be either *deleted* from the priority queue now or *ignored* later when they get extracted from the priority queue.

   (d) Two new spike bisectors $B(e_a, R_i^n)$ and $B(R_i^s, e_b)$ are created with startpoints a' and b' respectively. Moreover, at most two new new spike events defined by the intersection of bisector $B(R_i^s, e_b)$ and its lower neighbor and by the intersection of bisector $B(e_a, R_i^n)$

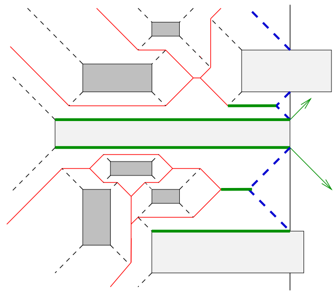

Figure 12: A right edge event.

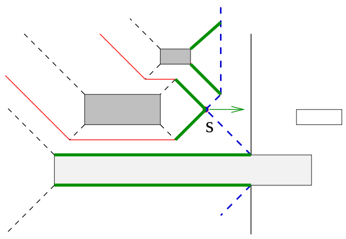

Figure 13: A spike event.

and its upper neighbor will be created and inserted to the priority queue. Note that the priority values of these spike events should be recorded correctly.

(e) Update $\mathcal{T}$ by inserting R_i^n and R_i^s, and the new spike bisectors $B(e_a, R_i^n)$ and $B(R_i^s, e_b)$. In Figure 11, these two new spike bisectors, $B(e_a, R_i^n)$ and $B(R_i^s, e_b)$ are shown with an arrowhead.

2. Right edge of rectangle R_i (Figure 12). In this case the two elements R_i^s and R_i^n must be present in $\mathcal{T}$ in consecutive order. These two elements in $\mathcal{T}$ need to be deleted from $\mathcal{T}$ and replaced by two new bisector elements, $B(R_i^e, R_i^s)$ and $B(R_i^n, R_i^e)$ respectively (shown with arrowheads in Figure 12). As in case 1(d), these two newly created bisectors should be checked against their neighboring elements to see if they would create spike events to be inserted in the priority queue with the appropriate priority values.

3. Spike event s (Figure 13). s is the intersection of two spike bisectors, say $B(R_k^u, R_j^v)$ and $B(R_j^v, R_i^z)$ for some wavefront rectangles R_i, R_j, R_k where u, v, z stand for north, east or south. If either of the two bisectors is not currently in $\mathcal{T}$ (due to case 1(c)) then we discard s, as it is no longer relevant. Otherwise, the two bisectors $B(R_k^u, R_j^v)$ and $B(R_j^v, R_i^z)$ must be adjacent in $\mathcal{T}$. Both bisectors should be deleted from $\mathcal{T}$ and replaced by a new spike bisector $B(R_k^u, R_i^z)$. (The Voronoi cell of their common owner R_j^v becomes part of the final diagram). This new spike bisector should be checked against its neighboring elements for possible intersections, creating spike events to be inserted in the priority queue with the appropriate priority values.

Lemma 2 *The Voronoi diagram in the L_∞-metric for a set of rectangles can be computed in $O(n \log n)$ time.*

Proof. The number of event points, including spike events, is $O(n)$, and the number of elements maintained in the sweep-line status is also $O(n)$. The operation associated with each event point

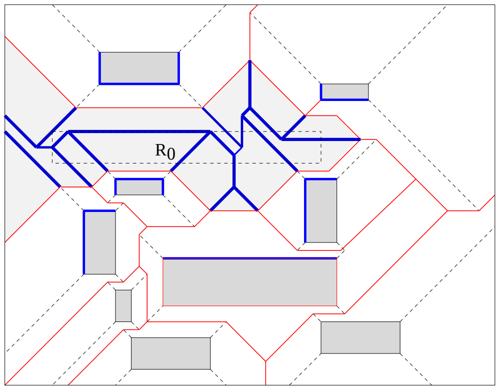

Figure 14: 2nd Order Voronoi diagram in the Voronoi cell of R

takes $O(\log n)$ time, giving rise to an overall time complexity of $O(n \log n)$. The correctness of the algorithm follows from the result of Dehne and Klein[3]. □

