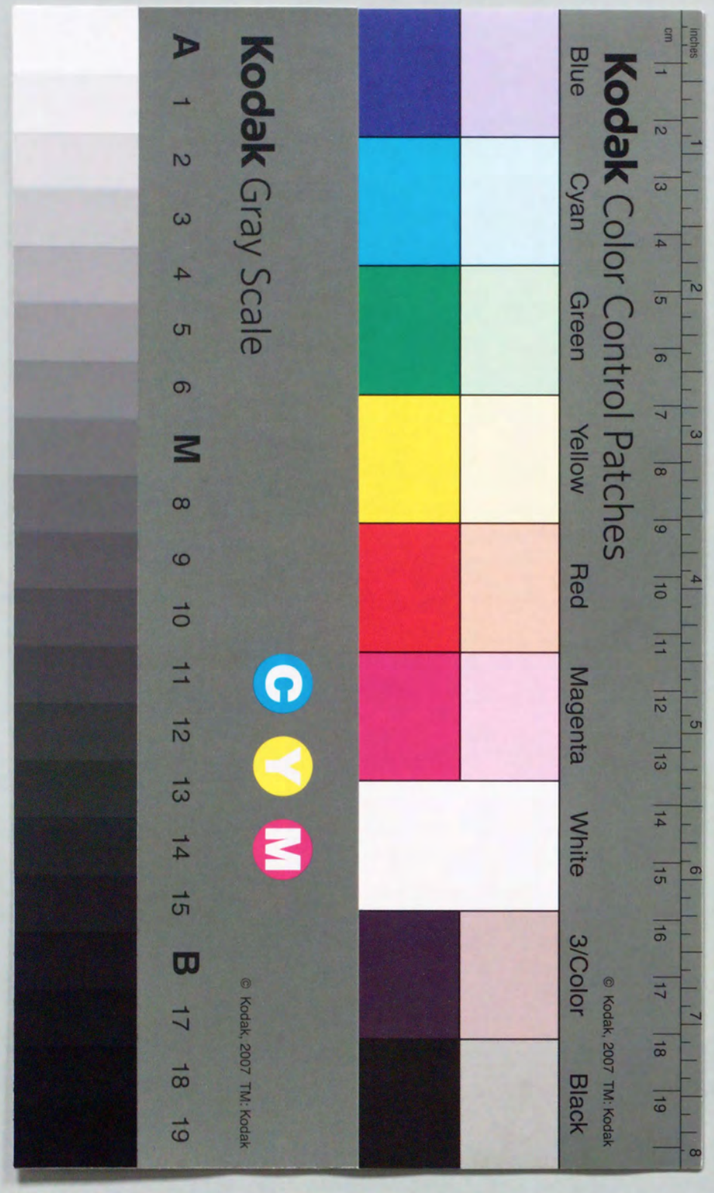

Geometric Algorithms with Imprecise Input

Supervisor
Nobuki Tokura

Author
Takayuki Nagai

Submitted for the degree of Doctor of Engineering,

Osaka University
January 2000

Department of Information and Computer Sciences,
Faculty of Engineering Science, Osaka University

Geometric Algorithms with Imprecise Input

Supervisor
Nobuki Tokura

Author
Takayuki Nagai

Submitted for the degree of Doctor of Engineering,

Osaka University
January 2000

Department of Information and Computer Sciences,
Faculty of Engineering Science, Osaka University

Abstract

Computational geometry is a field of computer science that addresses geometric problems. To satisfy the increasing need for geometric computation that appears in many application areas such as computer graphics, computer aided design, circuit layout, robotics and geographic information systems, many efficient data structures and algorithm techniques have been developed.

On the other hand, much attention has been paid on the *accuracy* of the computed results. The accuracy of the computation is often dominated by the accuracy of floating point arithmetic that is widely used in engineering computation. To fill up the weakness of the arithmetic, many techniques have been developed. However, with the aid of powerful computers, exact computation is now becoming a reasonable choice for geometric computing.

With the decrease of arithmetic error, the accuracy of computed results should be considered with *input error*. While the former can be eliminated with additional computational costs, the latter is essential in real input data such as spatial data. To guarantee the accuracy of computed results, we need to turn our attention from arithmetic error to input error.

In the dissertation, we consider geometric problems under an assumption that all input values are given by accuracy guaranteed numbers. Our primary concern is exact analysis and computation of the effect of input error. For the purpose, we formulate geometric problems as computation of the trace of the boudaries of all possible output figures.

In Section 3, we address convex hull problem. For n points in the plane, we show that the intersection and the union of all possible convex hulls can be computed in $O(n \log n)$ time. We also show that the intersection of all possible convex hulls can be computed incrementally in $O(n \log n)$ time (best case) or in $O(n^2)$ time (worst case) if a non-empty solution for any three points is given.

In Section 4, we address the problem of calculating the diameter of a set

of points. For n points in the plane, we show that the minimum of possible diameters can be computed in $O(n \log n)$ time, and show that the maximum of possible diameters can be computed in $O(n \log^* n \log n)$ expected time.

In Section 5, we address the problem of Voronoi diagram construction. For n points (called *sites*) in the plane, we show that the intersections of all possible Voronoi regions can be computed for all sites in $O(n \log^* n \log n + k)$ time. The unions of all possible Voronoi regions can be computed for all sites in $O(n^2 \alpha(n))$ time (best case) or in $O(n^2 \log^* n \log n)$ time (worst case). In the previous statements, k denotes the number of overlaps between input points and $\alpha(n)$ denotes the inverse of Ackermann's function.

We show that accuracy guaranteed solutions are available for some geometric problems with imprecise input in reasonable computational costs.

List of Publications

- Papers

[1] T. Nagai, S. Yasutome and N. Tokura, *Convex hull problem with imprecise input and its solution*, IEICE Transactions vol. J81-D-I, no. 6, pp.615-625, 1998 in Japanese.

[2] T. Nagai, S. Yasutome and N. Tokura, *Voronoi region with imprecise input*, IEICE Transactions vol. J83-D-I, no.1, pp.68-79, 2000 in Japanese.

[3] T. Nagai, S. Yasutome and N. Tokura, *Convex hull problem with imprecise input*, Proceedings of Japan Conference on Discrete and Computational Geometry '98, Lecture Notes in Computer Science*(will be appeared)*.

[4] T. Nagai, S. Yasutome and N. Tokura, *Convex hull and diameter of a set of points with imprecise coordinates (submitted to IEICE Transactions)* in Japanese.

- Technical Reports

[5] T. Nagai, S. Yasutome and N. Tokura, *Toward exact decision of segments intersection*, Technical Report of IEICE, COMP96-23, pp.11-20, 1996 in Japanese.

[6] T. Nagai, S. Yasutome and N. Tokura, *Convex Hull problem with Imprecise input*, Technical Report of IEICE, COMP96-53, pp.31-40, 1996 in Japanese.

[7] T. Nagai, S. Yasutome and N. Tokura, *Voronoi Region with Imprecise Input*, Technical Report of IEICE, COMP98-31, pp.85-92, 1998 in Japanese.

[8] T. Nagai, S. Yasutome and N. Tokura, *Cluster Voronoi diagram for point clusters that are equivalent under parallel translation* Technical Report of IEICE, COMP99-73, pp.33-40, 2000.

Contents

List of Figures

Chapter 1

Introduction

Computational geometry is a field of computer science that addresses geometric problems. To satisfy the increasing need for geometric computation that appears in many application areas such as computer graphics, computer aided design, circuit layout, robotics and geographic information systems, many efficient data structures and algorithm techniques for *geometric problems* have been developed. Since its birth in 1970s[31], computational geometry has developed to a vast, attractive area of computer science. Many fundamental geometric problems such as convex hull, Voronoi diagram, point location, shortest path, have been proposed and solved efficiently in a sense of computational complexity. In parallel to the theoretical success, implementations of theoretically efficient algorithms has been a great concern since they need complex data structures and procedures. Today, implementations of geometric algorithms are available as software packages such as LEDA[21],CGAL[5].

On the other hand, much attention has been paid on the *accuracy* of the computed results. The accuracy of the computation is often dominated by the accuracy of floating point arithmetic that is widely used in engineering computation. To fill up the weakness of the arithmetic[16], many techniques have been developed [10],[23],[15], [22],[18],[33]. However, with the aid of powerful computers, exact computation is now becoming a reasonable choice for geometric computing[3],[36]. The arithmetic error is now becoming a matter of choice.

With the decrease of arithmetic error, the accuracy of computed results must be considered with *input error*. The big difference between arithmetic error and input error is that we can control the former with additional computational costs while the latter is essential in real input data such as spatial data[12]. To guar-

antee the accuracy of computed results, we need to turn our attention from arithmetic error to input error.

Many existing geometric algorithms assume that input data is accurate; however, real input data does not satisfy such an assumption. Input coordinates are often only approximations of their true values because of round-off errors, measurement errors, and so on. If input error is inevitable, we can not rely on any results based on the assumption. Instead, we need *accuracy guaranteed* results to get meaningful information. In recent years, accuracy guaranteed computation has attracted great attentions in a field of numerical analysis [17]. There should be potential needs for accuracy guaranteed computation also in geometric computing.

There are some studies on geometric problems with imprecise input. Franciosa, Gaibisso, and Talamo[11] consider to derive a good approximation of the convex hull of a set of points in the plane under the assumption that only arbitrary finite approximations of the real valued coordinates can be known. Cai and Keil[4] also interested in capturing meaningful information from imprecise input data. They consider the problem of computing visibility information in an inaccurate simple polygon. However, compared to the variety of geometric problems, there are too few studies about geometric problems with imprecise input.

In the dissertation, we consider geometric problems under an assumption that all input values are given by accuracy guaranteed numbers. Our primary concern is exact analysis and computation of the effect of input error. Since there are many possibilities of exact locations of points, many possible output figures are considered for a problem instance. For the purpose, we formulate geometric problems as computation of the trace of the boudaries of all possible output figures. The trace explicitly represents the accuracy of a possible output figure. Since the trace itself is a geometric object, we can compute the trace efficiently in the framework of computational geometry.

In the dissertation, we consider three geometric problems: convex hull, diameter of a set of points and Voronoi diagram construction. The computational cost is measured under the *real* RAM(Random Access Machine) model[31], which can store and process real values (not just integers) in constant time. In convex hull problem, we compute the intersection and the union of all possible convex hulls. In a computation of the diameter of a set of points, we compute the minimum and the maximum of possible diameters. In Voronoi diagram construction, we compute the intersection and the union of all possible Voronoi regions for each

sites.

In Section 2, we introduce terminologies and definitions commonly used in the dissertation. In Section 3, we address the problem of convex hull. For n points in the plane, we show that the intersection and the union of all possible convex hulls can be computed in $O(n \log n)$ time. We also show that the intersection of all possible convex hulls can be computed incrementally in $O(n \log n)$ time (best case) or in $O(n^2)$ time (worst case) if a non-empty solution for any three points is given. In Section 4, we address the problem of diameter of a set of points. For n points in the plane, the minimum of possible diameters can be computed in $O(n \log n)$ time. The maximum of possible diameters can be computed in $O(n \log^* n \log n)$ expected time. In Section 5, we address the problem of Voronoi diagram. For n points in the plane, the intersections of all possible Voronoi regions can be computed for all sites in $O(n \log^* n \log n + k)$ time. The unions of all possible Voronoi regions can be computed for all sites in $O(n^2 \alpha(n)$ time (best case) or in $O(n^2 \log^* n \log n)$ time (worst case). In the previous statement, k denotes the number of overlaps between input points and $\alpha(n)$ denotes the inverse of Ackermann's function.

In the dissertation, we formalize some geometric problems with imprecise input as problems of constructing the trace of the boundaries of all possible output figures. Based on the analysis of geometric properties of traces, we show that accuracy guaranteed solutions are available for some geometric problems in reasonable computational costs.

Chapter 2

Preliminaries

In the dissertation, we assume *real*-RAM(random-access machine) model[31] as a computational model. In the model, each storage location is capable of holding a single real number. In addition, the following operations are primitive and are available at unit time (the statements are cited from [31]):

1. The arithmetic operations(+,-, ×, /)
2. Comparisons between two real numbers($<, \leq, =, \neq, \geq, >$)
3. Indirect addressing of memory (integer address only)
4. k-th root, trigonometric functions, EXP, and LOG (in general, analytic functions)

An imprecise input point is modeled as a pair of x-interval and y-interval. We call such a point ε-Point defined as follows.

Definition 2.1 (ε-Point)
We regard the following set of points as a point (x,y) whose coordinates contain input error $\epsilon_x, \epsilon_y (> 0)$ respectively.

$$\{(p,q) | x - \epsilon_x \leq p \leq x + \epsilon_x, y - \epsilon_y \leq q \leq y + \epsilon_y\}$$

We call this set an ε-Point $(x, y, \epsilon_x, \epsilon_y)$.

An ε-Point represents a point whose location is ambiguous but known up to some accuracy (see Fig. 2.1). For problems in the dissertation, a set of ε-Points

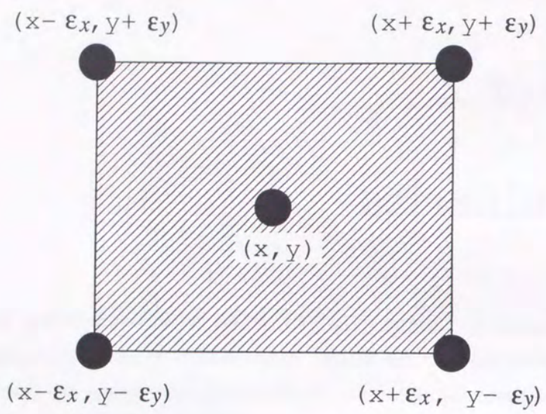

Figure 2.1: ε-Point$(x, y, \epsilon_x, \epsilon_y)$

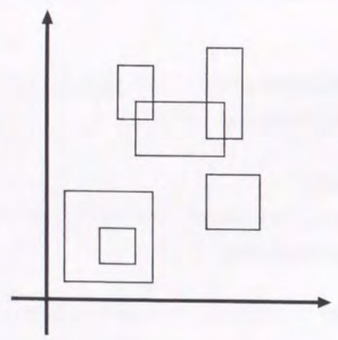

Figure 2.2: An example configuration of ε-Points

is given as an input. ε-Points would have different sizes of input error, and are allowed to intersect each other (for example, see Fig. 2.2). To distinguish ε-Points from ordinary points, we use the notation $\hat{}$ (e.g. $\hat{p}$,$\hat{q}$) for ε-Points.

Next, we define the meaning of a set of ε-Points, by specifying the figures it represents.

Definition 2.2 (Possible Sets of Points) *Let S be a set of ε-Points $\hat{p_1}$, ..., $\hat{p_n}$. "A set T of points $q_1, \ldots, q_n$ is a possible set of points of S" means that there is a one-to-one function $f_T : S \to T$ s.t. $f_T(\hat{p_i}) = q_j$ iff $q_j \in \hat{p_i}$. We call each q_j a possible point of $\hat{p_i}$. $A(S)$ dentoes all possible sets of points.*

Definition 2.2 states that the cardinality of S and T must be the same. Each input point is considered to occupy exactly one position in any possible set of points. Sometimes, we use the term *possible arrangement* instead of possible set when we refer to the locations of each possible points.

In the following, we introduce geometric terms commonly used in the dissertation.

Definition 2.3 (Euclid distance) *For points p and q, $d(p, q)$ represents the euclid distance between them.*

Definition 2.4 (Convex Hull) *Convex hull $conv(S)$ of a set S of points in the plane is the smallest convex polygon containing all points in S.*

Definition 2.5 (Left and Right Half-Plane of a Directed Line) *For a directed line l in the plane, $H_{\mathcal{L}}(l)$ denotes the closed left half-plane of l. $H^-_{\mathcal{L}}(l)$ denotes the open left half-plane of l. Similarly,$H_{\mathcal{R}}(l)$ and $H^-_{\mathcal{R}}(l)$ denote the right half-plane of l.*

Definition 2.6 (Angle of a Directed Line) *We say "the angle of a directed line in the plane is θ" if its direction vector is $(k\cos\theta, k\sin\theta)$ for some positive constant k.*

Definition 2.7 (Left Tangent/Right Tangent) *Let l be a directed tangent of a region R in the plane. l is a left tangent of R if it contains R in $H_{\mathcal{L}}(l)$, or is a right tangent of R if it contains R in $H_{\mathcal{R}}(l)$. $t_{\mathcal{L}}(R, \theta)$ and $t_{\mathcal{R}}(R, \theta)$ denotes the left and the right tangent of R with angle θ respectively.*

Definition 2.8 (Common tangents) *For regions R_1 and R_2 in the plane, let l be a common tangent of them directed from R_1 to R_2. We distinguish l in four types as follows:*

1.common left tangent	$t_{\mathcal{CL}}(R_1, R_2)$	*if* $R_1 \subseteq H_{\mathcal{L}}(l) \wedge R_2 \subseteq H_{\mathcal{L}}(l)$
2.common right tangent	$t_{\mathcal{CR}}(R_1, R_2)$	*if* $R_1 \subseteq H_{\mathcal{R}}(l) \wedge R_2 \subseteq H_{\mathcal{R}}(l)$
3.common left-right tangent	$t_{\mathcal{LR}}(R_1, R_2)$	*if* $R_1 \subseteq H_{\mathcal{L}}(l) \wedge R_2 \subseteq H_{\mathcal{R}}(l)$
4.common right-left tangent	$t_{\mathcal{RL}}(R_1, R_2)$	*if* $R_1 \subseteq H_{\mathcal{R}}(l) \wedge R_2 \subseteq H_{\mathcal{L}}(l)$

Chapter 3

Convex Hull Problem

In this chapter, we address the problem of computing convex hull of accuracy guaranteed points in the plane. To guarantee the accuracy of a computed convex hull, we calculate the intersection and the union of all *possible convex hulls*. A possible convex hull is the convex hull of a possible set of points. For example, in Fig. 3.1, two possible convex hulls are shown. Since a point may occupy any location within a tolerance of input error, there are infinitely many possible convex hulls.

To demonstrate our idea, consider a simple accuracy bound: the maximum of input error. If the size of input error is uniform, the accuracy bound is reasonable as shown in Fig 3.2. However, if various sizes of error appears, such a bound becomes meaningless. Consider a case that a point with the largest error is in the middle of others with small errors (see Fig. 3.3). The point dominates the bound although it does not contribute to the convex hull. If others are far from the point, such a bound is meaningless. To retrieve a good bound, the location of each points and their contribution to the convex hull should be considered as well as the size of each error.

To compute a good bound, we concentrate on the intersection and the union of all possible convex hulls, which we call *inner convex hull* and *outer convex hull*. Clearly, any possible hull lies between these two hulls; the distance from it to them gives the accuracy of the hull. Remember that possible convex hulls determine the inner and the outer convex hull: both hulls are not affected by input errors that does not contribute to possible hulls. For example, for the set of ε-Points in Fig. 3.3, the inner and the outer convex hull become as shown in Fig. 3.4. In the example, the point with largest error has no effect on both hulls.

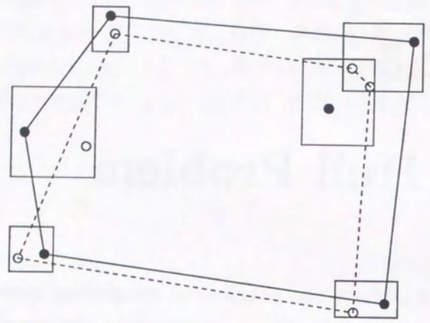

Figure 3.1: Two possible convex hulls for a set of ε-Points

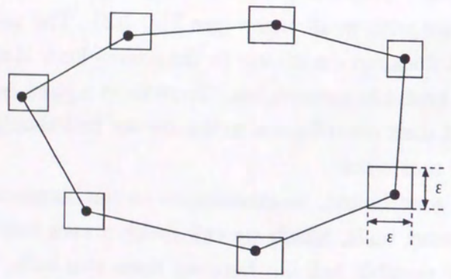

Figure 3.2: The accuracy of the possible convex hull is ϵ

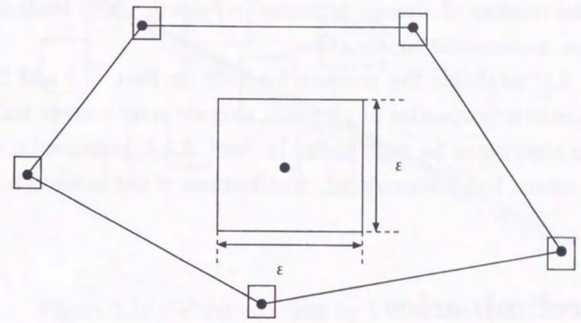

Figure 3.3: Is ϵ a reasonable accuracy of the possible convex hull?

