$L\infty$ Voronoi Diagrams and Applications to VLSI Layout and Manufacturing

Evanthia Papadopoulou

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

Abstract. In this paper we address the L_∞ Voronoi diagram of polygonal objects and present applications in VLSI layout and manufacturing. We show that in L_∞ the Voronoi diagram of segments consists only of straight line segments and is thus much simpler to compute than its Euclidean counterpart. Moreover, it has a natural interpretation. In applications where Euclidean precision is not particularly important the L_∞ Voronoi diagram can provide a better alternative. Using the L_∞ Voronoi diagram of polygons we address the problem of calculating the *critical area* for shorts in a VLSI layout. The critical area computation is the main computational problem in VLSI yield prediction.

1 Introduction

The Voronoi diagram of polygonal objects has been given considerable attention because of its numerous applications in diverse areas such as biology, geography, robot motion planning, computer graphics. In this paper we present applications of Voronoi diagrams in VLSI layout and manufacturing. It is well known that the ordinary Voronoi diagram of polygonal objects has linear combinatorial complexity and consists of straight line segments and parabolic arcs. Several efficient algorithms using divide and conquer, plane sweep, or incremental construction are known for its computation (see [2,3] for a survey). However, the existence of parabolic arcs in the diagram makes it hard to compute in practice. In a recent paper [9], a new complexity model for geometric algorithms was introduced, called the *degree* of an algorithm, which characterizes the complexity of the test computations of an algorithm. In the construction of the Voronoi diagram of segments Burnikel showed that the well-known *incircle test* can be solved with degree 40 [4,9]. This is very high for practical problems involving Voronoi diagrams.

In a recent paper Aichholzer and Aurenhammer [1] introduced a new type of skeleton for polygonal objects in the plane called the *straight skeleton*. The straight skeleton consists of angular bisectors between edges and thus consists of straight line segments. The straight skeleton captures the *shape* of the defining elements in a natural manner. Its main advantage over the Voronoi diagram is the elimination of parabolic arcs. However, straight skeletons do not provide the proximity information that Voronoi diagrams do and thus, they do not always provide an alternative solution to Voronoi diagrams.

K.-Y. Chwa and O. H. Ibarra (Eds.): ISAAC'98, LNCS 1533, pp. 9–19, 1998.

In applications where Euclidean accuracy is not particularly important a practical solution to the high degree problem of Euclidean Voronoi diagrams of segments may be the use of a different geometry, in particular, the L_∞-metric. The L_∞ Voronoi diagram of polygons (or segments) consists of straight line segments and its combinatorial complexity is similar to the Euclidean case. Moreover, if the input vertices are on rational coordinates the Voronoi vertices are also rational. An intuitive way to view the L_∞ Voronoi diagram of polygonal shapes is to view it as the locus of points corresponding to centers of isothetic squares[1] that touch the boundary in at least two points. In contrast, the Euclidean Voronoi diagram can be regarded as the locus of centers of circles that touch the boundary in at least two points. Note that in the rectilinear case where the edges of the polygons are either horizontal or vertical, the L_∞ Voronoi diagram coincides with the straight-skeleton of [1].

Our motivation for considering L_∞ Voronoi diagrams comes from applications in VLSI layout and manufacturing, as described below. Note that layout shapes have many orthogonal edges but are not necessarily rectilinear. The proximity information preserved in the L_∞ Voronoi diagram can be used to calculate the *critical area*, a measure used in estimating the sensitivity of a VLSI layout to defects during manufacturing. The critical area calculation is the main computational problem in predicting the *yield* of a VLSI chip. Other applications of Voronoi diagrams in extracting from the physical description of a design the equivalent resistance are discussed in [12,16]. The L_∞ version would be as good in this case and much simpler to obtain [15]. Due to the simplicity of L_∞ Voronoi diagrams we expect them to also find applications in other areas. For example, in [17] an automatic generation for finite element meshes for multiply connected planar domains with polygonal boundaries is described. The Euclidean Voronoi diagram of the domain was used as a starting point and could be substituted by the L_∞ version [15] which is easier to compute.