Once the Voronoi diagram is available we can compute the 2nd order subdivision within each Voronoi cell in a similar fashion. Figure 14, shows the 2nd order partitioning in the Voronoi cell of the middle rectangle R_0. The Voronoi cell of R_0 is shown shaded. The partitioning line segments are depicted in thicker lines, and they are just extensions of those Voronoi edges that are incident on a Voronoi vertex on the boundary of $reg(R_0)$. The owners of the 2nd order cells, drawn in thicker lines, are rectangle edges that are associated with the Voronoi edges on the boundary of $reg(R_0)$. The computation of the 2nd order subdivision is done independently in every Voronoi cell. Thus we concentrate on the computation of a single cell, say $reg(R_0)$.

We first identify the neighboring cells and the edges that, together with an edge of R_0, define a Voronoi edge on the boundary of $reg(R_0)$. These are the only edges involved in the computation of the 2nd order partition within the Voronoi cell of R_0. The identification can be easily done from the Voronoi edges bounding $reg(R_0)$. We simply compute the Voronoi diagram of these edges and truncate it within the interior of $reg(R_0)$. As before, we'll use a vertical sweep-line sweeping from left to right. The abscissae of the endpoints of site edges will be sorted in ascending order and placed in the event-list. The sweep-line status contains at most *two* edges of a rectangle, one south edge and one north edge (since we are partitioning only one cell), plus a number of spike bisectors. As before, the adjacent elements in the sweep-line status induce *spike events* which correspond to potential Voronoi vertices. The handling of the event points is exactly the same. Once the Voronoi cell of one edge is computed, it gets truncated by the boundary of $reg(R_0)$. A Voronoi edge is always truncated by the incident Voronoi vertex along the boundary of $reg(R_0)$. The time complexity and the correctness of this algorithm follow from the previous discussion.

The data structure to store the Voronoi diagram can be any standard data structure for planar subdivisions. See for example the *doubly connected edge list* [15] or the *quad edge* data structure [7]. However, for the critical area computation there is no need to explicitely maintain the Voronoi diagram. As the plane sweep proceeds, as soon as an individual Voronoi cell is computed we can directly compute the 2nd order subdivision within the cell and derive the portion of critical area due to that cell. Linked lists data structures are thus sufficient.

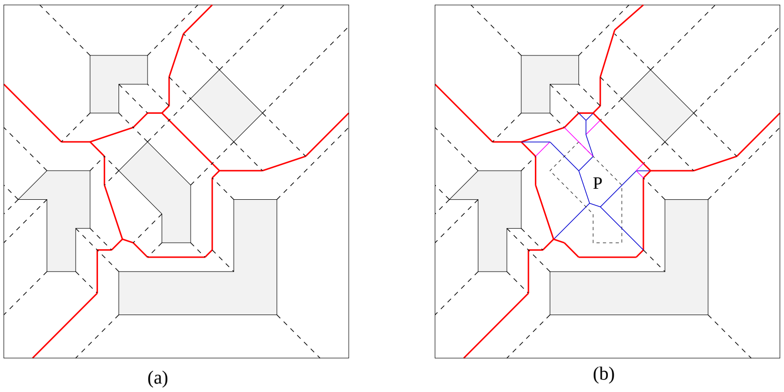

Figure 15: The Voronoi diagram of ortho-45 polygons.

7 Critical Area Computation in Ortho-45 layouts

In this section we extend the result of Section 5 to *ortho-45* layouts i.e., layouts containing shapes with edges of slope ± 1 in addition to orthogonal edges.