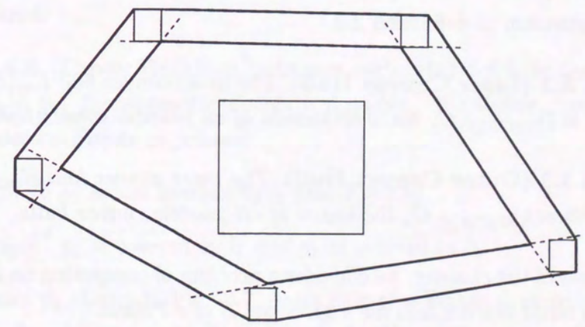

Figure 3.4: Example of an inner and an outer convex hull

We show that for n ε-Points in the plane, the inner and the outer convex hull can be constructed in $O(n \log n)$ time. Also we show that if a non-empty inner convex hull is given, we can update the inner convex hull in $O(\log k) \sim O(k)$ time (k is the number of already processed ε-Points); which leads to $O(n \log n)$ $\sim O(n^2)$ time incremental construction.

In Sect. 3.1, we define the problem formally. In Sect. 3.2 and Sect. 3.3, we examine geometric properties of an outer and an inner convex hull, and show construction algorithms for each hulls. In Sect. 3.3.4, incremental construction of an inner convex hull is considered. Applications of our method is described in Sect. 3.4.

3.1 Preliminaries

First, we give a formal definition of the problem. we define *possible convex hull*, *inner convex hull* and *outer convex hull* as follows.

Definition 3.1 (Possible Convex Hulls) *Possible Convex Hulls $CH(S)$ of a set S of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$ is defined as follows:*

$$CH(S) = \{conv\,(\cup_{i=1\ldots n}\{p_i\})\,|p_i \in \hat{p}_i (1 \leq i \leq n)\}$$

Since each possible set of points has the unique convex hull, definition 3.1 is a natural extension of definition 2.2.

Definition 3.2 (Inner Convex Hull) *The inner convex hull $R_{in}(S)$ of a set S of ε-Points is $\bigcap_{C \in CH(S)} C$, the intersection of all possible convex hulls.*

Definition 3.3 (Outer Convex Hull) *The outer convex hull $R_{out}(S)$ of a set S of ε-Points is $\bigcup_{C \in CH(S)} C$, the union of all possible convex hulls.*

In the rest of the chapter, we consider a problem of computing an inner convex hull and an outer convex hull for a given set S of ε-Points.

To state about geometric properties of an inner convex hull, we introduce more definitions. In the discussion below, we assume that the boundary of a polygon is represented by a sequence of directed edges such that the polygon lies in the left. Also, we assume that all lines are directed.

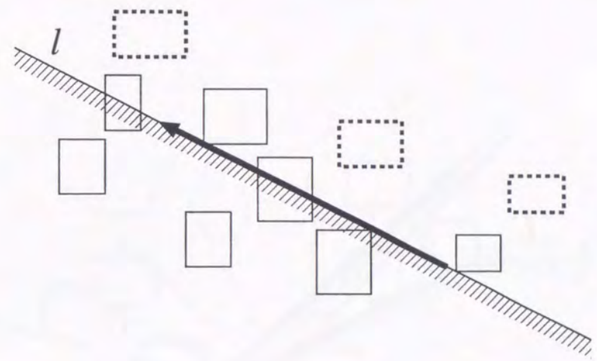

Figure 3.5: ε-Points covered by a directed line l

Definition 3.4 (Boundary Line) *We say "a line l is a boundary line of a convex polygon" if l is obtained by extending a boundary edge of the polygon.*

Definition 3.5 (Cover Relation between ε-Point and Line) *We say " a line l covers an ε-Point $\hat{p}$" if $\hat{p} \cap H_{\mathcal{L}}(l) \neq \emptyset$.*

In Fig. 3.5 the line l covers ε-Points drawn in solid line, and does not cover ε-Points drawn in dots. Since $H_{\mathcal{L}}(l)$ is a closed half-plane, ε-Points touching l are also covered.

Definition 3.6 (Cover Relation between ε-Points) *Let l_1 be $t_{C\mathcal{R}}(\hat{p}, \hat{p_1})$ and let l_2 be $t_{C\mathcal{R}}(\hat{p}, \hat{p}_2)$ for distinct ε-Points $\hat{p}, \hat{p_1} and \hat{p}_2$. We define "cover relation $<_{\hat{p}}, =_{\hat{p}}$" between ε-Points as follows:*

1. $\hat{p}_1 <_{\hat{p}} \hat{p}_2 \overset{\text{def}}{\Leftrightarrow} \hat{p}_2$ *is not covered by* l_1 *(see Fig.3.6).*
2. $\hat{p}_1 =_{\hat{p}} \hat{p}_2 \overset{\text{def}}{\Leftrightarrow} \hat{p}_2$ *is covered by* l_1 *and* $\hat{p}_1$ *is covered by* l_2.

Like $\hat{p}_1$ and $\hat{p}_2$ shown in Fig. 3.7, some pair of ε-Points does not satisfy the relation $<_{\hat{p}}$. In addition, note that the relation $\hat{p}_i =_{\hat{p}} \hat{p}_j$ arises only when the relation $<_{\hat{p}}$ can not be defined between them.

Definition 3.7 (Maximal ε-Point)

Let S be a set of ε-Points. For an ε-Point $\hat{p}$ in S, an ε-Point α in S is a maximal ε-Point in S if $\alpha <_{\hat{p}} \beta$ does not hold for any β in S except $\hat{p}$ and α.

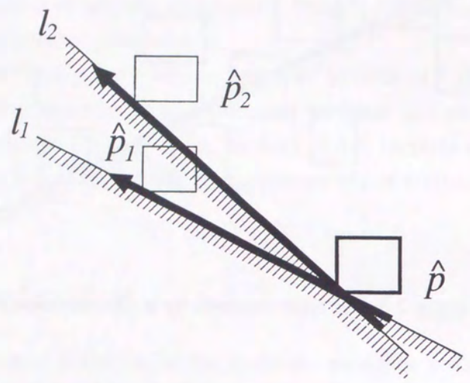

Figure 3.6: Case : $\hat{p_1} <_{\hat{p}} \hat{p_2}$

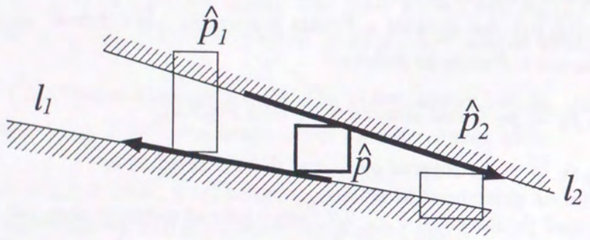

Figure 3.7: Case : relation $<_{\hat{p}}$ is not defined between $\hat{p_1}$ and $\hat{p_2}$

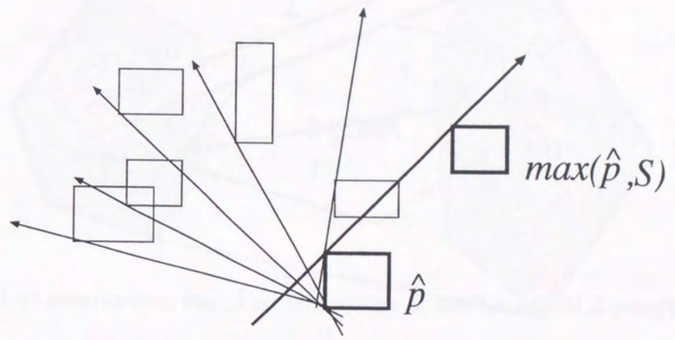

Figure 3.8: A maximal $max(\hat{p}, S)$ around $\hat{p}(\in S)$

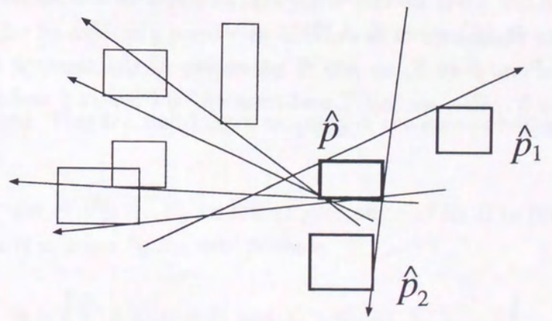

Figure 3.9: Case : No maximum around $\hat{p}(\in S)$ exist

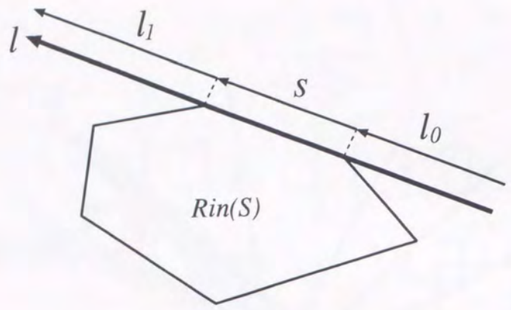

Figure 3.10: An ε-Point is pre-covered by l_1 and post-covered by l_0

As shown in Fig. 3.9, some ε-Points possibly does not have maximums around them. In addition, there may be more than one maximum around an ε-Point. In the case, the relation $=_{\hat{p}}$ holds between them from definition 3.7.

We distinguish a cover relation according to how a line covers an ε-Point.

Definition 3.8 (Pre-Cover/Post-Cover)

Let S be a set of ε-Points and let l be a boundary line of $R_{in}(S)$. Then, we divide l into three parts along its direction: "the part before l touches $R_{in}(S)$ (half line l_0 in fig.3.10)", "the part l touches $R_{in}(S)$ (segment s in fig.3.10)", "the part after l touches $R_{in}(S)$ (l_1 in fig.3.10)".

For an ε-Point $\hat{p}$ in S, we say "l pre-covers $\hat{p}$" if l covers $\hat{p}$ and intersects with $\hat{p}$ only in l_1. Also, we say "l post-covers $\hat{p}$" if l covers $\hat{p}$ and intersects with $\hat{p}$ only in l_0.

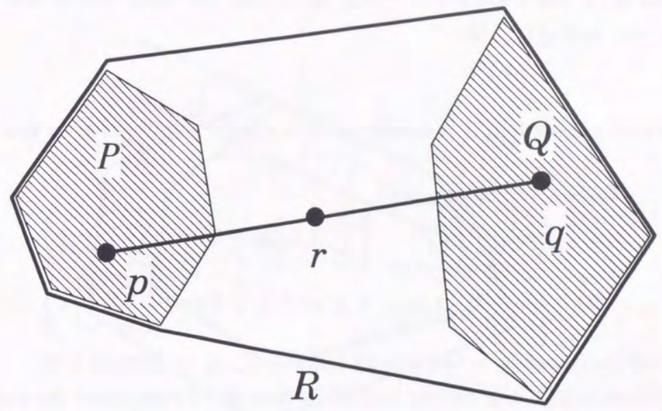

Figure 3.11: For a point r in $conv(P \cup Q)$, there are points $p \in P$ and $q \in Q$ s.t. $\overline{pq}$ passes through r

3.2 Outer Convex Hull

The important property of an outer convex hull is its convexity. To prove it, we use the following property of a convex hull.

Lemma 3.9 *Let P and Q be convex polygons and let R be the convex hull of $P \cup Q$. Then for an arbitrary point r in R, there is a segment $\overline{pq}$ s.t. $p \in P$ and $q \in Q$ and $r \in \overline{pq}$ (see the example in Fig. 3.11).*

Proof. Omitted. This is a well-known property of the convex hull of two convex bodies. □

Lemma 3.10 *Let $P_1, P_2, \ldots, P_n$ be convex polygons and let R be the convex hull of them. Then R is given by the next formula*

$$R = \left\{ \sum_{i=1}^{n} \lambda_i p_i \,\middle|\, p_i \in P_i \text{ and } \lambda_i \geq 0 \text{ and } \sum_{i=1}^{n} \lambda_i = 1 \right\}$$

Proof. Let P be the convex hull of $P_1 \cup \ldots \cup P_{n-1}$, and let Q be P_n, and let r be an arbitrary point in R. Then by inductively applying Lemma 3.9 for P, Q, and r, we get the lemma. □

By lemma 3.10, the convexity of an outer convex hull is easily shown as follows.

Theorem 3.11 *For a set S of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$, the outer convex hull $R_{out}(S)$ is the convex hull of $\bigcup_{i=1}^{n} \hat{p}_i$.*

Proof.

The outer convex hull is represented by a convex combination as follows:

$$\begin{aligned} R_{out}(S) &= \bigcup_{C \in CH(S)} C \\ &= \bigcup_{p_1 \in \hat{p}_1, \ldots, p_n \in \hat{p}_n} \left\{ \sum_{i=1}^{n} \lambda_i p_i \,\middle|\, \lambda_i \geq 0 \text{ and } \sum_{i=1}^{n} \lambda_i = 1 \right\} \\ &= \left\{ \sum_{i=1}^{n} \lambda_i p_i \,\middle|\, p_i \in \hat{p}_i \text{ and } \lambda_i \geq 0 \text{ and } \sum_{i=1}^{n} \lambda_i = 1 \right\} \end{aligned}$$

This set corresponds to the convex hull of $\bigcup_{i=1}^{n} \hat{p}_i$ by Lemma 3.10. □

By Theorem 3.11, the convex hull of vertices of ε-Points gives the outer convex hull. Since each ε-Point has only four vertices, a traditional convex hull algorithm[13] can construct the outer convex hull in $O(n \log n)$ time.

3.3 Inner Convex Hull

3.3.1 Geometric Properties

We characterize an inner convex hull by boundary lines of it.

Theorem 3.12 *Let l be a boundary line of $R_{in}(S)$. Then at least two ε-Points tangent to l exist in $H_{\mathcal{R}}(l)$.*

Proof. Assume that l is given by extending a boundary edge e of $R_{in}(S)$. Since e is a boundary edge of $R_{in}(S)$, a possible convex hull C exists that contains e in its boundary. Because each vertex of C is a possible point of S, l passes through at least two ε-Points. Now consider the set S_l of ε-Points that are passed through by l. We can conclude that l supports at least one ε-Point in S_l as follows.

If l does not support any ε-Points in S_l, l can be moved vertically to itself toward the left side in a sufficiently small distance ϵ without changing S_l (see Fig. 3.12). Since the moved line l_ϵ satisfies $S_{l_\epsilon} = S_l$, a possible convex hull C' exists that has a segment on l_ϵ as its boundary edge. Note that C' lies in $H_{\mathcal{L}}^{-}(l)$ because l_ϵ is contained in $H_{\mathcal{L}}^{-}(l)$. This implies that the boundary of C' passes through the inside of $R_{in}^{*}(S)$; this is a contradiction. Therefore l supports at least one

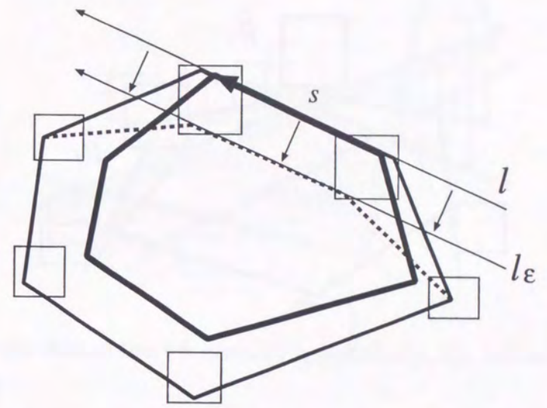

Figure 3.12: The line l can be moved to l_ϵ with the same set of intersecting ε-Points during the movement

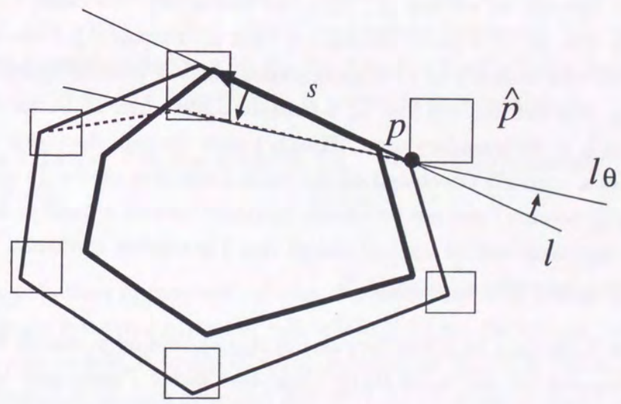

Figure 3.13: The line l can be rotated counterclockwise to l_θ with the same set of intersecting ε-Points during the movement

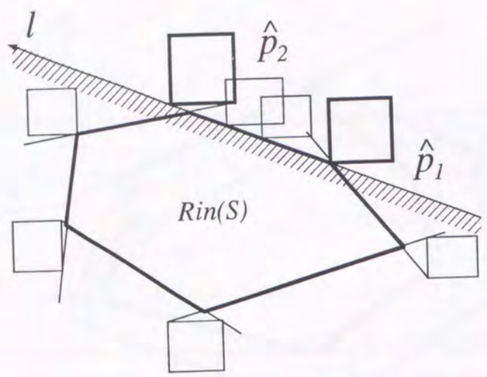

Figure 3.14: The boundary of $R_{in}(S)$ is determined by maximal ε-Points

ε-Point in S_l. In addition, ε-Points supported by l lie in $H_{\mathcal{R}}(l)$ because l could not be moved to the left without changing S_l.

Next, we show that l supports at least two ε-Points in S_l. Assume that l supports only one ε-Point in S_l. Let l support an ε-Point $\hat{p_0}$ at a point p and passes through an ε-Point $\hat{p_1}$. Here, we distinguish two cases: (1) l passes through $\hat{p_0}$ then $\hat{p_1}$, (2) l passes through $\hat{p_1}$ then $\hat{p_0}$. In case (1), l can be rotated counterclockwise around p in a sufficiently small angle θ without changing S_l (see Fig. 3.13). For this rotated line l_θ, a possible convex hull C' exists that has a segment on l_θ as its boundary edge. Since l_θ passes through the inside of $R_{in}(S)$, this implies a contradiction based on the same discussion above. In addition, $\hat{p_1}$ lies in $H_{\mathcal{R}}(l)$ because l can not be rotated counterclockwise around p. In case (2), the same argument can be applied except that l is rotated clockwise. Therefore l supports at least two ε-Points in S_l. □

Theorem 3.13 *Let l be a boundary line of $R_{in}(S)$, and let $\hat{p_1}$ and $\hat{p_2}$ be ε-Points that are tangent to l and lie in $H_{\mathcal{R}}(l)$ (Note that such ε-Points exist by Theorem 3.12). If l is directed from $\hat{p_1}$ to $\hat{p_2}$, then $\hat{p_2} = \max(\hat{p_1}, S)$ (see Fig.3.14).*

Proof.

Assume that $\hat{p_2}$ is not a $\max(\hat{p_1}, S)$. Then l does not cover at least one ε-Point $\hat{p}$ in S. For a point p in $\hat{p}$, let $C(p)$ be the convex hull of $\{p\} \cup R_{in}(S)$,

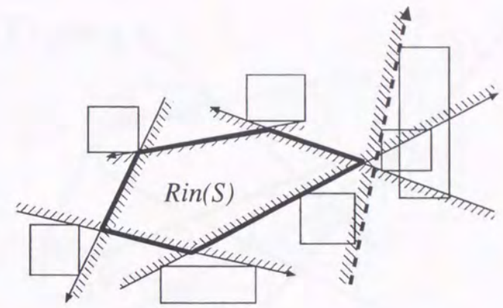

Figure 3.15: The dashed line l is given by $t_{C\mathcal{R}}(\hat{p}, \max(\hat{p}, S))$, but not a boundary line of $R_{in}(S)$

and $D(p)$ be the set of possible convex hulls: $\{C | C \in CH(S) \wedge\ p \in C\}$. Then all C in $D(p)$ satisfy $C(p) \subseteq C$ by the definition of $R_{in}(S)$ and the convexity of C. This implies $C(p) \subseteq \cap_{C \in D(p)} C$, and also $\cap_{p \in \hat{p}} C(p) \subseteq R_{in}(S)$ because $\cap_{p \in \hat{p}} C(p) \subseteq \cap_{p \in \hat{p}} \left\{ \cap_{C \in D(p)} C \right\} = \cap_{C \in CH(S)} C = R_{in}(S)$.