2 L_∞ Voronoi Diagrams

The L_∞ distance between two points $p = (x_p, y_p)$ and $q = (x_q, y_q)$ is the maximum between the horizontal and the vertical distance between p and q i.e., $d(p,q) = \max\{d_x(p,q), d_y(p,q)\}$ where $d_x(p,q) = |x_p - x_q|$ and $d_y(p,q) = |y_p - y_q|$. The L_∞ distance between a point p and a line l is $d(p,l) = \min\{d(p,q), \forall q \in l\}$. Intuitively, the L_∞ distance between two elements (points or lines) can be defined in terms of the smallest isothetic square touching the two elements. We shall refer to a line of slope (± 1) simply as a *45-degree* line, denoted 45° line.

Consider a point p and the four 45° rays emanating away from p (see Fig. 1). They partition the plane into four quadrants. In each quadrant the L_∞ distance between p and any point q simplifies to either the vertical (q in upper and lower quadrant) or the horizontal distance (q in right and left quadrant) between p and q. Any non-45° line l that does not contain p intersects two 45° rays at

[1] An isothetic square is one whose sides are parallel to the coordinate axes.

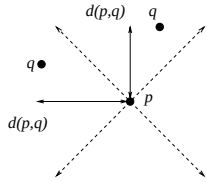

Fig. 1. The 45° rays emanating from p partition the plane into four quadrants.

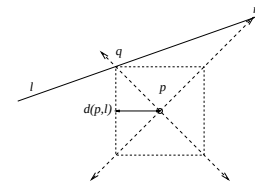

Fig. 2. The L_∞ distance form p to l is $d_x(p,q)$.

points q and r. Assuming that $d_x(p,q) < d_x(p,r)$, the L_∞ distance between p and l is clearly $d_x(p,q)$ (see Fig. 2).

Lemma 1. *If the L_∞ distance from a point p to a line l of slope $\pm s, s \geq 0$ is r, the Euclidean distance from p to l is $S_l r$ where $S_l = \frac{s+1}{\sqrt{s^2+1}}$.*

The *bisector* of two elements (points or lines) is the locus of points equidistant from the two elements. The L_∞ bisector can be regarded as the locus of centers of squares touching the two elements. Figures 3 and 4 illustrate the bisector of two points and two lines respectively. 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 Fig. 3). Without creating any significant difference we will assign one region to one point and consider only the outermost boundary of the bisecting region as the bisector (thick lines in Fig. 3). The bisectors of two lines have slopes given by the following lemma.

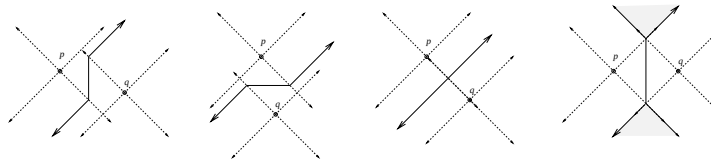

Fig. 3. The L_∞ bisector of two points.

Lemma 2. *The L_∞ bisector of two non-vertical lines l_1 and l_2 of slopes b_1 and b_2, respectively, consists of two lines as follows:*

- *If $b_1 \geq b_2 \geq 0$ the bisector consists of two lines of slopes -1 and $\frac{(b_1+b_2+2b_1b_2)}{(b_1+b_2+2)}$.*
- *If $b_1 \leq b_2 \leq 0$ the slopes of the bisector are $+1$ and $\frac{(-b_1-b_2+2b_1b_2)}{(b_1+b_2-2)}$.*
- *If $b_1 > 0$ and $b_2 < 0$ the bisector has slopes $\frac{b_2-b_1+2b_1b_2}{b_1+b_2}$ and $\frac{b_1+b_2}{b_1-b_2+2}$.*

Note that if one of these two lines is vertical (i.e., $b_1 = \pm\infty$), the L_∞ bisector consists of a 45° line and a line of slope $(1 + 2b_2)$ for $b_2 > 0$ (resp. $(-1 + 2b_2)$ for $b_2 < 0$). Moreover if $b_2 = -b_1$ the bisector is a vertical and a horizontal line and if $b_1 = \infty$, $b_2 = 0$ the bisector consists of two 45° lines.