As it was shown in [13], the L_∞ Voronoi diagram of ortho-45 polygons contains edges in eight orientations: vertical, horizontal, (± 1)-slope, ($\pm 1/3$)-slope and (± 3)-slope lines. Figure 15a illustrates the L_∞ Voronoi diagram of ortho-45 shapes. Unlike the rectilinear case, now there are vertices that have Voronoi cells of there own. These are vertices incident to two 45° edges and vertices incident to acute angles. Within the Voronoi cell of a vertex v, distance is measured according to the vertical or horizontal distance from v depending on which quadrant of point v that contains the cell. Thus, within $reg(v)$, v can be regarded as a horizontal or vertical edge respectively of length zero. (Recall that the four 45° rays emanating from a point p partition the plane into four quadrants where distance simplifies to vertical or horizontal, see Figure 3). Figure 15b illustrates the 2nd order subdivision within the cell of polygon P.

The decomposition into rectangles and triangles of the Voronoi cell of a vertical or horizontal edge (or vertex) is obtained similarly to Section 5. Moreover Voronoi edges are classified into red and blue in the same manner. The only difference now is that the hypotenuse of some orthogonal triangles may have slope ± 3 or $\pm 1/3$ in addition to ± 1. Thus, the formulas of Lemma 1 for non-isosceles triangles get slightly modified as follows.

Lemma 3 *The critical area within a non-isosceles diverging triangle T_{div} and a non-isosceles converging triangle T_{cnv} in the decomposition of the (2nd order) Voronoi cell of an orthogonal edge is given by the following formulas, using the "r_0^2/r^3" defect density distribution:*

$$A_c(T_{div}) = \frac{r_0^2}{2}(3\ln(\frac{r_k}{r_j}) - \frac{l}{r_k}) \tag{5}$$

$$A_c(T_{cnv}) = \frac{r_0^2}{2}(\frac{l}{r_j} - 3\ln(\frac{r_k}{r_j})) \tag{6}$$

where l is the size of the edge of T_{div} and T_{cvn} parallel to e, $r_k, r_j, r_k > r_j$ are the maximum and the minimum critical radius of their vertices.

Proof. The difference from the derivation of Lemma 1 is that $l = 3w$, where l, w are the length of the vertical and the horizontal edge of T_{cnv} and T_{div} respectively. (Note that the slope of the hypotenuse is ± 3 for a vertical owner and $\pm 1/3$ for a horizontal owner.) For a defect radius

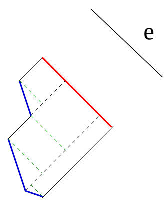

Figure 16: The decomposition of the Voronoi cell of a 45°-edge.

$r, r_j < r \leq r_k$, $l_r = 3w_r$ i.e., $l_r = 3(r - r_j)$. The formulas follow in the same fashion as Lemma 1. $\square$

For a (2nd order) Voronoi cell V of a (-1)-slope edge e, the decomposition of V into trapezoids is obtained by drawing lines of slope $+1$ from the vertices of V. The decomposition of each trapezoid into triangles and rectangles is obtained by drawing (-1)-slope lines from its vertices. Triangles are characterized as diverging or converging similarly to Section 5. Figure 16 shows the decomposition for the cell of a (-1)-slope edge in the Voronoi diagram of Figure 15. Similarly for the Voronoi cell of $(+1)$-slope edge. The formulas of Lemma 1 for Voronoi cells of 45° edges are modified as follows.