On the other hand, any possible point p of $\hat{p}$ lies in $H^-_{\mathcal{R}}(l)$ because $\hat{p}$ is in $H^-_{\mathcal{R}}(l)$. This implies that $H^-_{\mathcal{R}}(l)$ has a common region with all possible convex hulls of S, also with $\cap_{p \in \hat{p}} C(p)$. With the previous statement, this means that $R_{in}(S)$ has a common region with $H^-_{\mathcal{R}}(l)$. This is a contradiction because l is a boundary line of $R_{in}(S)$. Therefore $\hat{p_2}$ is proved to be a $\max(\hat{p_1}, S)$. □

Theorem 3.14 *Let l be $t_{C\mathcal{R}}(\hat{p}, \max(\hat{p}, S))$. Then $R_{in}(S) subseteq H_{\mathcal{L}}(l)$ holds. (see Fig. 3.15).*

Proof.

By the definition of maximal ε-Point, a possible set A of points that are all in $H_{\mathcal{L}}(l)$ exists because l covers all ε-Points in S. Thus the convex hull C of A, a possible convex hull, is in $H_{\mathcal{L}}(l)$; which means that $R_{in}(S)$ is also in $H_{\mathcal{L}}(l)$. □

By Theorem 3.13, each boundary line of $R_{in}(S)$ is given by $t_{C\mathcal{R}}(\hat{p}, \max(\hat{p}, S))$ for some ε-Point $\hat{p}$. However, observe that this kind of line is not always a boundary line of $R_{in}(S)$ (such a line is shown as a dashed line in Fig.3.15). Theorem 3.14 states that such redundant lines have no effect with the boundary of $R_{in}(S)$. Thus the boundary of $R_{in}(S)$ can be constructed by finding the line

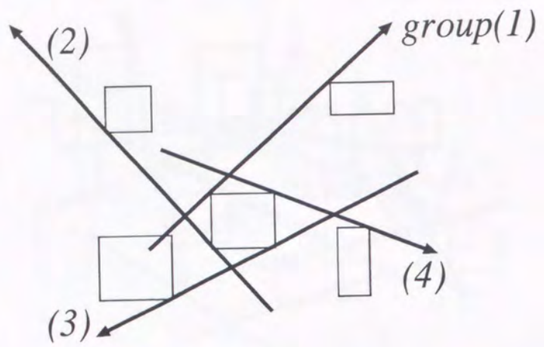

Figure 3.16: Four types of common tangent lines between axis-parallel rectangles

$t_{C\mathcal{R}}(\hat{p}_i, \max(\hat{p}_i, S))$ for each $\hat{p}_i$, then calculating the intersection of left half-planes of these lines.

3.3.2 Construction for a set of ε-Points

From the discussion above, the main problem is to find $\max(\hat{p}_i, S)$ for each $\hat{p}_i$. Instead of testing every pair of ε-Points, we focus on geometric properties of a common tangent line $t_{C\mathcal{R}}(\hat{p}_i, \max(\hat{p}_i, S))$.

Since all ε-Points are axis-parallel rectangles, common tangent lines can be classified into four groups based on their angle θ with respect to x-axis as follows: (1) lines whose angle satisfies $0 \leq \theta < \pi/2$, (2) $\pi/2 \leq \theta < \pi$, (3) $\pi \leq \theta < 3\pi/2$, (4) $3\pi/2 \leq \theta < 2\pi$. In each group, a common tangent line l supports two ε-Points in the same manner, that is, l in (1) is tangent to both at their upper-left vertices, (2) at lower-left vertices, (3) at lower-right vertices, (4) at upper-right vertices (see Fig.3.16).

Now consider finding all the lines in group (1). Since they are tangent to two ε-Points at their upper-left vertices, the convex hull C of all upper-left vertices of ε-Points can be used to find them. Observe that only the boundary lines of C can belong to group (1) because each boundary line l of C covers all ε-Points except two ε-Points $\hat{p}$ and $\hat{q}$ whose upper-left vertices l passes through (see Fig. 3.17). The rest of work is to test if $\hat{p}$ and $\hat{q}$ are in $H_{\mathcal{R}}(l)$, which means that l corresponds to $t_{C\mathcal{R}}(\hat{p}, \hat{q})$. For example, in Fig. 3.17 only the boundary lines shown in solid

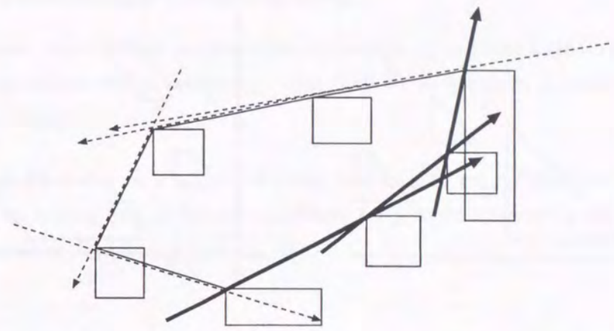

Figure 3.17: The convex hull of upper-left vertices and its boundary lines

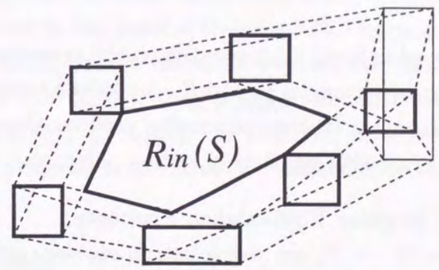

Figure 3.18: The inner convex consists of four convex hulls

are in group (1). As each ε-Point has only four vertices, this test can be done in constant time for a boundary line of C. Hence, we can find all the lines in group (1) by calculating a convex hull and scanning its boundary. The similar method can be used for the other three groups.

Consequently, we can find all maximal ε-Points by calculating four convex hulls. Moreover, note that $R_{in}(S)$ is the intersection of these four convex hulls. In Fig. 3.18, for example, the inner convex hull is given by the intersection of four dashed convex hulls. Clearly, all these process can be done in $O(n \log n)$ time for n ε-Points.

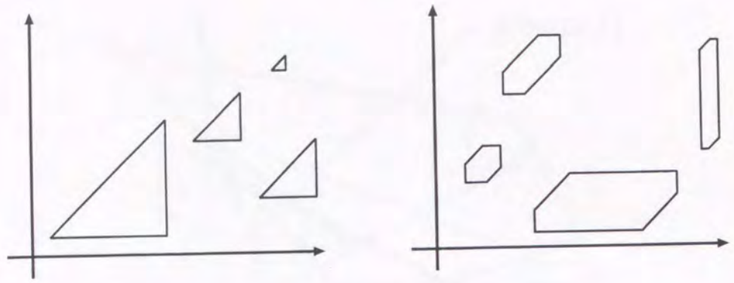

Figure 3.19: Examples of angular equivalent polygons

3.3.3 Construction for a set of Angular Equivalent Polygons

The algorithm showed in Sect. 3.3.2 depends on the property of a common tangent line of axis-parallel rectangles R_1 and R_2: a common tangent line $t_{C\mathcal{R}}(R_1, R_2)$ supports both rectangles at their corresponding vertices. Observe that this property arises from the following property of convex polygons.

Definition 3.15 (Angular Equivalence Property)
Convex polygons $P_1, \cdots, P_n$ are said to satisfy angular equivalence if they have the same sequence of interior angles and their corresponding edges are parallel.

In the axis-parallel rectangle case, the sequence of interior angle is $90° - 90° - 90° - 90°$, and their corresponding edges are parallel. In Fig. 3.19, the left example shows angular equivalent polygons whose sequence of interior angle is $45° - 90° - 45°$, and the right example shows angular equivalent polygons whose sequence of interior angle is $90° - 135° - 135° - 90° - 135° - 135°$.

By a similar discussion in Sect. 3.3.2, the common tangents of angular equivalent polygons can be classified into as many groups as the number of vertices of a polygon. As in the axis-parallel rectangle case, lines in each group can be found by calculating the convex hull of the corresponding vertices of ε-Points.

When angular equivalent n polygons with k vertices are given as ε-Points, the inner convex hull can be constructed by calculating k convex hulls and making the intersection of them. Hence the inner convex hull is calculated in $O(kn \log n)$ time.

3.3.4 Incremental Construction

In this section, we consider an incremental construction of an inner convex hull. First, we introduce useful properties that lead to an efficient incremental construction of $R_{in}(S)$.

Theorem 3.16 *Let S be a set of ε-Points, and let $\hat{p}$ be an ε-Point not contained in S. The necessary and sufficient condition for $\hat{p}$ to be covered by all boundary lines of $R_{in}(S)$ is $R_{in}(S \cup \{\hat{p}\}) = R_{in}(S)$.*

Proof.

1. Proof of necessity

   Let l be a boundary line of $R_{in}(S)$. Because $\hat{p}$ is covered by all boundary lines of $R_{in}(S)$, $\hat{p}$ and all other elements of S are covered by l. Hence, by a similar discussion in the proof of theorem 3.13, $\cap_{C \in CH(S \cup \{\hat{p}\})} C$ is proved to be contained in $H_{\mathcal{L}}(l)$. In the same way, we can conclude that for any boundary line l of $R_{in}(S)$, $\cap_{C \in CH(S \cup \{\hat{p}\})} C$ is contained in $H_{\mathcal{L}}(l)$. This implies $R_{in}(S \cup \{\hat{p}\}) \subseteq R_{in}(S)$. In addition, since $S \subset S \cup \{\hat{p}\}$ holds, $R_{in}(S \cup \{\hat{p}\}) \supseteq R_{in}(S)$ is also true. We can conclude that $R_{in}(S \cup \{\hat{p}\}) = R_{in}(S)$.

2. Proof of sufficiency

   Assume that there is a boundary line l of $R_{in}(S)$ that does not cover $\hat{p}$. This means that $\hat{p} \subseteq H_{\mathcal{R}}^{-}(l)$ holds. Now, consider a convex hull C_p determined by the vertices of $R_{in}(S)$ and a point p in $\hat{p}$. Since $\cap_{p \in \hat{p}} C_p = R_{in}(S \cup \{\hat{p}\})$ holds, we can conclude that $R_{in}(S) \subset R_{in}(S \cup \{\hat{p}\})$ holds by a similar discussion in the proof of theorem 3.13. However, it contradicts to $R_{in}(S \cup \{\hat{p}\}) = R_{in}(S)$. Hence, $\hat{p}$ is covered by all lines which construct $R_{in}(S)$.

□

Theorem 3.17 *Let S be a set of ε-Points, and let l be a left tangent of $R_{in}(S)$. Consider a point v at which l touches $R_{in}(S)$, edges e_1 and e_2 incident to v, and lines l_1 and l_2 given by extending e_1 and e_2 respectively. Then ε-Points that are pre-covered by l_1 and post-covered by l_2 are only the ones that are not covered by l (see. fig. 3.20).*

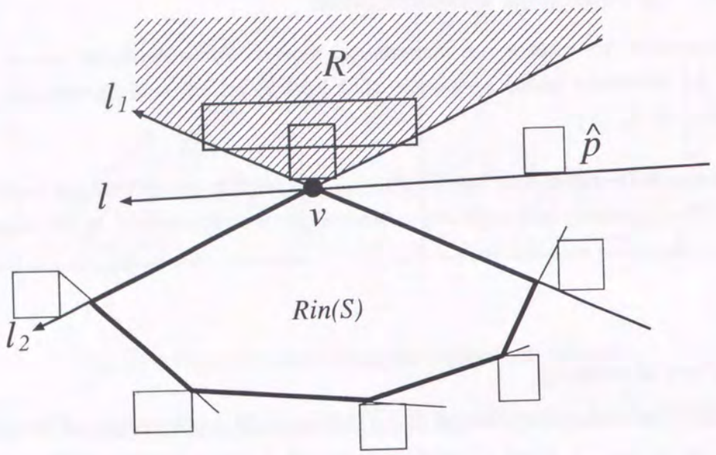

Figure 3.20: Maximums around $\hat{p}$ exist in the shaded region

Proof.

Among ε-Points in S, only the ones in $H_{\mathcal{R}}^{-}(l)$ are not covered by l. Since l is a left tangent of $R_{in}(S)$, if we move along l according to its direction, l passes through the intersection of $H_{\mathcal{R}}(l_1)$ and $H_{\mathcal{L}}(l_2)$ first, then v, and finally the intersection of $H_{\mathcal{L}}(l_1)$ and $H_{\mathcal{R}}(l_2)$. This means that ε-Points in $H_{\mathcal{R}}^{-}(l)$ are pre-covered by l_1 and post-covered by l_2. □

By applying theorem 3.17 to the line $l = t_{\mathcal{RL}}(\hat{p}, R_{in}(S))$, $max(\hat{p}, S \cup \{\hat{p}\})$ is proved to exist in $H_{\mathcal{R}}(l)$; a common tangent between $R_{in}(S)$ and a newly added ε-Point $\hat{p}$ can be used to select the candidates of maximums around $\hat{p}$ in an incremental process. Also, by theorem 3.16 and theorem 3.17, once an ε-Point is proved to be covered by all lines that construct $R_{in}(S)$ and not pre-covered around any vertices of $R_{in}(S)$, the ε-Point can be excluded from the process without any effect on the final shape of the inner convex hull.

Theorem 3.18 *For a boundary line l of $R_{in}(S)$, there is at least one ε-Point in S that is in $H_{\mathcal{R}}(l)$ and pre-covered by l, and at least one that is in $H_{\mathcal{R}}(l)$ and post-covered by l.*

Proof.

By theorem 3.16, we assume without loss of generality that all ε-Points in S does not intersect with $R_{in}(S)$.

Let S_l be a set of ε-Points that theorem 3.12 guarantees to exist. And as in definition 3.6, let l be divided along its direction into l_0,s, and l_1 at which l starts to touch $R_{in}(S)$ and stops to touch it. Assume that any elements in S_l are not pre-covered by l. Then, S_l has a possible arrangement in which all points are on l_0. This means that there is a possible convex hull for S that does not contain the edge s in it, which contradicts to the definition of $R_{in}(S)$. Hence, at least one element in S_l must be post-covered by l. Similarly, at least one element in S_l must be pre-covered by l. From the discussion above, the elements described in this proposition are proved to exist in S. □

Theorem 3.19 *Let l_0 and l_1 be boundary lines of $R_{in}(S)$ that are adjacent at a vertex v of $R_{in}(S)$. Then, at least one ε-Point in S is pre-covered by l_0 and post-covered by l_1.*

Proof.

By theorem 3.18, there exists at least one ε-Point $\hat{p}$ which is in $R(l_0)$ and pre-covered by l_0. From theorem 3.14, $\hat{p}$ is also covered by l_1 because all elements of S are covered by all boundary lines of $R_{in}(S)$. Notice that since $\hat{p}$ satisfies $\hat{p} \subseteq R(l_0)$, $\hat{p}$ is post-covered by l_1. Hence, the element of S described in the proposition is proved to exist. □

Remember we mentioned at theorem 3.17 that we can select the candidates of maximums for a newly added $\hat{p}$. Theorem3.19 guarantees at least one candidate exists.

In the following, we introduce an incremental construction of inner convex hull. In the construction, a non-empty inner convex hull is constructed at first. Then, for a newly added ε-Point, we find all the maximums and update the boundary of the processed inner convex hull.

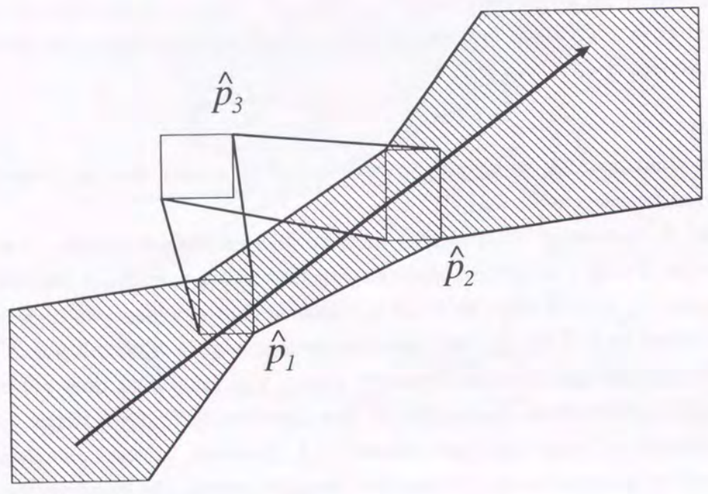

Figure 3.21: Case : Inner convex hull is not empty

Initial inner convex hull construction

A non-empty inner convex hull is needed prior to the incremental step. It suffices to construct an inner convex hull determined by three ε-Points $\hat{p_1}, \hat{p_2}$ and $\hat{p_3}$. Notice that as shown in Fig.3.21 and Fig.3.22, the inner convex hull determined by three ε-Points is not always non-empty. As we described in [24], the emptiness can be tested in constant time by determining the relative location of $\hat{p_3}$ and the trace of all stabbing lines of $\hat{p_1}$ and $\hat{p_2}$.

By applying this emptiness test for each triples, a non-empty inner convex hull is calculated in $O(1)$ time (best case), or in $O(n^3)$ time (worst case).

Find new maximums

Consider to find all maximums $\max(\hat{p}, S)$ from a set S of n ε-Points. Since not all pairs of ε-Points satisfy the relation $<_{\hat{p}}$, we can not find all maximums by a simple sorting. However, observe that if common tangents $t_{\mathcal{CR}}(\hat{p}, \hat{p}_i)$ and $t_{\mathcal{CR}}(\hat{p}, \hat{p}_j)$ touch the same vertex v of $\hat{p}$, either $\hat{p}_i <_{\hat{p}} \hat{p}_j$ or $\hat{p}_j <_{\hat{p}} \hat{p}_i$ always holds according to their angle around v (see Fig. 3.6).

For a set of angular equivalent polygons with k vertices, they can be classified

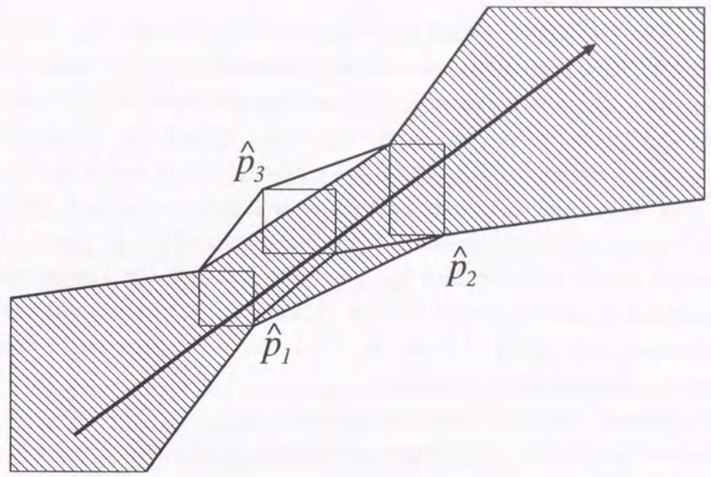

Figure 3.22: Case : Inner convex hull is empty

into k groups around $\hat{p}$ based on the same idea in Sect.3.3.2. By the observation above, we can find local maximums in each group by comparing each ε-Points around p in $O(n)$ time. By examining all pairs of these $O(k)$ local maximums we can find maximums with additional $O(k^2)$ time.

Incremental construction of an inner convex hull

The incremental construction algorithm is shown in Fig. 3.23. Here, $S_k(|S_k| = k)$ denotes a set of already processed ε-Points, and R denotes the inner convex hull of S_k. By theorem 3.19, each vertex of $R_{in}(S_k)$ has its associated ε-Points. We assume that each vertex of R keeps its associated ε-Points in a list.