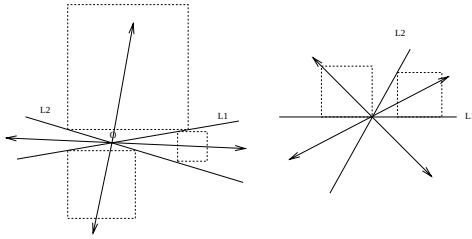

Fig. 4. The L_∞ bisector of two lines.

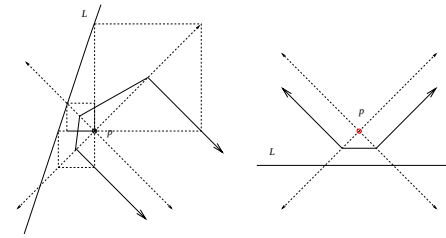

Fig. 5. The L_∞ bisector between a point and a line.

The bisector of a point p and a line l ($p \notin l$, see Fig. 5) consists of at most four parts, one for each quadrant of p. Each part corresponds to the portion of the bisector between l and a vertical or horizontal line through p depending on the quadrant of p. Thus, each part is a line segment or ray whose slopes can be derived by Lemma 2. The unbounded parts of the bisector are always parallel 45° rays (see Fig. 5). For a point p along a non-orthogonal line l the bisector is a 45° line through p and for a point p along an orthogonal line l the bisector is the area enclosed by the 45° lines through p. For simplicity we only consider the boundary of this region as the bisector between p and l.

The L_∞ Voronoi diagram of a set of elements $S = \{e_1, e_2, \ldots, e_n\}$, denoted $\mathcal{V}(S)$, is a partitioning of the plane into regions, called *Voronoi cells*, each of which is associated with an element, called the *owner*, of the cell. The boundary that borders two Voronoi cells is called a *Voronoi edge*, and adjacent Voronoi edges of each Voronoi cell are common to a *Voronoi vertex*. The collection of Voronoi edges and vertices is also called the Voronoi diagram. Fig. 6 illustrates the L_∞ Voronoi diagram in the interior of a simple polygon (the *medial axis*) and Fig. 7 illustrates the L_∞ Voronoi diagram of polygons.

Let G be a *planar straight line graph* on n points in the plane as defined in [1,3] i.e., a set of non-crossing line segments spanned by these points. By the above discussion it is clear that the L_∞ Voronoi diagram of a planar straight-line graph G consists only of straight line edges. Moreover, if the vertices of G are on rational coordinates the L_∞ Voronoi vertices are also rational. Hence, the L_∞ Voronoi diagram is computationally much simpler than its Euclidean counterpart. Most of the existing algorithms to compute the Euclidean Voronoi diagram of segments can be easily modified to compute the L_∞ Voronoi diagram in $O(n \log n)$ time. The well known *incircle test* to determine whether an element is in or out of the circle defined by three other elements (lines or points) now simplifies to a test involving the square defined by the three elements. (Note that three elements need not always define a square). In the worst case the L_∞ *incircle test* corresponds to 1) determining the intersection point I of two lines (bisectors), 2) determining the intersection point J of a line and a 45° line through I, and 3) comparing the horizontal or vertical distances between I and J and between I and one of the defining elements. Note that no square root computations are involved.

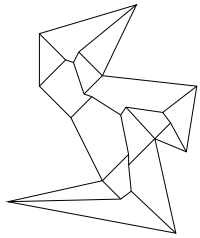

Fig. 6. The L_∞ medial axis of a simple polygon.

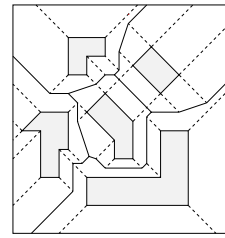

Fig. 7. The L_∞ Voronoi diagram of polygons.

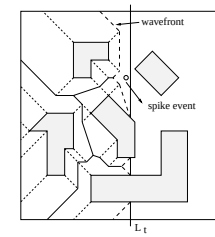

Fig. 8. The L_∞ Voronoi diagram of G_t.