Lemma 4 *The critical area within a rectangle $\mathcal{R}$, a diverging triangle T_{div} and a converging triangle T_{cnv} in the decomposition of the (2nd order) Voronoi cell of a 45° edge is given by the following formulas, using the "r_0^2/r^3" defect density distribution:*

$$A_c(\mathcal{R}) = \frac{r_0^2}{\sqrt{2}}(\frac{l}{r_j} - \frac{l}{r_k}) \tag{7}$$

$$A_c(T_{div}) = \frac{r_0^2}{\sqrt{2}}(t\sqrt{2}\ln(\frac{r_k}{r_j}) - \frac{l}{r_k}) \tag{8}$$

$$A_c(T_{cnv}) = \frac{r_0^2}{\sqrt{2}}(\frac{l}{r_j} - t\sqrt{2}\ln(\frac{r_k}{r_j})) \tag{9}$$

where l is the size of the edge of $\mathcal{R}, T_{div}$ and T_{cnv} parallel to e, $r_k, r_j, r_k > r_j$ are the maximum and the minimum critical radius of their vertices, and t is a factor reflecting the slope of the hypotenuse; $t = 2$ if the hypotenuse has slope $\pm 3, \pm 1/3$, and $t = 1$ otherwise.

Proof. Let e be the 45° edge which is the owner of the (2nd order) Voronoi cell V. The area of the critical region within $\mathcal{R}$ for a defect radius r, $r_j \leq r \leq r_k$, is $A(r, \mathcal{R}) = lw_r$, where l is the length of the side parallel to e and w_r is the length of the side perpendicular to e within L_∞ distance r from e. Here $w_r = \sqrt{2}(r - r_j)$, thus $A(r, \mathcal{R}) = \sqrt{2}l(r - r_j)$. The formula follows in the same manner as in Lemma 1.

Let T_{div} be a diverging triangle where l, w is the length of the side parallel and perpendicular to e respectively. Let's assume without loss of generality that e has slope $+1$. Then the hypotenuse of T_{div} can have slope $s = 0, 1/3, 3, \infty$. Figure 17 illustrates diverging triangles with hypotenuse of slope $s = 3$ and $s = 1/3$ respectively. Let r_j and r_k denote the minimum and maximum critical radius of T_{div} respectively. Given a defect radius $r, r_j < r \leq r_k$, the area of critical region within T is the area of triangle OAB (see Figure 17) i.e., $A(r, T_{div}) = \frac{1}{2}l_r w_r$, where $w_r = |OA|$ and $l_r = |AB|$. Clearly $w_r = \sqrt{2}(r - r_j)$. For $s = 0, \infty$, $l_r = w_r$. To determine l_r in the case of $s = 3, 1/3$, let's treat O as the origin of the coordinate system and let's rotate the coordinate system by $\phi = -45°$ so that AB becomes horizontal. In the new coordinate system the slope of the hypotenuse OB is $s' = \frac{s-1}{s+1} = \pm 1/2$ Thus, $l_r = |AB| = \frac{w_r}{|s'|} = 2w_r$. Thus, $A(r, T_{div}) = t(r - r_j)^2$ where where $t = 2$ if $s = 3, 1/3$, and $t = 1$ otherwise.

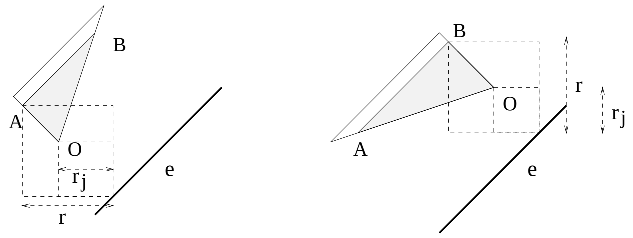

Figure 17: A diverging triangle T_{div} in the cell of a 45° edge.

Following the derivation of Lemma 1 we get $A_c(T_{div}) = r_0^2 t(\ln(\frac{r_k}{r_j}) + \frac{r_j - r_k}{r_k})$. But $l = t\sqrt{2}(r_k - r_j)$. Thus, $A_c(T_{div}) = \frac{r_0^2}{\sqrt{2}}(t\sqrt{2}ln(\frac{r_k}{r_j}) - \frac{l}{r_k})$.