Consider the time complexity in a incremental step. By theorem 3.19, each vertex of $R_{in}(S_k)$ has its associated ε-Points in S_k. Hence, $R_{in}(S_k)$ has at most k vertices. Since $R_{in}(S_k)$ is convex, the common tangents $l = t_{\mathcal{RL}}(\hat{p}, R)$, and $l' = t_{\mathcal{LR}}(\hat{p}, R)$ can be found in $O(\log k)$ time[31] at step 2-1 (see Fig. 3.24). In a case that no common tangent is found, the rest of the update process can be skipped because $R_{in}(S \cup \{\hat{p}\}) = R_{in}(S)$ holds by theorem 3.16. Let l and l' touch vertices v_0 and v_1 of R respectively, and let S_l and S'_l be the sets of associated

ε-Points around v_0 and v_1.

Next, new maximums around $\hat{p}$ are found at step 2-3 (see Fig. 3.25). By theorem 3.17, new maximums are only the elements in S_l or S'_l. Since each set contains from one up to k elements, all maximums are found in $O(1)$ time (best case) or $O(k)$ time (worst case) by the algorithm in Sec.3.3.4. Those ε-Points that are turned out not to be maximal are removed from S_l and S'_l at this step.

At step 2-4 a common tangent between $\hat{p}$ and a new maximum is calculated in $O(1)$ time to update the boundary of the inner convex hull. At step 2-5 by an incremental convex hull algorithm for a set of points [30], the boundary of the inner convex hull can be updated in $O(\log k)$ time since at most constant number of vertices are newly added to R (see fig. 3.26). The set of associated ε-Points of affected vertices of R are also updated.

For summary, one incremental step takes $O(\log k)$ time (best case), or $O(k)$ time (worst case); which leads to the overall time complexity of $O(n \log n)$ (best case), or $O(n^2)$ (worst case). It is easy to show that the algorithm shows its best behavior for an input in which no three points are on the same line among all possible sets of points.

$S = \{\hat{p_1}, \hat{p_2}, \ldots, \hat{p_n}\}, V = \emptyset$
Step 1: construct a non-empty inner convex hull
$\hat{p_1} = S.pop, \hat{p_2} = S.pop, \hat{p_3} = S.pop$
$S_3 = \{\hat{p_1}, \hat{p_2}, \hat{p_3}\}$,$R = R_{in}(S_3)$
Step 2:construct $R_{in}(S_k \cup \{\hat{p}\})$ from $R_{in}(S_k)$
while $S \neq \emptyset$
$\hat{p} = S.pop$
Step 2-1:calculate common tangents between $\hat{p}$ and R(fig.3.24)
l = [$\hat{p}$ findTangentTo:R type:"RL"]
l' = [$\hat{p}$ findTangentTo:R type:"LR"]
if ((l == NULL) $\vee$ (l' == NULL)) goto Step2-6
Step 2-2: select candidates for maximums
Let l touches a vertex v of R, and l' touches a vertex v' of R.
S_l = { associated ε-Points of v }
$S_{l'}$ = { associated ε-Points of v' }
Step 2-3:find maximums (fig. 3.25)
find max_1 s.t. $max_1 = (\hat{p}, S_l)$
find max_2 s.t. $\hat{p} = (max_2, S_{l'})$
Step 2-4:calculate common tangents between $\hat{p}$ and maximums
l_1 = [$\hat{p}$ findTangentTo:max_1 type:"RR"]
l_2 = [max_2 findTangentTo:$\hat{p}$ type:"RR"]
Step 2-5: update R (fig. 3.26)
R = updateHull(R,l,l',l_1,l_2)
Step 2-6:
$S_{k+1} = S_k \cup \{\hat{p}\}$

Figure 3.23: Construction of an inner convex hull

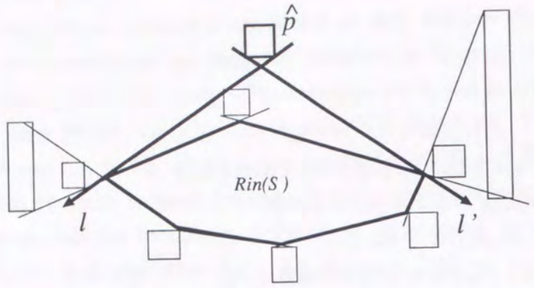

Figure 3.24: Common tangent lines from $\hat{p}$ to $R_{in}(S)$

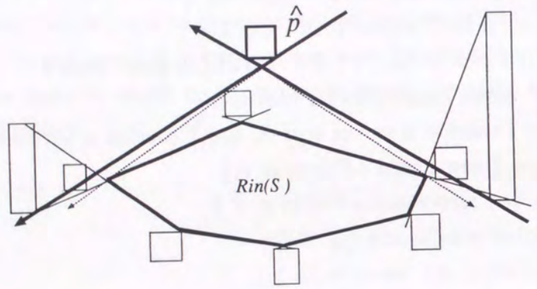

Figure 3.25: Search for maximums

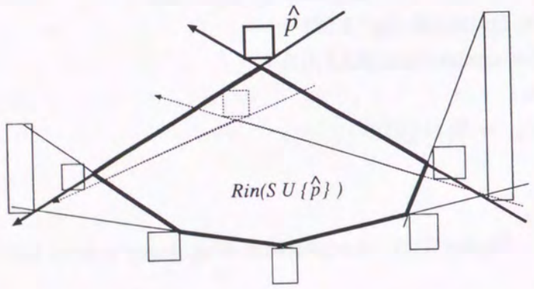

Figure 3.26: Reconstruction of the boundary of $R_{in}(S)$

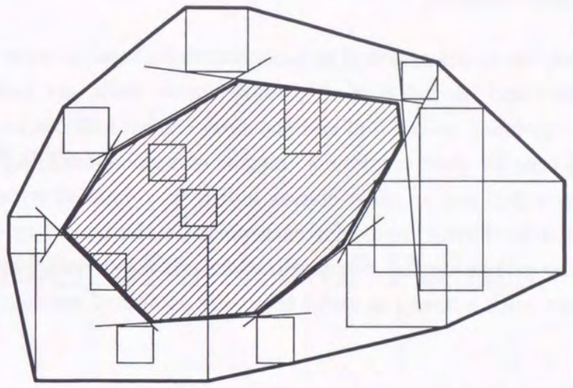

Figure 3.27: Output example

3.4 Applications

We show how inner/outer convex hull can be used to calculate reliable results. One application is calculation of the diameter of a set P of points. The diameter of P is given by that of the convex hull of P. With the existence of input error, the diameter of the inner/outer convex hull gives a good lower/upper bound. See the example in Fig. 3.27. Observe that the inner convex hull is not affected by the lower-left ε-Point that has a large error although the outer convex hull is affected by the ε-Point. In our method, the effect of possible points that does not contribute to possible convex hulls is removed.

Another application is to find stable extreme points for a set of points. Some extreme points may be sensitive to input errors, and need to be avoided to make a strongly convex hull. For a given set S of ε-Points, $\hat{p}_i$ gives an extreme point in all possible sets of points if and only if $\hat{p}_i \wedge R_{out}(S \setminus \{\hat{p}_i\}) = \emptyset$. For each vertex in a possible convex hull, its convexity can be verified in this way, and we can construct a strongly convex hull by choosing only stable extreme points.

3.5 Conclusion

In this chapter, the computation of an inner convex hull and an outer convex hull, the intersection and the union of all possible convex hulls, are considered. We showed that the inner convex hull and the outer convex hull can be constructed in $O(n \log n)$ time for given n ε-Points. Instead of a global error bound, a pair of an inner convex hull and an outer convex hull gives a detailed error bound. We can compute a local error bound for each point on the boundary of a possible convex hull by measuring the distance to them. If input data contains a wide range of errors, such a bound is useful to qualify computed results.

Chapter 4

Diameter of a Set of Points

In this chapter, we address the problem of calculating the maximum and the minimum of possible diameters of accuracy guaranteed points. The diameter of a set of points is the maximum distance between any two points. It is well known that the diameter of a set of points is equal to the diameter of its convex hull and once we have the convex hull, its diameter is determined in linear time by traversing all antipodal pairs of it[31]. Hence, the diameter of n points can be calculated in $O(n \log n)$time.

If a set of points is given by ε-Points, its diameter ranges over the diameters of all possible convex hulls. Apparently, the inner and the outer convex hull of the set gives an upper and a lower bound of possible diameters. However, in some configuration, the bound becomes meaningless. In Fig. 4.1, the lower bound is underestimated since vertices of the inner convex hull are apart from ε-Points. Also, the upper bound is overestimated due to the large ε-Point. The minimum and the maximum of possible diameters for the ε-Points is shown in Fig. 4.2.

If quadratic computational time is allowed, we can calculate, for each pair of ε-Points, the minimum and the maximum of possible distances between them.

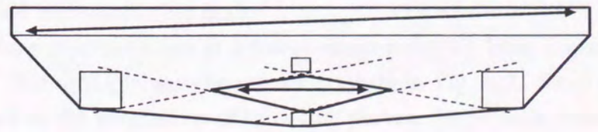

Figure 4.1: An upper bound and a lower bound of possible diameters

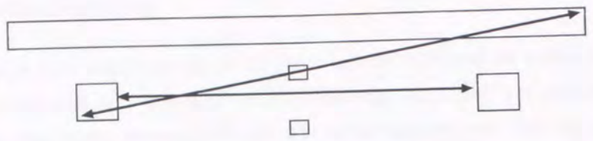

Figure 4.2: The maximum and the minimum of possible diameters

Then, we find the maximum of these maximums and find the maximum of these minimums. It is easy to show that these two maximums give the maximum and the minimum of possible diameters. The problem is to determine which pair of possible points gives the maximum or the minimum of possible diameters in sub-quadratic time.

We introduce the notion of *half-plane representation* of a convex region, and show how to use it to compute the maximum and the minimum of possible diameters of points. We show that for given n ε-Points, the minimum of possible diameters can be determined in $O(n \log n)$ time, and the maximum can be determined in $O(n \log^* n \log n)$ expected time. Also, we show that half-plane representation relates the inner/outer convex hull and the minimum/maximum of possible diameters of a set of ε-Points.

In Sect. 4.1, we give the definition of half-plane representation and show half-plane representations of an inner and an outer convex hull. In Sect. 4.2, we show how the minimum and the maximum of possible diameters can be efficiently calculated with the aid of the half-plane representations.

4.1 Half-Plane Representation

In this section, we introduce *half-plane representation* of a convex region in the plane. The representation relates a convex region to a function that represents a set of half-planes. We give a preliminary lemma that the intersection and the convex hull of two convex regions are represented by simple operations on half-plane representation.

First, we introduce *angular representation* of a line. The representation relates a function to a set of directed lines.

Definition 4.1 (Angular Representation of a Directed Line) *For a real-valued function $d(\theta)$, $l(d,\omega)$ denotes a directed line: $x \sin\omega - y\cos\omega = d(\omega)$.*

Our main idea is that once the range of θ is specified, a real-valued function $d(\theta)$ can be identified with a convex region $\cap\{H_{\mathcal{L}}(l(d,\theta))\}$. Based on the idea, we introduce *half-plane representation* as follows.

Definition 4.2 (Half-Plane Representation of a Convex Region) *Let R be a convex region in the plane. A real-valued function $d(\theta)$ is a half-plane representation of R if it satisfies*

$$\begin{cases} d(\theta + 2\pi) = d(\theta) \\ R = \cap_{0 \le \theta < 2\pi} H_{\mathcal{L}}(l(d,\theta)) \end{cases}$$

We denote $d(\theta) \equiv d'(\theta)$ if both $d(\theta)$ and $d'(\theta)$ are half-plane representations of the same convex region.

Notice that any convex region R in the plane has a half-plane representation of it because there is at most one left tangent $t_{\mathcal{L}}(R,\theta)$ for any θ (if R is bounded, there is exactly one left tangent). In the following, we assume that all half-plane representations are total functions on $\Re$ with a period 2π and their domains are not specified without further notice.

Half-plane representation is a one-to-many mapping from convex regions to functions. For example, see the regular triangle in Fig. 4.3. Since a triangle is represented by the intersection of three half-planes, the periodic discrete function in Fig. 4.4 is a half-plane representation of the triangle. On the other hand, the triangle has infinitely many left tangents shown as dashed lines in Fig. 4.3. Thus the periodic continuous function shown in Fig. 4.5 is also a half-plane

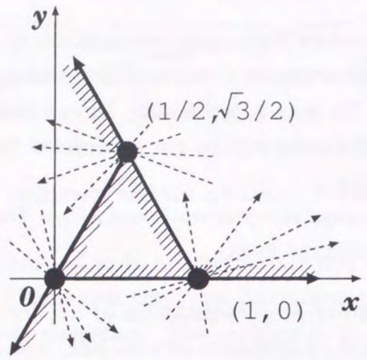

Figure 4.3: A regular triangle given by the intersection of half-planes

representation of the triangle. Moreover, redundant half-planes may be contained in a half-plane representation. Not all the half-planes represented by a half-plane representation of a region R need to share a point with the boundary of R.

We introduce a *normalized half-plane representation* that has a direct relationship to the corresponding convex region.

Definition 4.3 (Normalized Half-Plane Representation of a Convex Region) *Let $d(\theta)$ be a half-plane representation of a convex region R in the plane. If $l(d,\theta)$ equals to $t_{\mathcal{L}}(R,\theta)$ for any θ, $d(\theta)$ is called the normalized half-plane representation of R. The normalized half-plane representation of R is denoted as $d^*(R,\theta)$.*

Notice that each convex region has its unique normalized half-plane representation because there is at most one left tangent for any angle θ. For example, the function in Fig.4.5 is the normalized half-plane representation of the triangle in Fig. 4.3, while the function in Fig. 4.4 is not the one. Also, a normalized half-plane representation has a property that it represents left and right tangents at the same time.

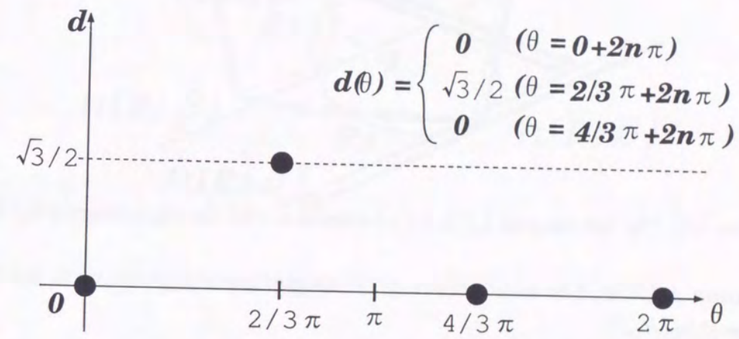

Figure 4.4: A half-plane representation of a triangle by a discrete function

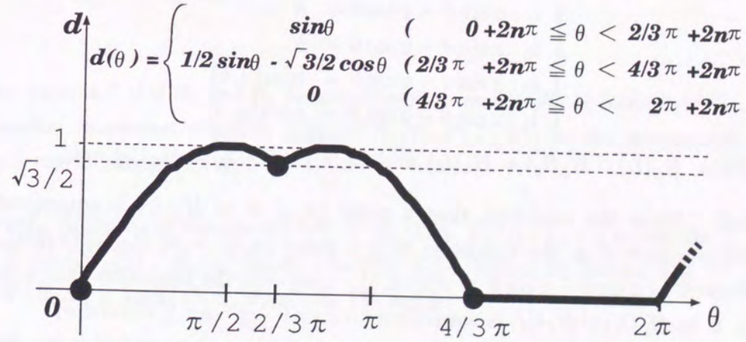

Figure 4.5: A half-plane representation of a triangle by a continuous function

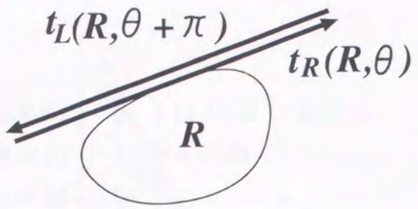

Figure 4.6: The left tangent $t_{\mathcal{L}}(R,\theta+\pi)$ coincides with the right tangent $t_{\mathcal{R}}(R,\theta)$

Lemma 4.4 *For $d(\theta) = -d^*(R,\theta+\pi)$, a directed line $l(d,\omega)$ equals to the right tangent $t_{\mathcal{R}}(R,\omega)$*

Proof. The left tangent $t_{\mathcal{L}}(R,\theta+\pi)$, which is $l(d^*,\theta+\pi)$, occupies the same location as the right tangent $t_{\mathcal{R}}(R,\theta)$, and has the opposite direction to $t_{\mathcal{R}}(R,\theta)$ (see Fig. 4.6). □

The advantage of half-plane representation is that geometric operations on convex regions can be reduced to operations on half-plane representations. We give the reduction in the following lemma 4.5, theorem 4.6 and theorem 4.7.

Lemma 4.5 *Let l_1, l_2, l_3 and l_4 be directed lines as follows*

$$\begin{cases} l_1 : x\sin\theta - y\cos\theta = & a \\ l_2 : x\sin\theta - y\cos\theta = & b \\ l_3 : x\sin\theta - y\cos\theta = & \min\{a,b\} \\ l_4 : x\sin\theta - y\cos\theta = & \max\{a,b\} \end{cases}$$

Then, $H_{\mathcal{L}}(l_1) \cap H_{\mathcal{L}}(l_2) = H_{\mathcal{L}}(l_3)$ and $H_{\mathcal{L}}(l_1) \cup H_{\mathcal{L}}(l_2) = H_{\mathcal{L}}(l_4)$ hold.

Proof. Since the condition that a point (x,y) is in $H_{\mathcal{L}}(l_1)$ is equivalent to $x\sin\theta - y\cos\theta \le a$, the condition that a point (x,y) is in $H_{\mathcal{L}}(l_1) \cap H_{\mathcal{L}}(l_2)$ is equivalent to $x\sin\theta - y\cos\theta \le \min\{a,b\}$. Similarly, the condition that a point (x,y) is in $H_{\mathcal{L}}(l_1) \cup H_{\mathcal{L}}(l_2)$ is equivalent to $x\sin\theta - y\cos\theta \le \max\{a,b\}$. □

By lemma 4.5,the following theorems on half plane representation are established.

Theorem 4.6 *Let $d_1(\theta)$ be a half-plane representation of R_1 and let $d_2(\theta)$ be a half-plane representation of R_2. Then, the piecewise minimum function*
$\min\{d_1(\theta), d_2(\theta)\}$ *is a half-plane representation of $R_1 \cap R_2$.*

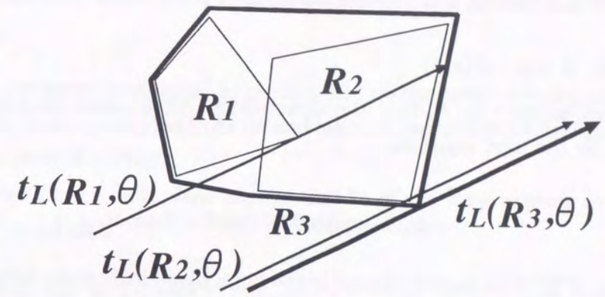

Figure 4.7: A left tangent of the convex hull of $R_1 \cup R_2$

Proof.

$$\begin{aligned} & R_1 \cap R_2 \\ = & \left(\bigcap_{0\le\theta<2\pi} H_{\mathcal{L}}(l(d_1,\theta))\right) \bigcap \left(\bigcap_{0\le\theta<2\pi} H_{\mathcal{L}}(l(d_2,\theta))\right) \\ = & \bigcap_{0\le\theta<2\pi} \left(H_{\mathcal{L}}(l(d_1,\theta)) \bigcap H_{\mathcal{L}}(l(d_2,\theta))\right) \\ & < \textit{By lemma } 4.5 > \\ = & \bigcap_{0\le\theta<2\pi} (H_{\mathcal{L}}(l(\min\{d_1,d_2\},\theta))) \end{aligned}$$

□

Theorem 4.7 *Let R_1 and R_2 be convex regions and let R_3 be $conv(R_1 \cup R_2)$. The piecewise maximum function $\max\{d^*(R_1,\theta), d^*(R_2,\theta)\}$ is the normalized half-plane representation of R_3*

Proof. Since R_3 is the convex hull of $R_1 \cup R_2$, $H_{\mathcal{L}}(t_{\mathcal{L}}(R_3,\theta))$ equals to $H_{\mathcal{L}}(t_{\mathcal{L}}(R_1,\theta)) \cup H_{\mathcal{L}}(t_{\mathcal{L}}(R_2,\theta))$ (see Fig. 4.7). Hence, by lemma 4.5, $d^*(R_3,\theta) = \max\{d^*(R_1,\theta), d^*(R_2,\theta)\}$ holds. □

We show examples of half-plane representation.