The combinatorial complexity of the L_∞ diagram is similar to the Euclidean one. However, there is some difference in the number of Voronoi vertices produced by reflex angles. The number of 45° rays emanating from a concave vertex v depends on the slopes of the incident edges and can be either one, two, or three. In the Euclidean case the *normals* bisecting a vertex from the incident edges are always two and in the *straight-skeleton* of [1] the corresponding angular bisector is always one. Thus the combinatorial complexity of the L_∞ diagram can be lower or higher than the Euclidean one depending on the input. An exact bound can be derived following the proof of [3] for the Euclidean case.

Lemma 3. *Let G be a planar straight line graph on n points in the plane such that G has t terminals (vertices of degree 1), t_n of which are incident to non-orthogonal edges, r_2 reflex angles inducing two 45° rays, and r_3 reflex angles inducing three 45° rays. The number of (finite and infinite) vertices of the L_∞ Voronoi diagram of G is exactly $2n + t + t_n + r_2 + 2r_3 - 2$.*

Note that in the Euclidean case this bound is $2n + t + r - 2$ where r is the total number of reflex vertices (i.e., $r = r_1 + r_2 + r_3$ for r_1 reflex vertices inducing a single 45° ray) [3]. In the rectilinear case the bound becomes $2n + t - 2$ as for the straight skeletons.

3 A Sweep-Line Algorithm for the L_∞ Voronoi Diagram

In [5], Dehne and Klein described a plane sweep paradigm for the Voronoi diagram of points in "nice metrics". In this section we modify their algorithm to accommodate segments in the L_∞ metric.

Let G be a *planar straight-line graph* on n points. We will compute the L_∞ Voronoi diagram of G, $\mathcal{V}(G)$. Consider a vertical sweep-line L sweeping across the entire plane from left to right. At any instant t of the sweeping process the sweep-line partitions the set of segments (edges) of G into three subsets S_l, S_m and S_r, corresponding to those that lie totally to the left of L, intersect L, and lie totally to the right of L, respectively. The segments in S_m cut L into $|S_m| + 1$ sweepline segments, two of which are unbounded, denoted by L_t. Let S_t denote the portions of segments in S_m to the left of L. In other words, $S_t = \{s(t) \mid s \in S_m\}$ where $s(t)$

denotes the portion of s to the left of L. At every instant t of the sweeping process we compute the Voronoi diagram of $G_t = S_l \cup S_t \cup L_t$. Note that at any instant t the sweep-line segments are treated as being part of G. (In case G is a collection of simple polygons and we only want to compute the Voronoi diagram in the exterior of the polygons we exclude from L_t the segments in the interior of polygons, Fig. 8).

The boundary of the Voronoi cell of each segment in L_t is referred to as an *individual wavefront* and the collection of individual wavefronts for all segments in L_t is called the *wavefront*. In Fig. 8, the wavefront is shown in dashed lines. Note that in the case of points the whole wavefront is a single individual wavefront. Clearly, in the L_∞ metric the wavefront is y-monotone. The Voronoi edges that have an endpoint common with the wavefront are called *spike bisectors*.

As the plane sweep proceeds, the wavefront, and therefore the endpoints of spike bisectors, as well as the endpoints of segments in S_t move continuously to the right until an *event* takes place which causes the wavefront to change. Following [5], we have two kinds of events: 1) events corresponding to the occurrence of a vertex or a vertical edge in G, referred to as *site events* and 2) events corresponding to the intersection point of two neighboring spike bisectors, referred to as *spike events*. A site event induced by a point p takes place at $t = x_p$, where x_p is the abscissa of point p. A spike event C takes place at $t = x_c + w$ where x_c is the abscissa of the intersection point and w is the distance of the intersection from the inducing element.