For a converging triangle T_{cnv}, $A_c(T_{cnv}) = A_c(\mathcal{R}) - A_c(T_{div})$ as in Lemma 1. □

The Voronoi edges of cell V are colored *red* and *blue* in a manner similar to the rectilinear case. However, there may be 45° edges that do not contribute to the formulas of Lemma 4 which are not colored. In particular, 45° Voronoi edges parallel to the owner e are colored red if the interior and the owner of the cell lie on opposite sides of the edge; otherwise they are colored blue. 45° Voronoi edges perpendicular to the owner are unclassified with respect to this owner and they do not contribute to critical area within V. Unclassified edges are colored *neutral*. The remaining Voronoi edges of V induce the hypotenuse of a triangle and they are colored red or blue depending on whether the triangle is diverging or converging respectively. Any boundary edges bounding V must be colored blue.

As in the rectilinear case, the total critical area is derived by adding the critical areas within every (2nd order) Voronoi cell. Every Voronoi edge bounds exactly two cells and receives the same coloring (with the exception of neutral) with respect to both cells. Note that an edge may be colored red (resp. blue) with respect to one cell and neutral with respect to the other. We generalize Theorem 1 as follows. The term *prime* is used to denote Voronoi edges that are parallel to their owner i.e., Voronoi edges corresponding to bisectors of parallel edges.

Theorem 2 *Given the 2nd order L_∞ Voronoi diagram of polygons in an ortho-45 layer of a circuit layout C and assuming that defects are squares following the "r_0^2/r^3" defect density distribution, the critical area for shorts in that layer is given by the following formula:*

$$A_c = r_0^2\Big(\sum_{\substack{\text{red,}\\ \text{prime } e_i}} h_i\frac{l_i}{r_i} - \sum_{\substack{\text{blue, prime}\\ e_i}} h_i\frac{l_i}{r_i} + \sum_{\substack{\text{red, non-}\\ \text{prime } e_j}} f_j \ln\frac{r_k}{r_j} - \sum_{\substack{\text{blue, non-}\\ \text{prime } e_j}} f_j \ln\frac{r_k}{r_j} - \frac{1}{2}\sum_{\substack{\text{blue,}\\ \text{prime } b_i}} \frac{l_b}{r_b} - \sum_{\substack{\text{blue, non-}\\ \text{prime } b_j}} \ln\frac{r_k}{r_j}\Big)$$

where l_i and r_i denote the length and the critical radius of a prime Voronoi edge, $r_k, r_j, r_k > r_j$ denote the maximum and the minimum critical radius of a non-prime Voronoi (or boundary) edge, and l_b, r_b denote the length and the critical radius of a boundary edge. Factor $h_i = 1$ for orthogonal prime Voronoi edges and $h_i = \sqrt{2}$ for 45° prime Voronoi edges. Factor $f_i = 1, 2, \frac{1}{2}, \frac{7}{2}$ depending on the slope of the Voronoi edge and its owners. In particular, $f_i = 1$ for a 45° Voronoi edge with orthogonal owners, $f_i = 1/2$ for a 45° Voronoi edge with a 45° owner, $f_i = 2$ for an orthogonal Voronoi edge of 45° owners, and $f_i = \frac{7}{2}$ for a $(\pm 3, \pm\frac{1}{3})$-slope Voronoi edge. The first two summations are taken over all red and all blue prime Voronoi edges respectively. The third and forth summations are taken over all red and all blue non-prime Voronoi edges respectively. The last two summations are taken over all blue boundary edges, the former over prime boundary edges and the latter over non-prime ones.

Proof. As in theorem 1, by summing up critical areas within every (2nd order) Voronoi cell terms corresponding to internal decomposition vertices or edges cancel out. By equation 7 of lemma 4 prime 45° Voronoi edges contribute $\frac{\sqrt{2}}{2}\frac{l_i}{r_i}$ to critical area for each owner while orthogonal prime Voronoi edges contribute $\frac{1}{2}\frac{l_i}{r_i}$. Thus, the values of h_i can be derived.