Example 1. A point $P(a,b)$

The left tangent $t_{\mathcal{L}}(P,\theta)$ is the directed line that passes through P, and given by the next equation

$$x\sin\theta - y\cos\theta = a\sin\theta - b\cos\theta$$

Hence, $d^*(P,\theta) = a\sin\theta - b\cos\theta$ is the normalized half-plane representation of P.

Example 2. The convex hull of a set of points

Let P be a set of points $p_1,\ldots,p_n$. By theorem 4.7, the normalized half-plane representation of $conv(P)$ is given as follows

$$d^*(conv(P),\theta) = \max_{1\le i\le n}\{d^*(p_i,\theta)\}$$

In more detail, let (a_i,b_i) be the coordinates of p_i; $d^*(conv(P),\theta)$ is re-expressed as

$$d^*(conv(P),\theta) = \max_{1\le i\le n}\{a_i\sin\theta - b_i\cos\theta\}$$

Observe that the normalized half-plane representation in Fig. 4.5 satisfies the above equation.

Example 3. An ε-Point $\hat{p} = (x,y,\epsilon_x,\epsilon_y)$

Since an ε-Point is an axis-parallel rectangle with four vertices $(x-\epsilon_x, y-\epsilon_y)$, $(x-\epsilon_x, y+\epsilon_y)$, $(x+\epsilon_x, y-\epsilon_y)$ and $(x+\epsilon_x, y+\epsilon_y)$, the normalized half-plane representation is given as follows

$$d^*(\hat{p},\theta) = \max\left\{\begin{array}{l}(x-\epsilon_x)\sin\theta - (y-\epsilon_y)\cos\theta,\\ (x-\epsilon_x)\sin\theta - (y+\epsilon_y)\cos\theta,\\ (x+\epsilon_x)\sin\theta - (y-\epsilon_y)\cos\theta,\\ (x+\epsilon_x)\sin\theta - (y+\epsilon_y)\cos\theta\end{array}\right\}$$

4.1.1 Half-Plane Representation of an Inner Convex Hull

Based on theorems introduced in section 4.1, we construct a half-plane representation of an inner convex hull. As proved later, it consists of normalized half-plane representations of ε-Points.

First, we show a half-plane representation of the inner convex hull for a singleton set of ε-Point.

Lemma 4.8 *For an ε-Point $\hat{p}$, $\min_{p\in\hat{p}}\{d^*(p,\theta)\} = -d^*(\hat{p},\theta+\pi)$ holds.*

Proof. By lemma 4.5, $\min_{p\in\hat{p}}\{d^*(p,\theta)\}$ is the normalized half-plane representation of $H_{\mathcal{L}}(t_{\mathcal{R}}(\hat{p},\theta))$. Hence, by lemma 4.4, the proposition holds. □

Since $R_{in}(\{\hat{p}\}) = \cap_{p\in\hat{p}}p$, by theorem 4.6, $d(R_{in}(\{\hat{p}\}),\theta) \equiv \min_{p\in\hat{p}}\{d^*(p,\theta)\}$ holds; lemma 4.8 gives a half-plane representation of $R_{in}(\{\hat{p}\})$.

Next, we show a half-plane representation of the inner convex hull for a set of ε-Points.

Theorem 4.9 *Let S be a set of ε-Points $\hat{p}_1,\ldots,\hat{p}_n$. $-\min_{1\le i\le n}\{d^*(\hat{p}_i,\theta+\pi)\}$ is a half-plane representation of $R_{in}(S)$.*

Proof.

From the definition of inner convex hull, $R_{in}(S)$ is expressed by ε-Points as follows

$$\begin{aligned} R_{in}(S) &= \bigcap_{C\in CH(S)} C = \bigcap_{A\in A(S)} conv(A) = \bigcap_{A\in A(S)} conv\left(\bigcup_{i=1}^{n} f_A(\hat{p}_i)\right)\\ &= \bigcap_{p_i\in\hat{p}_i} conv\left(\bigcup_{i=1}^{n}\{p_i\}\right)\end{aligned}$$

Hence, the following function represented by the normalized half-plane representations of points is a half-plane representation of $R_{in}(S)$.

$$d(\theta) = \min_{p_i\in\hat{p}_i}\{\max_{1\le i\le n}\{d^*(p_i,\theta)\}\}$$

Since each point p_i is chosen independently from each ε-Point, by lemma 4.10, the above expression is modified as follows

$$\begin{aligned} d(\theta) &= \min_{p_i \in \hat{p}_i}\{\max_{1\le i\le n}\{d^*(p_i,\theta)\}\} \\ &< By\ lemma\ 4.10 > \\ &= \max_{1\le i\le n}\{\min_{p_i\in\hat{p}_i}\{d^*(p_i,\theta)\}\} \\ &< By\ lemma\ 4.8 > \\ &= \max_{1\le i\le n}\{-d^*(\hat{p}_i,\theta+\pi)\} \\ &= -\min_{1\le i\le n}\{d^*(\hat{p}_i,\theta+\pi)\} \end{aligned}$$

□

Lemma 4.10 *Let x and y be independent variables, and let $f(x)$ and $g(y)$ be functions $\Re^2 \mapsto \Re$ that have the minimum and the maximum in their domains. Then $\min_{x,y}\{\max\{f(x),g(y)\}\} = \max\{\min_x\{f(x)\},\min_y\{g(y)\}\}$ holds.*

Proof.

First, let fix x at x_0. Then,

$$\begin{aligned} \min_{x=x_0,y}\{\max\{f(x),g(y)\}\} &= \min_y\{\max\{f(x_0),g(y)\}\} \\ &= \begin{cases} f(x_0) & (f(x_0)\ge \min_y\{g(y)\}) \\ \min_y\{g(y)\} & (f(x_0) < \min_y\{g(y)\}) \end{cases} \\ &= \max\{f(x_0),\min_y\{g(y)\}\} \end{aligned}$$

Assume that $g(y)$ takes its minimum value at y_0. Since $\min_y\{g(y)\}$ is a constant independent of x, the above expression is modified as follows

$$\begin{aligned} \min_{x,y}\{\max\{f(x),g(y)\}\} &= \min_x\{\max\{g(y_0),f(x)\}\} \\ &= \max\{\min_x\{f(x)\},g(y_0)\} \\ &= \max\{\min_x\{f(x)\},\min_y\{g(y)\}\} \end{aligned}$$

□

We show that the half-plane representation of the inner convex hull of n ε-Points can be constructed in $O(n\log n)$ time and $O(n)$ space. Since the half-plane representation is essentially a lower envelope $\mathcal{L}$ of curves $\{d_i(\hat{p}_i,\theta)\}$, the time complexity of a construction depends on the complexity of $\mathcal{L}$ itself. Observe that the relative order between $d^*(\hat{p}_i,\theta)$ and $d^*(\hat{p}_j,\theta)$ changes at $\theta=\theta_0$ if and only if the left tangent of $\hat{p}_i$ become a common tangent of them at $\theta=\theta_0$. Since ε-Points are axis-parallel rectangles, the relative order between $d^*(\hat{p}_i,\theta)$ and $d^*(\hat{p}_j,\theta)$ changes at most once in each interval of $[0,\pi/2)$,$[\pi/2,\pi)$,$[\pi,3/2\pi)$,$[3/2\pi,2\pi)$. From the literature [2], we can conclude that $\mathcal{L}$ has a complexity of $O(n)$, and can be constructed in $O(n\log n)$ time and $O(n)$ space.

4.1.2 Half-Plane Representation of an Outer Convex Hull

We show that inner convex hull and outer convex hull are essentially the same under half-plane representation. In the following, we give a half-plane representation for an outer convex hull.

Lemma 4.11 *For an ε-Point $\hat{p}$, $\max_{p\in\hat{p}}\{d^*(p,\theta)\} = d^*(\hat{p},\theta)$ holds.*

Proof. From the definition of left tangent and lemma 4.5, a directed line $x\sin\theta - y\cos\theta = \max_{p\in\hat{p}}\{d^*(p,\theta)\}$ corresponds to $t_{\mathcal{L}}(\hat{p},\theta)$. □

Theorem 4.12 *Let S be a set of ε-Points $\hat{p}_1,\ldots,\hat{p}_n$. The function $d(\theta) = \max_{1\le i\le n}\{d^*(\hat{p}_i,\theta)\}$ is a half-plane representation of $R_{out}(S)$.*

Proof.

Since $R_{out}(S)$ is convex by theorem 3.11, $R_{out}(S)$ corresponds to $conv(R_{out}(S))$. Hence, a half-plane representation of $R_{out}(S)$ is calculated as follows

$$\begin{aligned} & d(R_{out}(S),\theta) \\ \equiv\ & d(conv(R_{out}(S)),\theta) \\ \equiv\ & d(conv(\cup_{C\in CH(S)}C),\theta) \\ & < By\ theorem\ 4.7 > \\ \equiv\ & \max_{C\in CH(S)}\{d^*(C,\theta)\} \\ \equiv\ & \max_{1\le i\le n, p_i\in\hat{p}_i}\{d^*(conv(\cup_{1\le i\le n}\{p_i\}),\theta)\} \\ & < By\ theorem\ 4.7 > \\ \equiv\ & \max_{1\le i\le n, p_i\in\hat{p}_i}\{\max_{1\le i\le n}\{d^*(p_i,\theta)\}\} \\ & < p_i's\ are\ independent\ each\ other > \\ =\ & \max_{1\le i\le n}\{\max_{p_i\in\hat{p}_i}\{d^*(p_i,\theta)\}\} \\ & < By\ lemma\ 4.11 > \\ =\ & \max_{1\le i\le n}\{d^*(\hat{p}_i,\theta)\} \end{aligned}$$

□

The theorem states that an inner convex hull and an outer convex hull are represented by the same kind of curve: envelope of sinusoidal curves. They can be calculated by the combinatorially same method although they seem to be defined as different kind of geometric objects.

By a similar discussion in section 4.3, the envelope of the curves $\{d^*(\hat{p}_i,\theta)\}$ can be constructed in $O(n\log n)$ time. The half-plane representation of an outer

convex hull introduced here is later used in section 4.2 to compute the maximum diameter of a set of points.

4.2 The Minimum and the Maximum of Possible Diameters

In this section, we show that the maximum and the minimum of possible diameters of a set of ε-Points can be calculated efficiently with half-plane representations of an inner convex hull and an outer convex hull.

Based on rotating caliper method[34], the diameter $diam(R)$ of a convex region R is given by $diam(R) = \max_{0\leq\theta<2\pi}\{w(R,\theta)\}$; here $w(R,\theta)$ is defined as $w(R,\theta) = d^*(R,\theta) - (-d^*(R,\theta+\pi))$, the distance between $t_{\mathcal{L}}(R,\theta)$ and $t_{\mathcal{L}}(R,\theta)$. Hence, the minimum and the maximum of possible diameters of a set of ε-Points $S = \{\hat{p_1},\ldots,\hat{p_n}\}$ is represented as follows

$$\begin{aligned} & \min_{A\in A(S)}\{diam(A)\} \\ = \ & \max_{0\leq\theta<2\pi}\{\min_{C\in CH(S)}\{w(C,\theta)\}\} \end{aligned}$$

$$\begin{aligned} & \max_{A\in A(S)}\{diam(A)\} \\ = \ & \max_{0\leq\theta<2\pi}\{\max_{C\in CH(S)}\{w(C,\theta)\}\} \end{aligned}$$

In the following, we show how $\min_{C\in CH(S)}\{w(C,\theta)\}$ and $\max_{C\in CH(S)}\{w(C,\theta)\}$ is represented by half-plane representations.

Theorem 4.13 *For a set S of ε-Points $\hat{p_1},\ldots,\hat{p_n}$, let $f(\theta)$ and $g(\theta)$ be functions defined as follows*

$$\begin{aligned} f(\theta) &= \max_{1\leq i\leq n}\{-d^*(\hat{p_i},\theta+\pi)\} \\ g(\theta) &= \min_{1\leq i\leq n}\{d^*(\hat{p_i},\theta)\} \end{aligned}$$

Then,

$$\min_{C\in CH(S)}\{w(C,\theta)\} = \max\{f(\theta)-g(\theta),0\}$$

holds.

Proof.

Let l_0 and l_1 be $l(f,\theta)$ and $l(g,\theta)$ respectively. We distinguish two cases according to the relative location of these two lines.

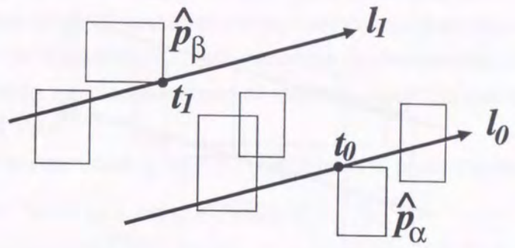

Figure 4.8: Case:$l_1 \subset H_{\mathcal{L}}(l_0)$

1. *Case* : $l_1 \subset H_{\mathcal{L}}(l_0)$ *(i.e.* $g(\theta) < f(\theta)$)

   By lemma 4.8 and lemma 4.11, l_0 is a right tangent of some ε-Point $\hat{p_\alpha}$, and l_1 is a left tangent of some ε-Point $\hat{p_\beta}$. By lemma 4.5, l_0 is the rightmost among the right tangents of ε-Points with angle θ; similarly l_1 is the leftmost. In the rest of the proof, t_0 and t_1 denotes points at which l_0 and l_1 touches $\hat{p_\alpha}$ and $\hat{p_\beta}$ respectively (see Fig. 4.8). Observe that t_0 and t_1 are distinct points: $\hat{p_\alpha}$ and $\hat{p_\beta}$ are separated by l_0 and l_1 since l_1 satisfies $l_1 \subset H_{\mathcal{L}}(l_0)$.

   We show that there is a possible convex hull C that satisfies $w(C,\theta) = f(\theta) - g(\theta)$. From the definition of $f(\theta)$ and $g(\theta)$, all ε-Points $\hat{p_i}$ satisfies $\hat{p_i} \cap H_{\mathcal{L}}(l_0) \neq \emptyset$ and $\hat{p_i} \cap H_{\mathcal{R}}(l_1) \neq \emptyset$. In addition, t_0 and t_1 are distinct points. Hence, a possible set of points A exists that satisfies

   $$\begin{cases} f_A(\hat{p_\alpha}) &= t_0 \\ f_A(\hat{p_\beta}) &= t_1 \\ f_A(\hat{p_i}) &\in H_{\mathcal{L}}(l_0)\cap H_{\mathcal{R}}(l_1)\ (i\neq\alpha,\beta) \end{cases}$$

   Note that t_0 and t_1 are vertices of $conv(A)$. It means that $w(A,\theta)$ corresponds to the distance between l_0 and l_1; $w(conv(A),\theta) = f(\theta) - g(\theta)$. Moreover, any possible convex hull C satisfies

   $$\forall C \in CH(S)\ \ w(C,\theta) \geq f(\theta) - g(\theta)$$

   since C contains at least one point p in $\hat{p_\alpha}$ and one point q in $\hat{p_\beta}$ that are separated by l_0 and l_1.

   From the discussion, we conclude that $\min_{C\in CH(S)}\{w(C,\theta)\} = f(\theta) - g(\theta)$.

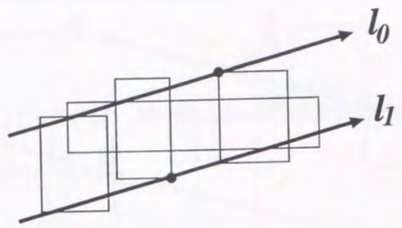

Figure 4.9: Case:$l_0 \subset H_{\mathcal{L}}(l_1)$

2. Case:$l_0 \subset H_{\mathcal{L}}(l_1)$ (*i.e.* $g(\theta) \geq f(\theta)$)

   We show that $\min_{C \in CH(S)}\{w(C,\theta)\} = 0$. Since $\min_{C \in CH(S)}\{w(C,\theta)\} \geq 0$, it is sufficient to prove that there is a possible arrangement A that satisfies $w(conv(A),\theta) = 0$.

   From the definition of $f(\theta)$ and $g(\theta)$, all ε-Points $\hat{p}_i$ satisfies

   $$\hat{p}_i \cap H_{\mathcal{L}}(l_0) \neq \emptyset \wedge \hat{p}_i \cap H_{\mathcal{R}}(l_1) \neq \emptyset$$

   Since all ε-Points are connected regions, this implies that they stride across the region $H_{\mathcal{R}}(l_0) \cap H_{\mathcal{L}}(l_1)$ (see Fig. 4.9). Hence, any line l that is parallel to l_0 and lies in $H_{\mathcal{R}}(l_0) \cap H_{\mathcal{L}}(l_1)$ intersects with all ε-Points; which implies that there is a possible arrangement A in which all points lie on l. Clearly, $w(conv(A),\theta)$ is 0.

   From the discussion above, we can conclude that $\min_{C \in CH(S)}\{w(C,\theta)\} = 0$.

By combing the two cases, we get the following equation

$$\begin{aligned} & \min_{C \in CH(S)}\{w(C,\theta)\} \\ = & \begin{cases} f(\theta) - g(\theta) & (f(\theta) > g(\theta)) \\ 0 & (f(\theta) \leq g(\theta)) \end{cases} \\ = & \max\{f(\theta) - g(\theta), 0\} \end{aligned}$$

□

Remember that $f(\theta)$ in theorem 4.13 is $-\min_{1 \leq i \leq n}\{d^*(\hat{p}_i, \theta + \pi)\}$. That is the half-plane representation $d(R_{in}(S))$ in theorem 4.9, and $\min_{C \in CH(S)}\{w(C,\theta)\}$ can be re-expressed as $\max\{d(R_{in}(S),\theta) + d(R_{in}(S),\theta + \pi), 0\}$.

Hence, once we have $d(R_{in}(S))$, $\max_{0 \leq \theta < 2\pi}\{\min_{C \in CH(S)}\{w(C,\theta)\}\}$ can be computed in time proportional to the complexity of the lower envelope of curves $\{d^*(\hat{p}_i,\theta)\}$. By the discussion in 4.3, the envelope can be constructed in $O(n \log n)$ time for n ε-Points, and the minimum of possible diameters can be computed in additional $O(n)$ time.

Further, we express $\max_{C \in CH(S)}\{w(C,\theta)\}$ by half-plane representations.

Theorem 4.14 *Let S be a set of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$.*

Then, $\max_{C \in CH(S)}\{w(C,\theta)\} = \max_{1 \leq i,j \leq n, i \neq j}\{d^*(\hat{p}_i,\theta) + d^*(\hat{p}_j,\theta + \pi)\}$.

Proof.

First, we re-express $\max_{C \in CH(S)}\{w(C,\theta)\}$ as follows.

$$\begin{aligned} & \max_{C \in CH(S)}\{w(C,\theta)\} \\ = & \max_{A \in A(S)}\{w(conv(A),\theta)\} \\ = & \max_{1 \leq i,j \leq n, p_i \in \hat{p}_i, p_j \in \hat{p}_j}\{d^*(p_i,\theta) - d^*(p_j,\theta)\} \\ = & \max\{0, \max_{1 \leq i,j \leq n, i \neq j, p_i \in \hat{p}_i, p_j \in \hat{p}_j}\{d^*(p_i,\theta) - d^*(p_j,\theta)\}\} \end{aligned}$$

Since p_i and p_j are independent if i and j are different, the second term in the previous expression can be rewrited as follows.

$$\max\left\{0, \max_{1 \leq i,j \leq n, i \neq j}\left\{\max_{p_i \in \hat{p}_i}\{d^*(p_i,\theta)\} - \min_{p_j \in \hat{p}_j}\{d^*(p_j,\theta)\}\right\}\right\}$$

Also, the second term of the above expression is not less than 0 since in case of $\max_{p_i \in \hat{p}_i}\{d^*(p_i,\theta)\} \leq \min_{p_j \in \hat{p}_j}\{d^*(p_j,\theta)\}$, then, once i and j are exchanged, $\max_{p_j \in \hat{p}_j}\{d^*(p_j,\theta)\} \geq \min_{p_i \in \hat{p}_i}\{d^*(p_i,\theta)\}$ holds.