Throughout the plane sweep we implicitly maintain the wavefront in a data structure referred to as the *sweep-line status* which is implemented as a *height-balanced* binary tree $\mathcal{T}$. In particular, the sweep-line status contains the spike bisectors and the edges in S_t. To simulate the wavefront movement to the right we parameterize the endpoints of spike bisectors and segments in S_t as linear functions of t. At any instant t the wavefront is the polygonal line obtained by connecting the endpoints of the elements in $\mathcal{T}$ in increasing y-coordinate value. The events are organized in a priority queue referred to as the *event list* in increasing priority value. Due to space limit we omit the details of the plane sweep algorithm. The number of site and spike events is $O(n)$ and thus the time complexity of the algorithm is $O(n \log n)$.

4 The L_∞ Voronoi Diagram and the Critical Area Problem

The yield of a VLSI chip is the percentage of functional chips among all chips manufactured. Predicting the yield of a chip is important due to the growing need to control the cost of manufacturing. In this section we describe an important problem in VLSI yield prediction that can be solved accurately and efficiently using the L_∞ Voronoi diagram of a layout.

Yield estimation models are based on the concept of critical area which is a measure reflecting the sensitivity of the layout to spot defects caused by particles such as dust and other contaminants in materials and equipment. The main

reason for yield loss during manufacturing are "extra material" defects causing shorts between different conducting regions. For information on yield estimation and spot defects see for example [6,7,8,10,11,14,18,19,20].

Given 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. Defects are usually modeled as circles and the density function has been estimated to follow the "$1/r^3$" distribution [7,14,18,20]. Existing methods require substantial CPU time for large layouts. They can be summarized into three categories: 1) Geometric methods, where $A(r)$ is computed for several different values of r independently and then the results are used to approximate A_c. The methods to compute $A(r)$ are usually based on *shape-expansion* followed by *shape-intersection* (see [20] for references). For rectilinear layouts there is a more efficient scan-line method [14]. 2) Monte Carlo simulation, where a large number of defects is drawn with their radii distributed according to $D(r)$. The critical area is estimated by dividing the number of defects causing faults over the total number of defects [21]. 3) Grid-based approach, where an integer grid is assumed over the layout and the critical radius (i.e., the radius of the smallest defect causing a fault at this point) of every grid point is computed.

For rectilinear layouts our solution to the critical area problem for shorts (assuming that defects are isothetic squares) appears in [13]. It was shown that the problem can be reduced to computing the *2nd order L_∞ Voronoi diagram* of polygons. Consider an arbitrary point p in the layout. The *critical radius* of p for shorts is the radius of the smallest defect which if centered at p overlaps with two different polygons. Such a defect causes a short between the two different conducting regions represented by the polygons. Thus, the critical radius of p is the distance of p from its second nearest polygon. Given the Voronoi diagram of the polygons in the layout, we subdivide the Voronoi cell of a polygon P by the Voronoi diagram of its neighbors. In other words we obtain the *2nd order Voronoi diagram* of polygons. A region of an element s (edge or reflex vertex) within the Voronoi cell of P is defined as $reg_P(s) = \{x \mid d(s,x) \leq d(t,x), \forall t \in C - P\}$, where C is the given collection of polygons. The critical radius of every point $x \in reg_P(s)$ is determined by the distance of x from element s; we say that s is the owner of $reg_P(s)$.

Having the critical area computation done based on Euclidean Voronoi diagrams would be out of the question in practice due to the difficulty of computing the Voronoi diagram of segments. However, the computation in L_∞ is practical. In [13], we showed that once the 2nd order Voronoi diagram is available, the critical area for shorts in rectilinear layouts can be done analytically. In particular, we showed that the critical area formula can be written as a function of the edges of the 2nd order Voronoi diagram. In the following we generalize this result to general non-rectilinear layouts.

4.1 The Critical Area for Shorts as a Function of Voronoi Edges

We have a layer in a circuit layout consisting of a collection of disjoint simple polygons C in arbitrary orientations. Our goal is to evaluate the integral $A_c = \int_0^\infty A(r)D(r)dr$, where $D(r) = r_0^2/r^3$ and r_0 is a minimum optically resolvable size. 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 disjoint polygons.

Let's assume that we are given the 2nd order L_∞ Voronoi diagram of C. Fig. 7 shows the L_∞ Voronoi diagram of a set of shapes and Fig. 9 shows the 2nd order subdivision within a bounded cell. Then $A_c = \sum_V A_c(V)$ 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.