The values of factor f_i are derived as follows. For a 45° non-prime Voronoi edge with orthogonal owners f_i is clearly 1 (as in the rectilinear case). A 45° non-prime Voronoi edge induced by a 45° owner and an orthogonal edge contributes 0 to critical area for the 45° owner (it is colored neutral) and $\frac{1}{2}\ln\frac{r_k}{r_j}$ for the orthogonal owner. Thus $f_i = \frac{1}{2}$. An orthogonal Voronoi edge of 45° owners contributes $\ln\frac{r_k}{r_j}$ for each owner (equations 8,9 of lemma 4) thus, $f_i = 2$ in this case. A $(\pm 3, \pm\frac{1}{3})$-slope Voronoi edge contributes $\frac{3}{2}\ln\frac{r_k}{r_j}$ because of its orthogonal owner (lemma 3) and $2\ln\frac{r_k}{r_j}$ because of the 45° owner (lemma 4). Thus, $f_i = \frac{7}{2}$.

The terms for boundary edges are derived similarly. □

8 A More Efficient Grid Method

In [18], Wagner and Koren consider the area of the layout as a set of I grid points of resolution γ. The input polygons (shapes) are also treated as collection of grid points. They compute the critical radius at every grid point by visiting on average $\sqrt{I}$ other grid points i.e, total time complexity is $O(I^{1.5})$. Once the critical radius at every grid point is known the integral of the total critical area can be easily computed; the error depends on the grid resolution γ.

Imagine now that the shape grid points are sources of waves that start propagating at the same time with the same speed. The wave has the id of the net where the source grid point belongs. Waves of the same id are considered identical (although they come from different sources). The propagation is done from one grid point t to its immediate neighbors. Assuming that defects are squares, all eight grid points neighboring t vertically horizontally and diagonally are considered to be the immediate neighbors of t. During the propagation exactly two distinct waves are allowed to overlap at any grid point.

The wave propagation will assign to every grid point t a pair of distance labels (d_1, d_2), where d_1 indicates the distance from the source grid point whose wave is the first one to reach t and d_2 indicates the distance from the source grid point of different id whose wave is the second one to reach t. Clearly d_2 is the critical radius at point t. During the propagation we allow exactly two distinct waves to overlap. When a grid point p is visited by wave w the following happens: 1) if p has already been visited by wave w or if p has already been visited by two other distinct waves the propagation stops at p. 2) Otherwise, p updates its labels (updates label d_1 if w is the first wave to reach p or label d_2 if w is the 2nd wave to reach p) and propagates w to its immediate neighbors (grid points) increasing the distance label by 1.

This construction corresponds to an implicit mechanical construction of the 2nd order L_∞ Voronoi diagram and thus correctness follows from the previous discussion. Every grid point is visited by the waves exactly twice and thus the critical radius at every grid point can be computed in $O(I)$ time. Then the critical area can be computed as described in [18].

9 Conclusion

We have shown how to compute and use the second order Voronoi diagram of rectilinear polygons to efficiently compute critical area for shorts in a single layer. The value of critical area can then be used in evaluating yield. Furthermore we can use the Voronoi diagram to identify the places in the layout that are most vulnerable to spot defects. Note that Voronoi edges provide immediate information about the partial critical area caused by the corresponding pair of layout edges. In other words, we can readily obtain information about those layout edges that contribute most to critical area. By slightly perturbing such edges we may be able to reduce the value of critical area and thus increase yield. Note that after moving an edge the Voronoi diagram can be updated dynamically in time proportional to the changes caused to the diagram because of the move. Thus, the Voronoi diagram approach is suitable for an interactive critical area tool. Note also that the *topographic map* of [18] essentially provides a bitmapped version of the more critical Voronoi edges. The Voronoi diagram approach can also be extended to layouts with shapes in arbitrary orientations (see [13]).