Hence, the above expression can be expressed as follows.

$$\begin{aligned} & \max_{C \in CH(S)}\{w(C,\theta)\} \\ = & \max_{1 \leq i,j \leq n, i \neq j}\left\{\max_{p_i \in \hat{p}_i}\{d^*(p_i,\theta)\} - \min_{p_j \in \hat{p}_j}\{d^*(p_j,\theta)\}\right\} \\ = & \max_{1 \leq i,j \leq n, i \neq j}\left\{\max_{p_i \in \hat{p}_i}\{d^*(p_i,\theta)\} + \max_{p_j \in \hat{p}_j}\{d^*(p_j,\theta + \pi)\}\right\} \\ & < \textit{By lemma } 4.11 > \\ = & \max_{1 \leq i,j \leq n, i \neq j}\{d^*(\hat{p}_i,\theta) + d^*(\hat{p}_j,\theta + \pi)\} \end{aligned}$$

□

By theorem 4.14, we can compute $\max_{C \in CH(S)}\{w(C,\theta)\}$ from the normalized half-plane representation of ε-Points. However, the direct evaluation of the function is inefficient; it involves the computation of envelopes of $O(n^2)$ curves.

In theorem 4.15, we show a simplified version of the function in theorem 4.14. We denote the second maximum value as $\max^{(2)}$ (if the second maximum value does not exist, the value of $\max^{(2)}$ is defined as $-\infty$).

Theorem 4.15 *For a set S of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$, let functions $f(\theta), f_2(\theta)$ and a mapping $M(\theta)$ from a real to a set of natural numbers be defined as follows*

$$\begin{aligned} f(\theta) &= \max_{1\leq i\leq n}\{d^*(\hat{p}_i,\theta)\} \\ f_2(\theta) &= \max^{(2)}_{1\leq i\leq n}\{d^*(\hat{p}_i,\theta)\} \\ i \in M(\theta) &\Leftrightarrow d^*(\hat{p}_i,\theta) = f(\theta) \end{aligned}$$

Then, $\max_{1\leq i,j\leq n, i\neq j}\{d^*(\hat{p}_i,\theta)+d^*(\hat{p}_j,\theta+\pi)\}$ *is given by the following equation*

$$\begin{aligned} &\max_{1\leq i,j\leq n,i\neq j}\{d^*(\hat{p}_i,\theta)+d^*(\hat{p}_j,\theta+\pi)\} \\ &= \begin{cases} f(\theta)+f(\theta+\pi) \\ \quad (M(\theta)\neq M(\theta+\pi) \vee |M(\theta)|\geq 2) \\ \max\{f(\theta)+f_2(\theta+\pi), f_2(\theta)+f(\theta+\pi), 0\} \\ \quad (M(\theta)=M(\theta+\pi)\wedge |M(\theta)|=1) \end{cases} \end{aligned}$$

Proof.

1. Case:$M(\theta)\neq M(\theta+\pi)\vee|M(\theta)|\geq 2$

   In this case, there are distinct ε-Points $\hat{p}_i$ and $\hat{p}_j$ that satisfy

   $$d^*(\hat{p}_i,\theta) = f(\theta)\ , d^*(\hat{p}_j,\theta+\pi) = f(\theta+\pi)$$

   In addition, the next inequality holds

   $$\begin{aligned} &\max_{1\leq i,j\leq n,i\neq j}\{d^*(\hat{p}_i,\theta)+d^*(\hat{p}_j,\theta+\pi)\} \\ \leq\ &\max_{1\leq i\leq n}\{d^*(\hat{p}_i,\theta)\}+\max_{1\leq j\leq n}\{d^*(\hat{p}_j,\theta+\pi)\} \end{aligned}$$

   Hence, we can conclude that:

   $$\max_{1\leq i,j\leq n,i\neq j}\{d^*(\hat{p}_i,\theta)+d^*(\hat{p}_j,\theta+\pi)\} = f(\theta)+f(\theta+\pi)$$

2. Case:$M(\theta)=M(\theta+\pi)\wedge|M(\theta)|=1$

   In this case, $\max_{1\leq i,j\leq n,i\neq j}\{d^*(\hat{p}_i,\theta)+d^*(\hat{p}_j,\theta+\pi)\}$ is NOT $f(\theta)+f(\theta+\pi)$ since both $f(\theta)$ and $f(\theta+\pi)$ are determined by the same ε-Point.

   By the definition of $\max^{(2)}$, if $n\geq 2$, the ε-Point that determines $f(\theta)$ and the ε-Point that determines $f_2(\theta)$ are always different. If $n=1$, $f_2(\theta)$ is $-\infty$ and also $w(C,\theta)=0$. From these observations, the next equation holds

$$\begin{aligned} &\max_{1\leq i,j\leq n,i\neq j}\{d^*(\hat{p}_i,\theta)+d^*(\hat{p}_j,\theta+\pi)\} \\ &= \max\left\{\begin{array}{l} \max_{1\leq i\leq n}\{d^*(d_i,\theta)\}+\max^{(2)}_{1\leq j\leq n}\{d^*(d_j,\theta+\pi)\}, \\ \max^{(2)}_{1\leq i\leq n}\{d^*(d_i,\theta)\}+\max_{1\leq j\leq n}\{d^*(d_j,\theta+\pi)\}, \\ 0 \end{array}\right\} \\ &= \max\{f(\theta)+f_2(\theta+\pi), f_2(\theta)+f(\theta+\pi), 0\} \end{aligned}$$

□

Note that $f(\theta)$ is the half-plane representation of an outer convex hull introduced in section 4.1.2; $f(\theta)$ can be constructed as a lower envelope in $O(n\log n)$ time. $M(\theta)$ can be simultaneously constructed as an additional attribute of the components of the envelope. $f_2(\theta)$ can be constructed as the 2-level envelope of the curves $\{d^*(\hat{p}_i,\theta)\}$ in $O(n\log^* n\log n)$ expected time and $O(n)$ space by a method in [1].

Once we have $f(\theta)$,$f_2(\theta)$ and $M(\theta)$, the maximum of possible diameters can be computed in time proportional to the complexity of $f_2(\theta)$, that is, $O(n\log^* n\log n)$. We can conclude that the overall time to compute the maximum of possible diameters is $O(n\log^* n\log n)$ expected time.

As a little improvement, after we have $f(\theta)$ and $M(\theta)$, we check a condition that $M(\theta)\neq M(\theta+\pi)$ or $|M(\theta)|\geq 2$ holds for any θ. If this condition is established, the construction of $f_2(\theta)$ become unnecessary; the maximum of possible diameters can be computed in $O(n\log n)$ time. For example, such a case occurs when the sizes of ε-Points are the same.

4.3 Construction of an Inner Convex Hull

In this section, we give a re-mapping from a half-plane representation of an inner convex hull to its boundary. The key of re-mapping is that only a few half-planes contribute to the boundary of an inner convex hull while a half-plane representation may represent infinitely many half-planes. In the following lemma 4.16 and theorem 4.17, we show which one contributes to the boundary of an inner convex hull among the half-planes represented by the half-plane representation in theorem 4.9.

Lemma 4.16 *Let R be a convex region in the plane, and let θ_0 and θ_1 be angles that satisfy $0<\theta_1-\theta_0<\pi$. Then, the region $R_0=\cap_{\theta_0\leq\theta\leq\theta_1}\{H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta))\}$ equals to $H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta_0))\cap H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta_1))$.*

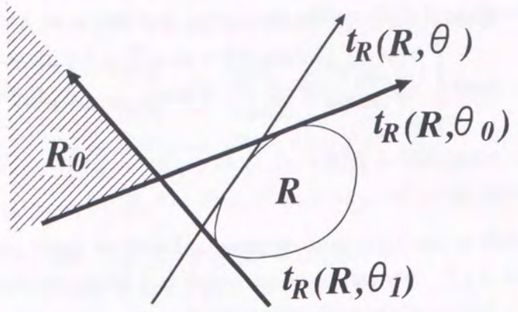

Figure 4.10: The right tangent $t_{\mathcal{R}}(R,\theta)$ does not intersect with the shaded region R_0

Proof. Observe that for any θ in $\theta_0 < \theta < \theta_1$, $t_{\mathcal{R}}(R,\theta)$ satisfies $H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta_0)) \cap H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta_1)) \subseteq H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta))$ (see Fig. 4.10).

Hence,

$$\begin{aligned} R_0 &= \bigcap_{\theta_0 \le \theta \le \theta_1} \{H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta))\} \\ &= H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta_0)) \bigcap H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta_1)) \bigcap (\bigcap_{\theta_0 < \theta < \theta_1} \{H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta))\}) \\ &= H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta_0)) \bigcap H_{\mathcal{L}}(t_{\mathcal{R}}(R,\theta_1)) \end{aligned}$$

□

Note that if $\theta_1 - \theta_0$ is greater than π in the lemma, R_0 becomes empty.

Theorem 4.17 *For a set S of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$, let $d(\theta)$ be defined by $d(\theta) = -\min_{1 \le i \le n}\{d^*(\hat{p}_i, \theta + \pi)\}$, and let $\theta_1, \ldots, \theta_N (0 \le \theta_1 < \cdots < \theta_N < 2\pi)$ be the angles at which the set of ε-Points that give a piecewise minimum changes.*

The restriction $d'(\theta)$ of $d(\theta)$ to $\theta = \theta_1, (\theta_1+\theta_2)/2, \theta_2, \ldots, \theta_i, (\theta_i+\theta_{i+1})/2, \theta_{i+1}, \ldots, (\theta_{N-1}+\theta_N)/2, \theta_N$ satisfies $d(\theta) \equiv d'(\theta)$.

Proof. Consider an interval $[\alpha, \beta]$ of θ such that an ε-Point $\hat{p}_i$ satisfies $d(\theta) = -d^*(\hat{p}_i, \theta + \pi)$ for any θ in the interval. By lemma 4.4, $l(d,\theta)$ equals to $t_{\mathcal{R}}(\hat{p}_i, \theta)$ for all θ in the interval.

Hence, by lemma 4.16, the restriction of $d(\theta)$ to $[0,\theta_i] \cup [\frac{\theta_i+\theta_{i+1}}{2}, \frac{\theta_i+\theta_{i+1}}{2}] \cup [\theta_{i+1}, 2\pi]$ does not affect its corresponding region (Note $\left|\frac{\theta_i+\theta_{i+1}}{2} - \theta_i\right| \le \pi$). □

Hence, once we have the coordinates of vertices of the lower envelope corresponding to the piecewise minimum function in theorem 4.9, the boundary of an

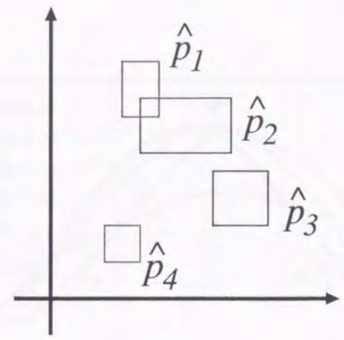

Figure 4.11: An example configuration of ε-Points

inner convex hull can be constructed by computing the intersection of $2N-1$ half-planes. In more detail, we construct the lower envelope $\mathcal{L}$ of n curves $\{d^*(\hat{p}_i, \theta + \pi) | 1 \le i \le n\}$ in the interval $[0, 2\pi]$; we get the coordinates $(\omega_i, \min\{d^*(\hat{p}_i, \omega_i + \pi)\})$ of N vertices of $\mathcal{L}$. Then for the function $d(\theta)$ in theorem 4.9, construct the intersection of half-planes $\bigcap_{\theta=\omega_1,(\omega_1+\omega_2)/2,\omega_2,\ldots,(\omega_{N-1}+\omega_N)/2,\omega_N} H_{\mathcal{L}}(l(d,\theta))$,which gives the boundary of the inner convex hull. For example, consider to calculate the inner convex hull of ε-Points in Fig. 4.11. Their normalized half-plane representations are given by piecewise maximum functions in Fig. 4.12. Based on these functions, a half-plane representation of an inner convex hull is given by the function in fig. 4.13. In this figure, $\hat{p}_4$, $\hat{p}_3$, $\hat{p}_1$ and $\hat{p}_4$ gives piecewise maximums in the intervals $[0,\theta_1]$, $[\theta_1,\theta_2]$, $[\theta_2,\theta_3]$ and $[\theta_3, 2\pi]$ respectively. The half-plane representation is simplified as shown in Fig. 4.14, and the boundary of the inner convex hull is given by the intersection of three half-planes as shown in Fig. 4.15.

Since the complexity of $\mathcal{L}$ is $O(n)$, the boundary of the inner convex hull is determined by the intersection of $O(n)$ half-planes. The intersection can be constructed in $O(n \log n)$ time and $O(n)$ space[31].

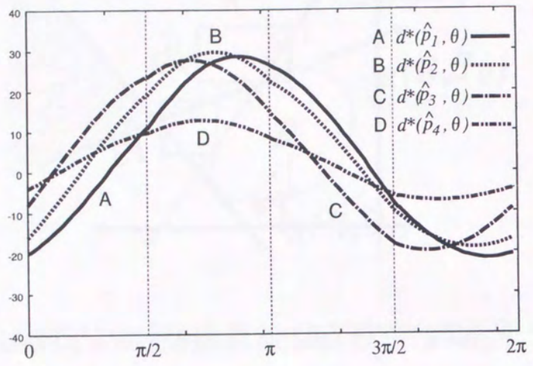

Figure 4.12: Half-plane representations of ε-Points in Fig. 4.11

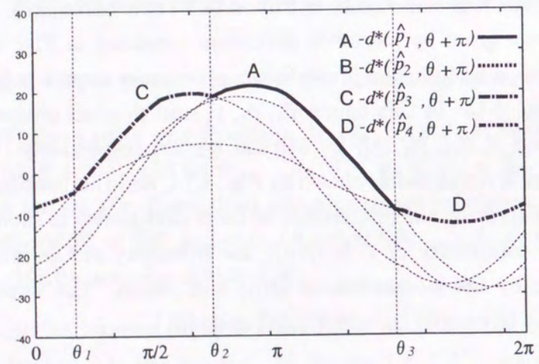

Figure 4.13: Half-plane representation of the inner reliable region of ε-Points in Fig. 4.11

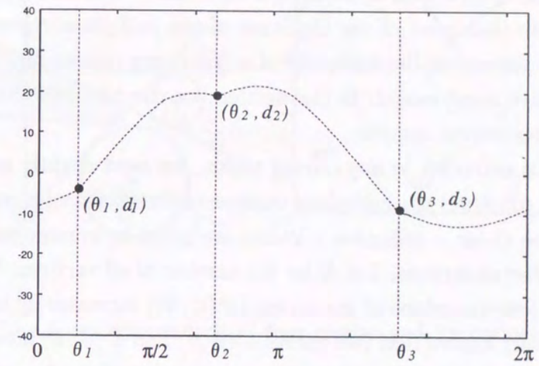

Figure 4.14: Simplified half-plane representation of the inner reliable region of ε-Points in Fig. 4.11

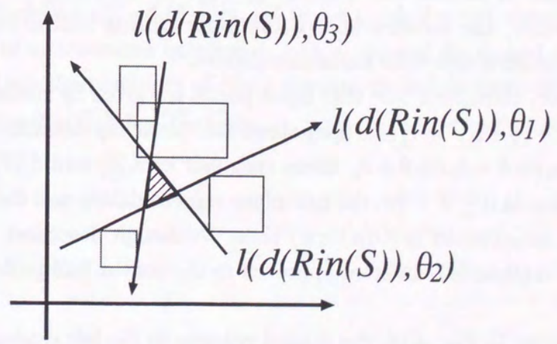

Figure 4.15: The inner reliable region of ε-Points in Fig. 4.11

4.4 Other Input Models

In the discussion above, each input point is modeled as an ε-Point; an axis-parallel rectangle. Note that most of our theorems about half-plane representation and diameter only depend on the convexity of ε-Point; any convex region can be used as an alternative input model. In this section, we describe how our methods are applied to other convex regions.

If ε-Point is extended to any convex region, we need slightly more computational time to construct the half-plane representation of an inner convex hull. We briefly describe about a case that ε-Points are given by convex polygons with a constant number of vertices. Let N be the number of all vertices. Now, the complexity of the lower envelope of the curves $\{d^*(\hat{p}_i, \theta)\}$ increases up to $O(N \log^* N)$ since we can only ensure that two curves $d^*(\hat{p}_i, \theta)$ and $d^*(\hat{p}_j, \theta)$ intersects at most constant times in $0 \leq \theta < 2\pi$. Thus, the computation of the boundary of $R_{in}(S)$ also needs $O(N \log^* N \log N)$ time. Similarly, the minimum of possible diameters can be computed in $O(N \log^* N \log N)$ time, and the maximum can be computed in $O(N \log^* N \log N)$ expected time.

In contrast, if each ε-Point is given as an axis-parallel rectangle with the same width and height, the problem becomes simple. Let p and q be points whose coordinates are $p = (p_x, p_y)$ and $q = (q_x, q_y)$ respectively, and let their corresponding ε-Points be $\hat{p} = (p_x, p_y, \epsilon, \delta)$ and $\hat{q} = (q_x, q_y, \epsilon, \delta)$. Then, as shown in the example in section 4.1, $d^*(\hat{p}, \theta_0) \leq d^*(\hat{q}, \theta_0)$ is equivalent to $d^*(p, \theta_0) \leq d^*(q, \theta_0)$. Hence, the construction of the inner convex hull is reduced to the convex hull construction with exact coordinates.

In addition, consider a case that input points are given by circles. For a circle $C_i : (x - a_i)^2 + (y - b_i)^2 = r_i^2$, its normalized half-plane representation is given by: $d^*(C_i, \theta) = a_i \sin\theta - b_i \cos\theta + r_i$. Since each pair of $d_i(\theta)$ and $d_j(\theta)$ intersects at most two times in $0 \leq \theta < 2\pi$, the half-plane representation and the inner convex hull can be constructed in $O(n \log n)$ time. Although described in a different context, our method is a different approach to the convex hull problem described in [32].

For example, in Fig. 4.16, the shaded polygon in the left represents the inner convex hull of the four circles C_1, C_2, C_3 and C_4: C_1 is centered at $(8, -8)$ with radius 4, C_2 is centered at $(11, -2)$ with radius 2, C_3 is centered at $(-6, 6)$ with radius 2, and C_4 is centered at $(-7, -7)$ with radius 4. Their normalized half-plane representations are

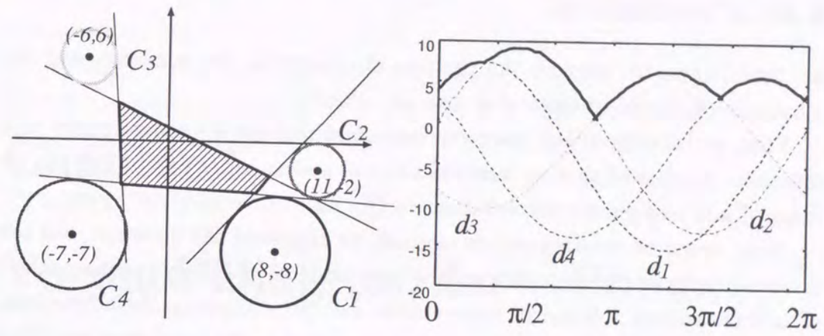

Figure 4.16: The inner convex hull of four circles and its corresponding upper envelope

$$
\begin{aligned}
d_1(\theta) &= 8\sin\theta + 8\cos\theta + 4 \\
d_2(\theta) &= 11\sin\theta + 2\cos\theta + 2 \\
d_3(\theta) &= -6\sin\theta - 6\cos\theta + 2 \\
d_4(\theta) &= -7\sin\theta + 7\cos\theta + 4
\end{aligned}
$$

The half-plane representation of the inner convex hull, that is, the upper envelope of $\{-d_1(\theta + \pi), -d_2(\theta + \pi), -d_3(\theta + \pi), -d_4(\theta + \pi)\}$ shown in the right side consists of intersections between d_1 and d_2, d_2 and d_3, d_3 and d_4, d_4 and d_1. This means that the boundary of the inner convex hull is given by $t_{\mathcal{CR}}(C_1, C_2)$, $t_{\mathcal{CR}}(C_2, C_3)$, $t_{\mathcal{CR}}(C_3, C_4)$, $t_{\mathcal{CR}}(C_4, C_1)$.