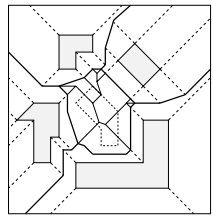

Fig. 9. The 2nd order subdivision within a Voronoi cell.

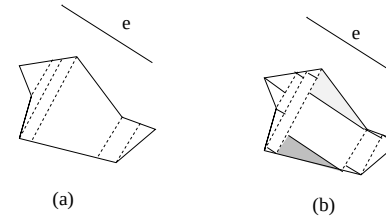

Fig. 10. The decomposition of V into trapezoids.

Let's first concentrate on a single (2nd order) Voronoi cell V having as owner an edge e of slope $\pm s, s \geq 0$. The Voronoi cell of a vertex is considered as the Voronoi cell of a vertical or horizontal edge. Consider a decomposition of V into trapezoids by drawing lines perpendicular to e (lines of slope $\pm 1/s$) emanating from the Voronoi vertices of V (see Fig. 10a). Each trapezoid $\mathcal{T}$ is further decomposed into orthogonal triangles and at most one rectangle $\mathcal{R}$ by drawing lines parallel to e (slope $\pm s$) through its vertices. (In case $\mathcal{T}$ is a slanted parallelogram continue the decomposition recursively). We distinguish between two kinds of triangles, *red* and *blue*, depending on the relevant position of the hypotenuse and the orthogonal apex with respect to e. In particular, if the owner e and the orthogonal apex lie on opposite sides of the hypotenuse the triangle is colored red, otherwise it is colored blue. In Fig. 10b, the lightly shaded triangle is red and the darker shaded one is blue.

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., the L_∞ distance of v_j and v_k from the line l through e respectively. We derive the following formulas for the critical area within an element of the above decomposition.

Lemma 4. *The critical area within a rectangle $\mathcal{R}$, a red triangle T_{red} and a blue triangle T_{blue} is given by the following formulas, using the "r_0^2/r^3" defect density distribution:*

$$A_c(\mathcal{R}) = \frac{r_0^2 S}{2}(\frac{l}{r_j} - \frac{l}{r_k}) \tag{1}$$

$$A_c(T_{red}) = \frac{r_0^2 S}{2}(ST \ln(\frac{r_k}{r_j}) - \frac{l}{r_k}) \tag{2}$$

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

where l is the size of the edge of $\mathcal{R}, T_{red}$ and T_{blue} parallel to e, $r_k, r_j, r_k > r_j$ are the maximum and the minimum critical radius of their vertices, $S = \frac{s+1}{\sqrt{s^2+1}}$, where s is the absolute value of the slope of e, and $T = \frac{ts+1}{|t-s|}$, where t is the absolute value of the slope of the hypotenuse (i.e, the slope of the corresponding Voronoi edge). For $s = \infty$, $S = 1$ and $T = t$.

We can 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 Sl/r_l corresponding to internal decomposition edges cancel out. Similarly, for logarithmic terms involving endpoints of the decomposition other than Voronoi vertices. Thus, the critical area within V can be written as a function of Voronoi edges. Let's color the Voronoi edges that induce the hypotenuse of red triangles as red and those inducing the hypotenuse of blue triangles as blue. Voronoi edges that are incident to a rectangle or induce an orthogonal edge of a triangle must be either parallel or perpendicular to the owner of the cell. Those Voronoi edges that are parallel to the owner are referred to as *prime* and they are portions of bisectors of two parallel edges. Prime Voronoi edges are colored red if the interior and the owner of the cell lie on opposite sides of the edge; otherwise they are colored blue. Those Voronoi edges that are perpendicular to the owner are colored *neutral*. In the formulas of Lemma 4, terms corresponding to red edges get added while terms corresponding to blue edges get subtracted. Neutral edges do not contribute in the formulas. The boundary of the layout is assumed to be a rectangle whose intersection points with the Voronoi diagram are treated as Voronoi vertices. Boundary edges are treated as Voronoi edges but they contribute at most once in the formulas.

We derive the following theorem, which is given without a proof due to space limitation.