Acknowledgment. The first author would like to thank Dr. Dan Ostapko for reading through an earlier draft and providing useful comments. The second author's research is supported in part by the Office of Naval Research under the Grants No. N00014-95-1-1007 and No. N00014-97-1-0514.

References

[1] F. Aurenhammer, "Voronoi diagrams: A survey of a fundamental geometric data structure," *ACM Comput. Survey,* 23 1991, 345-405.

[2] F. Aurenhammer and R. Klein, "Voronoi Diagrams," chapter 18, *Textbook on Computational Geometry*, J.Sack and G. Urrutia (eds), to appear.

[3] F. Dehne and R. Klein, "The Big Sweep: On the power of the Wavefront Approach to Voronoi Diagrams," *Algorithmica*(1997), 17, 19-32.

[4] A.V. Ferris-Prabhu, "Modeling the Critical Area in Yield Forecast," *IEEE J. of Solid State Circuits*, vol. SC-20, No4, Aug. 1985, 874-878.

[5] A.V. Ferris-Prabhu, "Defect size variations and their effect on the critical area of VLSI devices," *IEEE J. of Solid State Circuits*, vol. SC-20, No 4, Aug. 1985, 878-880.

[6] S. Gandemer, B.C. Tremintin, and J.J. Charlot, "Critical Area and critical levels calculation in IC yield modeling," *IEEE J. of Solid State Circuits*, vol.35, No 2, Feb. 1988, 158-166.

[7] L. Guibas and J. Stolfi, "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams", *ACM Transactions on Graphics*,Vol.4, No 2, April 1985, 74-123

[8] I. Koren, "The effect of scaling on the yield of VLSI circuits", *Yield Modeling and defect Tolerance in VLSI circuits* W.R. Moore, W. Maly, and A. Strojwas Eds., Bristol UK: Adam-Hilger Ltd., 1988, 91-99.

[9] D. T. Lee, "On k-nearest neighbor Voronoi diagrams in the plane," *IEEE Trans. Comput.,* Vol. C-31, No. 6, June 1982, 478-487.

[10] W. Maly, "Modeling of lithography related yield losses for CAD of VLSI circuits," *IEEE Transactions on Computer-Aided Design*, vol. CAD-4, no 3, 166-177, July 1985.

[11] W. Maly, "Computer Aided Design for VLSI Circuit Manufacturability," *Proc. IEEE*, Feb. 90, 356-392.

[12] W. Maly, and J. Deszczka, "Yield Estimation Model for VLSI Artwork Evaluation," *Electron Lett.* vol 19, no.6, 226-227, March 1983.

[13] E. Papadopoulou and D.T. Lee, "L_∞ Voronoi Diagrams and Applications to VLSI Layout and Manufacturing," Manuscript. Extended Abstract submitted to *International Symposium on Algorithms and Computation, 1998.*

[14] J. Pineda de Gyvez, C. Di, "IC Defect Sensitivity for Footprint-Type Spot Defects," *IEEE Trans. on Computer-Aided Design*, vol. 11, no 5, 638-658, May 1992.

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

[16] C. H. Stapper and R. J. Rosner, "Integrated Circuit Yield Management and Yield Analysis: Development and Implementation," *IEEE Trans. on Semiconductor Manufacturing* Vol. 8, No.2, 1995, 95-101.

[17] C.H. Stapper, "Modeling of Defects in integrated circuits photolithographic patterns," *IBM J. Research and Development*, vol.28, no.4, 461-475, 1984.

[18] I. A. Wagner and I. Koren, "An Interactive VLSI CAD Tool for Yield Estimation," *IEEE Trans. on Semiconductor Manufacturing* Vol. 8, No.2, 1995, 130-138.

[19] H. Walker, "VLASIC system user manual, release 1.3," CMU, Aug 1990.

[20] H. Walker and S.W. Director, " VLASIC: A yield simulator for integrated circuits," *IEEE Trans. on Computer-Aided Design*, vol. CAD-5, no 4, 541-556, Oct. 1986.