4.5 Conclusion

In this chapter, we consider the problem of computing the minimum and the maximum of possible diameters of a set of ε-Points.

First, we introduced *half-plane representation* that maps a convex region to a function. We showed that an inner convex hull and an outer convex hull can be mapped into half-plane representations in $O(n \log n)$ time and $O(n)$ space.

Next, based on rotating caliper method, we expressed the minimum and the maximum of possible diameters as functions made of half-plane representations of an inner convex hull and an outer convex hull. By computing these functions, the minimum of possible diameters can be computed in $O(n \log n)$ time and $O(n)$ space, and the maximum can be computed in $O(n \log^* n \log n)$ expected time and $O(n)$ space.

Finally, we showed that once we have a half-plane representation of an inner convex, its boundary can be constructed in $O(n \log n)$ time.

Chapter 5

Voronoi Diagram for ε-Points

In this chapter, we address the problem of computing Voronoi diagram for a set of ε-Points. For a given set P of points (called *sites*) $p_1, p_2, \ldots, p_n$, Voronoi diagram $V(P)$ is defined as the partition of the plane into Voronoi regions $V(i) = \{p | d(p, p_i) < d(p, p_j)\ ,\ i \neq j\}$: the partition of the plane according to the nearest sites. Voronoi diagram is a fundamental tool in computational geometry since it provides solutions for many apparently unrelated problems such as largest empty circle, nearest neighbor search and pattern recognition. It is known that Voronoi diagram can be constructed in optimal $\Theta(n \log n)$ time for n points in the plane[31]. The generalization of sites (e.g. segment, circle, polygon, curve) or distance metrics (e.g. L^k metric) has also been studied.

Now, consider what happens if input coordinates contain errors. Since each site moves within a tolerance of error, the *nearest* sites can not be defined any more; the partition of the plane is not unique. It means that existing algorithms can not be used since they are designed to construct the topological structure of the unique partition.

The key idea is that we redefine a Voronoi diagram as a set of Voronoi regions, and allow overlaps between them. Although a Voronoi diagram is originally defined as a partition of the plane, in other view, it can be seen as a set of Voronoi regions. Edges and vertices that appear in a Voronoi diagram can be considered as common parts between the regions. Without input error, Voronoi regions are pairwise disjoint except their boundaries; the Voronoi diagram is a partition of the plane. If input error exists, Voronoi regions overlap each other and do not represent a partition of the plane anymore. However, their intersections give the information about edges and vertices in possible diagrams.

Based on the idea, we concentrate on the construction of Voronoi regions. For a site, we consider two types of Voronoi regions: *weak Voronoi region* and *strong Voronoi region*. The weak Voronoi region of a site is the region in which the site can be the nearest site in at least one possible arrangement; the strong Voronoi region of a site is the region in which the site is always the nearest site in all possible arrangements. By a pair of the weak Voronoi region and the strong Voronoi region of a site, we represent the accuracy of the Voronoi region of the site.

For a set of n ε-Points given as sites, we consider the problem of constructing all weak and strong Voronoi regions. We show that weak Voronoi regions can be constructed in $O(n^2\alpha(n)) \sim O(n^2 \log^* n \log n)$ time ($\alpha(n)$ is the inverse of Ackermann's function) and that strong Voronoi regions can be constructed in $O(n \log^* n \log n + k)$ time (k is the number of intersections between ε-Points). The construction time depends on the configuration of ε-Points.

5.1 Preliminaries

Definition 5.1 (Distance between a point and an ε-Point)
For a point p, and an ε-Point $\hat{q}$, let $m(p,\hat{q}) = \min\{d(p,q)|q \in \hat{q}\}$ and $M(p,\hat{q}) = \max\{d(p,q)|q \in \hat{q}\}$ be distance functions between a point and an ε-Point.

The functions $m(p,\hat{q})$ and $M(p,\hat{q})$ represent the minimum and the maximum of possible distances between p and a possible point of $\hat{q}$.

Definition 5.2 (Weak Voronoi Region)
Let S be a set of ε-Points $\hat{p_1},\ldots,\hat{p_n}$. The weak Voronoi region r_i of $\hat{p_i}$ is defined as follows

$$p \in r_i \Leftrightarrow \exists A \in A(S), d(p, f_A(\hat{p_i})) = \min\{d(p, f_A(\hat{p_j}))|1 \le j \le n\}$$

Definition 5.3 (Strong Voronoi Region)
Let S be a set of ε-Points $\hat{p_1},\ldots,\hat{p_n}$. The strong Voronoi region R_i of $\hat{p_i}$ is defined as follows

$$p \in R_i \Leftrightarrow \forall A \in A(S), d(p, f_A(\hat{p_i})) = \min\{d(p, f_A(\hat{p_j}))|1 \le j \le n\}$$

In other words, r_i and R_i are the union and the intersection of all possible Voronoi regions $V(i)$.

In the following, we consider the construction of weak Voronoi regions and strong Voronoi regions. We identify a region with a set of points; the notation $\cap,\subseteq$ are used for regions in the same meaning for sets. The boundary of a region R is denoted as ∂R.

5.2 Weak Voronoi Regions

In this section, we introduce geometric properties of weak Voronoi regions. Based on the properties, we develop a construction algorithm.

5.2.1 Geometric Properties of a Weak Voronoi Region

First, we consider an elementary weak Voronoi region r_{ij} determined by two ε-Points $\hat{p_i}$ and $\hat{p_j}$.

Definition 5.4 (Weak Voronoi Region r_{ij})

$$r_{ij} = \{p|m(p,\hat{p_i}) \le M(p,\hat{p_j})\}$$

The region r_{ij} is the weak Voronoi region r_i in a case of $S = \{\hat{p_i},\hat{p_j}\}$(see Fig. 5.1). The boundary ∂r_{ij} consists of segments of lines and a parabola (depicted as dashed curve in Fig. 5.1). Since for any point that is not in r_{ij}, an ε-Point $\hat{p_i}$ can not be the nearest site in any possible arrangements, r_i is given by $\cap_{j\neq i} r_{ij}$.

We further investigate properties of a weak Voronoi region: its shape and connectivity.

Theorem 5.5 *Let S be a set of ε-Points $\hat{p_1},\ldots,\hat{p_n}$ and let $M^*(p)$ be the piecewise minimum function $\min\{M(p,\hat{p_i})|1 \le i \le n\}$. The necessary and sufficient condition for a point p to be in a weak Voronoi region r_i is $m(p,\hat{p_i}) \le M^*(p)$.*

Proof.

1. Proof of necessity

   If a point p satisfies $m(p,\hat{p_i}) \le M^*(p)$, there is a possible arrangement A that satisfies

   $$d(p, f_A(\hat{p_i})) \le M^*(p)$$

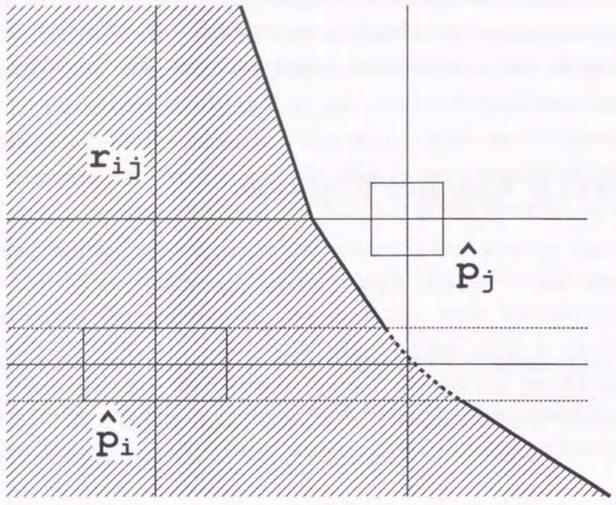

Figure 5.1: A weak Voronoi region r_{ij}

On the other hand, by the definition of $M^*(p)$, there is a possible arrangement A' that satisfies

$$\forall j \; j \neq i, \; d(p, f_{A'}(\hat{p_j})) \geq M^*(p)$$

Consider an arrangement B that is the same as A' but the possible point of $\hat{p_i}$ is the one in A. In the arrangement B, $f_B(\hat{p_i})$ is the nearest site of p since B satisfies the above two inequalities at the same time. From the definition 5.2, the existence of B implies $p \in r_i$.

2. Proof of sufficiency

   Let us assume $m(p, \hat{p_i}) > M^*(p)$. Then, any possible point of $\hat{p_i}$ can not be the nearest site of p in any possible arrangements. This contradicts to the condition of $p \in r_i$. □

In the following, we show that a weak Voronoi region is a simple connected region.

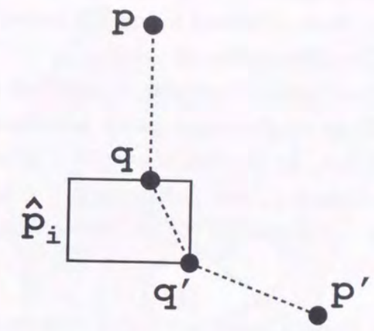

Figure 5.2: two points p and p' in a weak Voronoi region and their nearest points q and q' on $\partial\hat{p_i}$

Lemma 5.6 *Let S be a set of ε-Points. For an ε-Point $\hat{p_i}$ in S, the weak Voronoi region r_i satisfies $\hat{p_i} \subseteq r_i$.*

Proof.

Let p be any point in $\hat{p_i}$. Consider a possible arrangement A that satisfies

$$\begin{cases} d(p, f_A(\hat{p_i})) = 0 \\ d(p, f_A(\hat{p_j})) = M(p, \hat{p_j}) \quad (j \neq i) \end{cases}$$

Since $M(p, \hat{p_j}) \geq 0$, the nearest site of p in A is $f_A(\hat{p_i})$, p itself; which implies $p \in r_i$. Thus, we can conclude that $\hat{p_i} \subseteq r_i$. □

Theorem 5.7 *A weak Voronoi region is connected.*

Proof.

Consider a weak Voronoi region r_i and a point p in it. Let q be the point on $\partial\hat{p_i}$ that is the nearest point from p (see Fig. 5.2).

First, consider a possible arrangement of A that satisfies

$$f_A(\hat{p_i}) = q, d(p, f_A(\hat{p_j})) = M(p, \hat{p_j}) \; (j \neq i)$$

Since $d(p, q) = m(p, \hat{p_i})$, by theorem 5.5, the arrangement A satisfies

$$d(p, f_A(\hat{p_i})) = \min\{d(p, f_A(\hat{p_j}))|1 \leq j \leq n\}$$

Hence, in the Voronoi diagram $V(A)$, the point p is contained in the Voronoi region R of $q(= f_A(\hat{p}_i))$. Since a Voronoi region is a convex region, the segment $\overline{pq}$ is also contained in R; which implies $\overline{pq} \subseteq r_i$.

Next, consider any two points p and p' in r_i and their nearest points q and q' on $\partial\hat{p}_i$ respectively. From the discussion above, both the segments $\overline{pq}$ and $\overline{p'q'}$ satisfies $\overline{pq} \subseteq r_i$ and $\overline{p'q'} \subseteq r_i$. In addition, the segment $\overline{qq'}$ satisfies $\overline{qq'} \subseteq r_i$ since q and q' are in a convex region $\hat{p}_i$, and $\hat{p}_i$ satisfies $\hat{p}_i \subseteq r_i$ by lemma 5.6. Hence, the chain C of $\overline{pq} - \overline{qq'} - \overline{p'q'}$ satisfies $C \subseteq r_i$(see Fig. 5.2); which ensures the connectivity of r_i. □

Theorem 5.8 *A weak Voronoi region is a simple connected region.*

Proof.

Let r_i be a weak Convex region. Since we have shown the connectivity of r_i in theorem 5.7, it suffices to prove that r_i does not has a hole R (a hole R of r_i is a bounded region which satisfies $\partial R \subseteq r_i$ and whose interior is outside of r_i). By lemma 5.6, we can assume that R is outside of $\hat{p}_i$ without loss of generality.

Assume that r_i has a hole R. Let p be a point in R and let q be the nearest point of p on $\partial\hat{p}_i$, and let l be the half line emanating from q to p. Since R is a bounded region, ∂R is a closed curve; l intersects with ∂R at a point p_0 that is farther from q than p. (see Fig. 5.3).

Since p_0 satisfies $p_0 \in \partial R \subseteq r_i$, the segment $\overline{p_0 q}$ is contained in r_i by a similar argument in the proof of theorem 5.7. Hence, p, which in on $\overline{p_0 q}$, is in r_i; this contradicts to the condition that any point in R is not contained in r_i. From the discussion above, we can conclude that r_i has no hole. □

By theorem 5.7, theorem 5.8 and their proof, the shape of a weak Voronoi region is generalized-star-shaped defined in [20], that is, *a region R in the plane is generalized-star-shaped with nucleus C, $C \subseteq R$, if for any point $r \in R$ there exists a point $c \in C$ such that the line segment $\overline{rc}$ completely within R.*

Hence, essentially, we can compute the boundary of r_i, that is $\cap_{j \neq i} r_{ij}$, by a construction of the lower envelope of the curves $\{r_{ij}\}$ under a polar coordinate system around a point p in $\hat{p}_i$. Since any pair of curves r_{ij} and r_{ik} intersects at most constant times, we can compute the lower envelope in $O(n \log^* n \log n)$ time [2]; all weak Voronoi regions can be computed in $O(n^2 \log^* n \log n)$ time. However, this construction is inefficient since redundant ε-Points ,which does not contribute to the boundary of a weak Voronoi region r_i, are always involved.

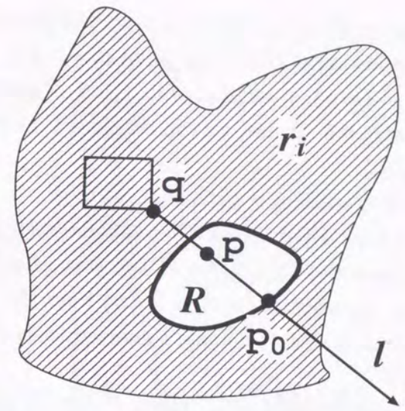

Figure 5.3: A weak Voronoi region r_i which has a hole R inside

Since the weak Voronoi region of an ε-Point $\hat{p}_i$ is determined by the neighbors of $\hat{p}_i$, we can improve the construction if we know neighbors of each ε-Point in advance. More exactly, by theorem 5.5, a point p is on the boundary of r_i only if p satisfies $m(p, \hat{p}_i) = M^*(p)$. If we know, for each point p in the plane, which ε-Point determines $M^*(p)$, we can exclude redundant ε-Points in the construction of r_i.

Based on the idea, we consider a partition of the plane according to which ε-Point determines $M^*(p)$.

Definition 5.9 (Partition $D(S)$ of the Plane)

Let S be a set of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$. $D(S)$ is the partition of the plane into regions $D_1(S), \ldots, D_n(S)$ defined as follows

$$D_i(S) = \{p | M(p, \hat{p}_i) = M^*(p)\}$$

Now, we introduce a theorem that relates the partition $D(S)$ and a weak Voronoi region.

Theorem 5.10 (Components of r_i)

Let S be a set of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$.

A weak Voronoi region r_i is expressed by $r_i = \cup_j (r_{ij} \cap D_j(S))$.

Proof.

First, we show that $r_i \supseteq \cup_j (r_{ij} \cap D_j(S))$ is true by proving $r_i \supseteq r_{ij} \cap D_j(S)$ for any j. From the definition of $D(S)$, any point p in $D_j(S)$ satisfies $M(p, \hat{p_j}) = M^*(p)$. If p is in $r_{ij} \cap D_j(S)$, p satisfies $m(p, \hat{p_i}) \leq M(p, \hat{p_j}) = M^*(p)$; which implies $p \in r_i$ by theorem 5.5. Hence, we can conclude that $r_{ij} \cap D_j(S) \subseteq r_i$, and also $r_i \supseteq \cup_j (r_{ij} \cap D_j(S))$.

Next, we show that r_i satisfies $r_i \subseteq \cup_j (r_{ij} \cap D_j(S))$.

Consider a point p in r_i. By theorem 5.5, p satisfies $m(p, \hat{p_i}) \leq M^*(p)$. Hence, for an ε-Point $\hat{p_j}$ that satisfies $M(p, \hat{p_j}) = M^*(p)$, p satisfies $m(p, \hat{p_i}) \leq M(p, \hat{p_j})$; which implies $p \in r_{ij}$. In addition, p is in $D_j(S)$ from the definition of $D_j(S)$. From these observations, we can conclude that p satisfies $p \in r_{ij} \cap D_j(S)$, and also $r_i \subseteq \cup_j (r_{ij} \cap D_j(S))$.

Finally, we can conclude that $r_i = \cup_j (r_{ij} \cap D_j(S))$ since r_i satisfies both $r_i \supseteq \cup_j (r_{ij} \cap D_j(S))$ and $r_i \subseteq \cup_j (r_{ij} \cap D_j(S))$. □

By theorem 5.10, the construction of r_i is reduced to constructions of $r_{ij} \cap D_j(S)$. To construct r_i, we need only ε-Points $\hat{p_j}$ that satisfy $r_{ij} \cap D_j(S) \neq \emptyset$. In addition, to construct the *boundary* of r_i instead of r_i itself, we need only ε-Points $\hat{p_j}$ that satisfies a condition that ∂r_{ij} passes through the interior of $D_j(S)$.

If a region $D_j(S)$ is bounded, we can construct the boundary of $r_{ij} \cap D_j(S)$ as follows. First, we find all the intersections between ∂r_{ij} and $\partial D_j(S)$ by traversing $\partial D_j(S)$. Then traverse ∂r_{ij} from a point of infinity while ordering these intersections according the order they appear on ∂r_{ij}; we have all the endpoints of fragments of ∂r_{ij} that appear in $D_j(S)$ (see Fig. 5.4). In a case of $\partial r_{ij} \cap \partial D_j(S) = \emptyset$, there is two possibilities of $D_j(S) \subseteq r_{ij}$ or $D_j(S) \cap r_{ij} = \emptyset$; which can be determined by testing a point q in $D_j(S)$ is also in r_{ij} or not.

On the other hand, if $D_j(S)$ is unbounded, we need to distinguish four cases to test whether $\partial r_{ij} \cap \partial D_j(S) = \emptyset$ or not. The four cases are depicted in Fig. 5.5; which are distinguished according to the relative location and the directions of ∂r_{ij} and $\partial D_j(S)$.

In the following theorems, we describe properties of $r_{ij} \cap D_j(S)$ in a case of $\partial r_{ij} \cap \partial D_j(S) = \emptyset$. Based on these theorems, we can construct the boundary of $r_{ij} \cap D_j(S)$ also for an unbounded $D_j(S)$.

Theorem 5.11 *Let C_j be an unbounded connected component of $D_j(S)$. If any point p on ∂C_j satisfies $m(p, \hat{p_i}) < M(p, \hat{p_j})$ and some point q on ∂r_{ij} satisfies $q \notin C_j$, then $C_j \cap r_{ij} = C_j$.*

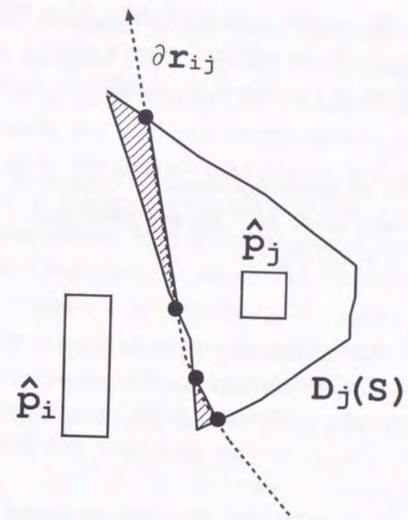

Figure 5.4: Common regions between $D_j(S)$ and r_{ij}

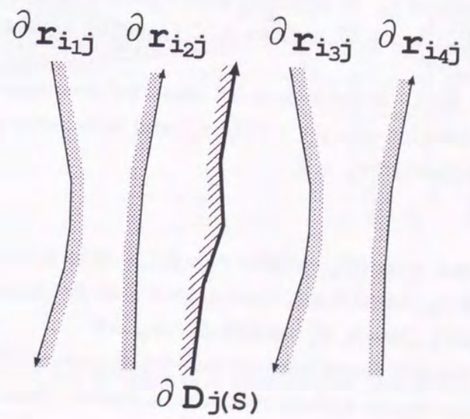

Figure 5.5: Possible locations of ∂r_{ij} when $D_j(S)$ is unbounded

Proof.