Theorem 1. *Given the 2nd order L_∞ Voronoi diagram of 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:*

$$A_c = r_0^2(\sum_{\substack{red,\\ prime\ e_i}} \frac{S_i l_i}{r_i} - \sum_{\substack{blue,\\ prime\ e_m}} \frac{S_i l_i}{r_i} + \frac{1}{2}\sum_{\substack{red, nonprime\ e_j,\\ wrt\ e}} S_e^2 T_e \ln\frac{r_k}{r_j} - \frac{1}{2}\sum_{\substack{blue, nonprime\ e_j,\\ wrt\ e}} S_e^2 T_e \ln\frac{r_k}{r_j} + \frac{B}{2})$$

where l_i and r_i denote the length and the critical radius of a prime Voronoi edge e_i, and $r_k, r_j, r_k > r_j$ denote the maximum and the minimum critical radius of a non-prime Voronoi edge e_j. The factors S_e and T_e are as defined in Lemma 4 for each owner e of the non-prime edge e_j. B is the sum of the blue terms for prime boundary edges and appears as a correction factor since boundary edges contribute at most once.

Acknowledgment

I wish to thank Prof. D.T. Lee for valuable contributions.

References

1. O. Aichholzer, F. Aurenhammer, "Straight Skeletons for general Polygonal Figures in the Plane", *Proc. 2nd Int. Computing and Combinatorics Conference,1996* Lecture Notes in Computer Science 1090, 117-126. 9, 10, 12, 13
2. F. Aurenhammer, "Voronoi diagrams: A survey of a fundamental geometric data structure," *ACM Comput. Survey,* 23 1991, 345-405. 9
3. F. Aurenhammer and R. Klein, "Voronoi Diagrams" chapter 18, *Textbook on Computational Geometry*, J.Sack and G. Urrutia (eds), to appear. 9, 12, 13
4. C. Burnikel, K, Melhorn S. Scirra, "How to compute the Voronoi Diagram of Line Segments: Theoretical and Experimental Results" *Proc. 2nd Annu. European Symp. on Algorithms, 1994*, LNCS 855, 227-239. 9
5. F. Dehne and R. Klein, "The Big Sweep": On the power of the Wavefront Approach to Voronoi Diagrams", *Algorithmica*(1997), 17, 19-32. 13, 14
6. 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 15
7. 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, No4, Aug. 1985, 878-880. 15
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 15
9. G. Liotta, F.P. Preparata, and R. Tamassia, " Robust Proximity Queries in Implicit Voronoi diagram", Brown University CS-96-16. 9
10. W. Maly, "Computer Aided Design for VLSI Circuit Manufacturability", *Proc. IEEE*, Feb. 90, 356-392. 15
11. W. Maly, and J. Deszczka, "Yield Estimation Model for VLSI Artwork Evaluation", *Electron Lett.* vol 19, no.6, 226-227, March 1983 15
12. S. N. Meshkat and C. M. Sakkas. " Voronoi diagram for multiply-connected polygonal domains II: Implementation and application", *IBM J. of Research and Development*, Vol. 31, No. 3, May 1987 10
13. E. Papadopoulou, D.T. Lee, "Critical Area Computation – A new Approach", Proc. *International Symposium on Physical Design*, 1998, 89-94. 15
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
15. V. Srinivasan Personal Communication. 10

16. V. Srinivasan, L.R. Nackman, " Voronoi diagram for multiply-connected polygonal domains II: Algorithm", *IBM Journal of Research and Development*, Vol. 31, No. 3, May 1987 10
17. V. Srinivasan, L.R. Nackman, J.M. Tang, and S.N. Meshkat, "Automatic Mesh Generation Using the Symmetric Axis Transformation of Polygonal Domains", *Proceedings of the IEEE*,Vol. 80, No. 9, Sept. 1992, 1485-1501. 10
18. C.H. Stapper, "Modeling of Defects in integrated circuits photolithographic patterns", *IBM J. Research and Development*, vol.28, no.4, 461-475, 1984. 15
19. 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. 15
20. 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. 15
21. 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. 15