Since any point p on ∂C_j satisfies $m(p,\hat{p}_i) < M(p,\hat{p}_j) = M^*(p)$, C_j satisfies $\partial C_j \subseteq r_{ij}$ and $\partial C_j \cap \partial r_{ij} = \emptyset$. In addition, since a point q on ∂r_{ij} is not in C_j, the entire ∂r_{ij} is outside of C_j. Hence, C_j satisfies $C_j \subseteq r_{ij}$; which implies $C_j \cap r_{ij} = C_j$ □

Theorem 5.12 *Let C_j be an unbounded connected component of $D_j(S)$. If any point p on ∂C_j satisfies $m(p,\hat{p}_i) < M(p,\hat{p}_j)$ and some point q on ∂r_{ij} satisfies $q \in C_j$, then $\partial(C_j \cap r_{ij}) = \partial C_j \cup \partial r_{ij}$*

Proof.

Since any point p on ∂C_j satisfies $m(p,\hat{p}_i) < M(p,\hat{p}_j) = M^*(p)$, C_j satisfies $\partial C_j \subseteq r_{ij}$ and $\partial C_j \cap \partial r_{ij} = \emptyset$. In addition, since a point q on ∂r_{ij} is in C_j, the entire ∂r_{ij} is inside of C_j. Hence, C_j satisfies $\partial C_j \subseteq r_{ij}$ and $\partial r_{ij} \subseteq C_j$; which implies $\partial(C_j \cap r_{ij}) = \partial C_j \cup \partial r_{ij}$. □

Theorem 5.13 *Let C_j be an unbounded connected component of $D_j(S)$. If any point p on ∂C_j satisfies $m(p,\hat{p}_i) > M(p,\hat{p}_j)$ and some point q on ∂r_{ij} satisfies $q \in C_j$, then C_j satisfies $C_j \cap r_{ij} = r_{ij}$.*

Proof. Since any point p on ∂C_j satisfies $m(p,\hat{p}_i) > M(p,\hat{p}_j) = M^*(p)$, the entire ∂C_j is outside of r_{ij}. In addition, since a point q on ∂r_{ij} is in C_j, the entire ∂r_{ij} is inside of C_j. Hence, C_j satisfies $r_{ij} \subseteq C_j$; which implies $C_j \cap r_{ij} = r_{ij}$ □

Theorem 5.14 *Let C_j be an unbounded connected component of $D_j(S)$. If any point p on ∂C_j satisfies $m(p,\hat{p}_i) > M(p,\hat{p}_j)$ and some point q on ∂r_{ij} is not in C_j, then C_j satisfies $C_j \cap r_{ij} = \emptyset$.*

Proof.

Since any point p on ∂C_j satisfies $m(p,\hat{p}_i) > M(p,\hat{p}_j) = M^*(p)$, the entire ∂C_j is outside of r_{ij}. In addition, since a point q on ∂r_{ij} is not in C_j, the entire ∂r_{ij} is outside of C_j. Hence, C_j satisfies $C_j \cap r_{ij} = \emptyset$ □

From the discussion above, once we have the diagram $D(S)$, we can construct each weak Voronoi region without redundant ε-Points. However, from a view of computational complexity, we need to care about the complexity of $D(S)$. In the next section, we introduce geometric properties of $D(S)$ and show that it can be constructed in $O(n \log n)$ time if all ε-Points have the same size of input error.

5.2.2 Cluster Voronoi Diagram

The diagram $D(S)$ introduced in this chapter has a close relationship with *cluster Voronoi diagram* described in [8]. Cluster Voronoi diagram was introduced by H. Edelsbrunner *et al.* in [8] as a tool for complete linkage clustering. A set of points is called *cluster*, and let S be a set of n clusters $\hat{p}_1 \ldots, \hat{p}_n$, and let $\delta(p,\hat{q}) = \max\{d(p,q)|q \in \hat{q}\}$ be the distance from a point p to a cluster $\hat{q}$. The *domain* $\mathcal{V}_i(S)$ of cluster $\hat{p}_i$ in S is given by $\mathcal{V}_i(S) = \{p|\delta(p,\hat{p}_i) \leq \delta(p,\hat{p}_j),\ 1 \leq j \leq n\}$. The *cluster Voronoi diagram* $\mathcal{V}_(S)$ is the partition of the plane into domains. As shown in Fig. 5.6, a domain is not necessarily connected and the complexity of a cluster Voronoi diagram is not necessarily $O(n)$. Also, the domain of a cluster may be empty. For example, if $\hat{p}_j$ properly contains $\hat{p}_i$, $\mathcal{V}_j(S)$ is empty since for any point p in the plane satisfies $\delta(p,\hat{p}_i) < \delta(p,\hat{p}_j)$.

Apparently, the diagram $D(S)$ is a kind of cluster Voronoi diagram if an ε-Points is identified with a set of its vertices.

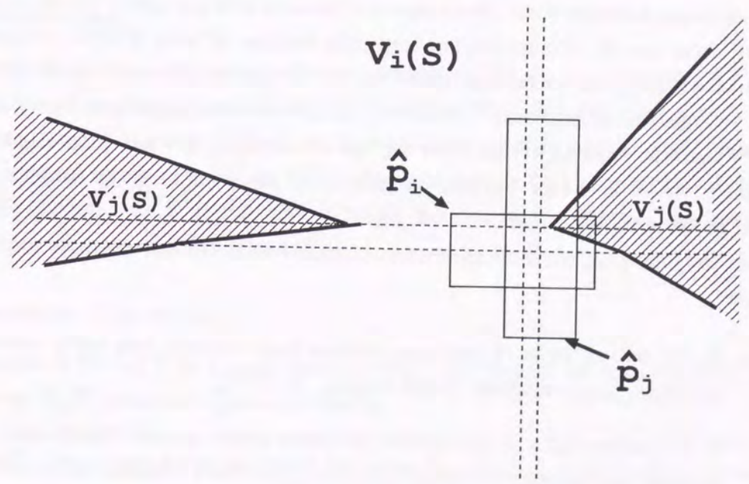

Figure 5.6: An example of a disconnected domain $\mathcal{V}_j(S)$

The literature [8] describes the following theorem 5.15 on a cluster Voronoi diagram.

Theorem 5.15 *Let S be a set of clusters in the plane, and let N be the sum of the cardinalities of the clusters. The cluster Voronoi diagram $\mathcal{V}(S)$ of S satisfies following:*

1. *The number of faces of $\mathcal{V}(S)$ is $O(N^2\alpha(N))$*
2. *If the convex hulls of any two clusters are disjoint, then $\mathcal{V}(S)$ contains at most $|S|$ regions. The number of edges and vertices in this case is $O(N)$.*
3. *$\mathcal{V}(S)$ can be constructed in $O(N^2\alpha(N))$ time.*

In the above statements, $\alpha(n)$ denotes the inverse of Ackermann's function.

The literature [8] also gives an example of $\mathcal{V}(S)$ that has $O(N^2)$ faces. Thus, the quadratic complexity of a cluster Voronoi diagram is essentially inevitable without posing any restriction on clusters.

By theorem 5.15, in general, if the size of ε-Points are small compared with the distances between them, the number of faces of $D(S)$ is $O(n)$. On the other hand, if the size of ε-Points become large, the number of faces of $D(S)$ will grow up to $O(n^2\alpha(n))$. It seems that if the size of ε-Point is sufficiently small, $D(S)$ can be constructed efficiently. However, the construction algorithm introduced in [8] takes at least $O(n^2)$ time for any set of clusters; the sub-quadratic time construction of a cluster Voronoi diagram with low complexity is listed as an open problem in [8].

In the following, we show theorems about a cluster Voronoi diagram as listed below:

1. If the convex hulls of any two clusters have exactly two right common tangents, the complexity of the diagram is $O(N)$.
2. If the convex hulls of clusters are the same under parallel translation, the diagram can be constructed in $O(n \log n)$ time for n clusters with constant number of points.

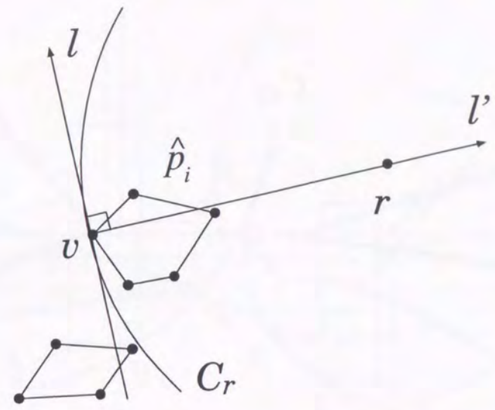

Figure 5.7: A point r is in $\mathcal{V}_i(S)$ if a circle C_r with center r contains only $\hat{p}_i$

Preliminaries

$B(\hat{p}, \hat{q})$ is the bisector between clusters defined as $B(\hat{p}, \hat{q}) = \{p|\delta(p, \hat{p}) = \delta(p, \hat{q})\}$. $R(\hat{p}, \hat{q})$ is the region given by $R(\hat{p}, \hat{q}) = \{p|\delta(p, \hat{p}) \leq \delta(p, \hat{q})\}$. For a set V of points, let $F(V)$ denote the furthest Voronoi diagram of V, and let $F_v(V)$ denote the Voronoi region of $v \in V$ in $F(V)$. For a cluster $\hat{p}$, $T(\hat{p})$ denotes the *skeleton* of the cluster, that is, the geometric graph induced by the edges of the furthest Voronoi diagram of the vertices of $\hat{p}$.

Geometric Properties

Lemma 5.16 *Let l be a right tangent of $\hat{p}_i$. If l covers all clusters in S, the domain $\mathcal{V}_i(S)$ contains a point of infinity.*

Proof. Let v be a vertex at which l touches $\hat{p}_i$, and let l' be the half line that is perpendicular to l and emanating from v. Consider the smallest enclosing circle C_r of $\hat{p}_i$ with center r on l' (see Fig. 5.7). Since all clusters are covered by l, we can choose r so far from v that C_r contains no cluster except $\hat{p}_i$; which implies that r is in $\mathcal{V}_i(S)$. When we move r along l' toward infinity, the proposition holds. □

From lemma 5.16, the following corollary is introduced.

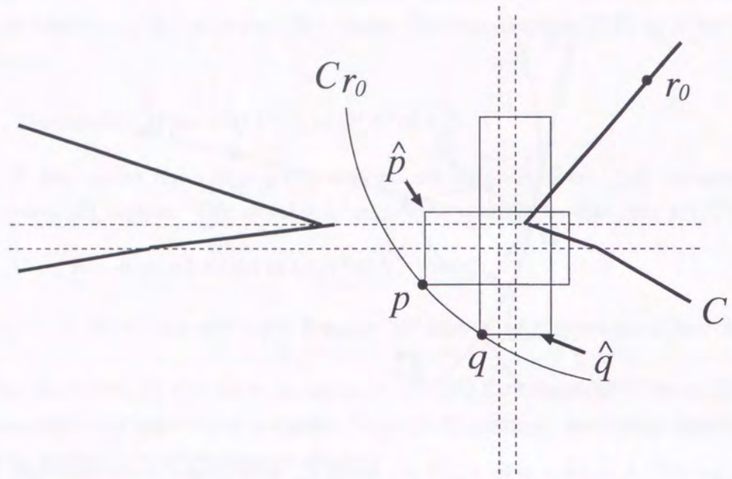

Figure 5.8: The circle C_{r_0} centered at r_0 on $B(\hat{p}, \hat{q})$ touches $\hat{p}$ and $\hat{q}$ at p and q.

Corollary 5.17 *Let l be a common right tangent of $\hat{p}_i$ and $\hat{p}_j$. If l covers all clusters in S, then $\partial\mathcal{V}_i(S)$ and $\partial\mathcal{V}_j(S)$ shares a half line.*

Lemma 5.18 *Let $\hat{p}$ and $\hat{q}$ be clusters, and let C be a connected components of $B(\hat{p}, \hat{q})$. A vertex of a cluster contributes to only one sequence of edges of C.*

Proof. The proof is based on the argument in [8].

Let C_r denote the smallest circle centered at r that contains entire $\hat{p}$ and $\hat{q}$. For a point r_0 on C, let p and q be the vertices at which C_{r_0} touches $\hat{p}$ and $\hat{q}$ respectively (see Fig. 5.8). By a chord $\overline{pq}$, C is separated into two circular arcs.

Imagine that a point r starts from r_0 and goes along C. As long as C_r touches $\hat{p}$ and $\hat{q}$ only at p and q, one circular arc of C_r continues to grow, and the other continues to shrink (see Fig. 5.9). We now prove that while r goes along C, the growing part of C_r continues to grow and the shrinking part continues to shrink.

At some time of the movement, r goes into a new edge of C and a vertex of a cluster appears on the shrinking arc of C_r. Without loss of generality, we assume that a vertex q' of $\hat{q}$ appears on the shrinking arc when r reaches r_1 (see Fig. 5.10). Apparently, q' is the adjacent vertex of q that was included in the

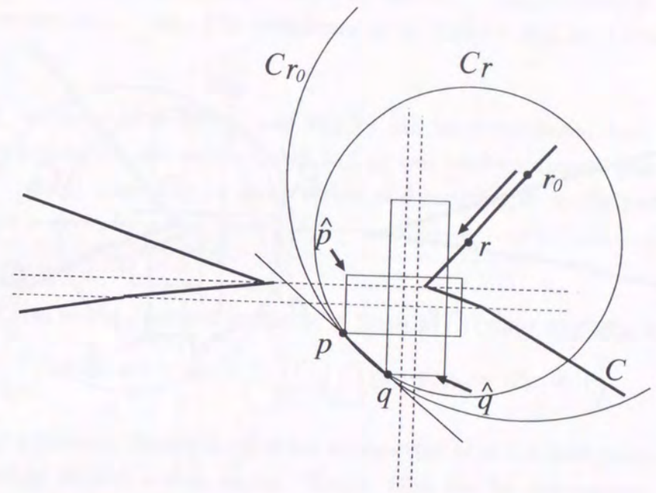

Figure 5.9: When r moves along an edge of C, one part of C_r shrinks and the other grows

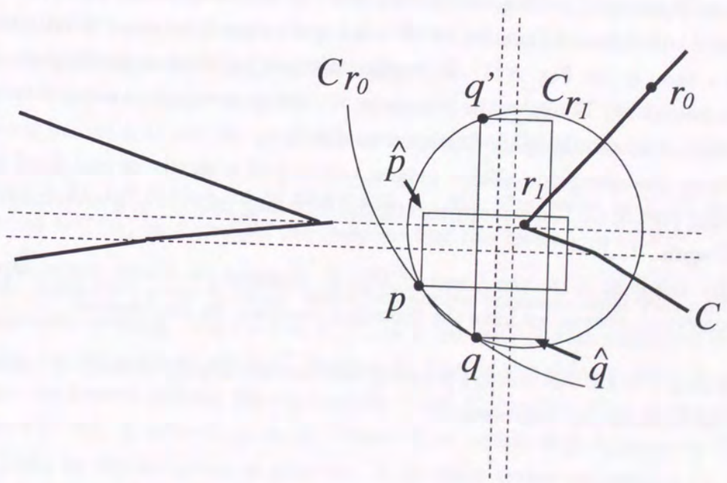

Figure 5.10: A new vertex q' appears on the shrinking arc of C_{r_1}

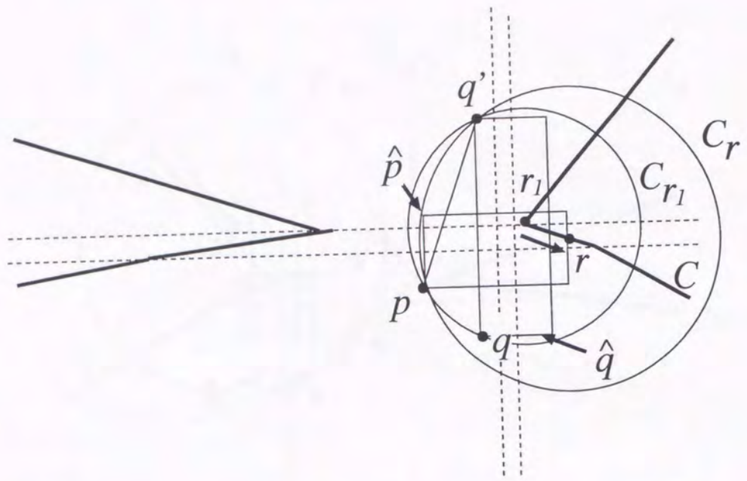

Figure 5.11: The vertex q is contained in the growing part of C_r when r starts from r_1

shrinking part of C_r. Observe that the chord of C_r changes from $\overline{pq}$ to $\overline{pq'}$ or $\overline{qq'}$. When r leaves from r_1, q must be included in C_r since q' becomes strictly farther from r than q (see Fig. 5.11). It implies that q is on the new growing arc when r reaches at r_1. Hence, while r goes on C, the growing arc continues to grow. Similarly, the shrinking arc continues to shrink.

From the discussion above, once a vertex q of a cluster is contained in the growing part of C_r, it never appears on the shrinking part of C_r; q never contribute to C again. □

By theorem 5.18, the bisector $B(\hat{p}, \hat{q})$ contains no closed curve, nor self-intersection. Hence, we have the following corollary on the bisector.

Corollary 5.19 *For clusters $\hat{p}$ and $\hat{q}$, the bisector $B(\hat{p}, \hat{q})$ consists of unbounded curves without self-intersections.*

Theorem 5.20 *Let $\hat{p}$ be a cluster with m vertices $v_1, \ldots, v_m$, and let $\hat{q}$ be a cluster with n vertices $w_1, \ldots, w_n$. The complexity of the bisector $B(\hat{p}, \hat{q})$ is $O(m+n)$.*

Proof.

First, we show that $R(\hat{p}, \hat{q})$ and $R(\hat{q}, \hat{p})$ can be decomposed into at most n and m convex regions respectively. $R(\hat{p}, \hat{q})$ can be decomposed into n parts $R_1(\hat{p}), \ldots, R_n(\hat{p})$ according to which vertex of $\hat{q}$ contributes to the part. Each part $R_i(\hat{p})$ is given by

$$\begin{aligned} & R_i(\hat{p}) \\ = \; & \{p | w_i \textit{ is the farthest vertex of } \hat{q} \textit{ from } p\} \bigcap \{p | \delta(p, \hat{p}) \leq d(p, w_i)\} \\ = \; & \left(\bigcap_{j=1}^{n} \{p | d(p, w_i) \geq d(p, w_j)\} \right) \bigcap \left(\bigcap_{k=1}^{m} \{p | d(p, v_k) \leq d(p, w_i)\} \right) \end{aligned}$$

As the expression shows, $R_i(\hat{p})$ is the intersection of $m+n$ half planes; which is a (possibly empty) convex region. Hence, $R(\hat{p})$ can be decomposed into at most n convex regions. Similarly $R(\hat{q})$ can be decomposed into at most m convex regions. Observe that since the plane is already decomposed into $R(\hat{p})$ and $R(\hat{q})$, the plane is decomposed into $O(m+n)$ convex regions. Since the geometric graph G induced by the decomposition of the plane is a planar graph with $O(m+n)$ faces and each vertex of G has at least three edges, G has at most $O(m+n)$ edges by Euler's relation for planar graphs. Hence, $B(\hat{p}, \hat{q})$ has at most $O(m+n)$ edges since all edges in $B(\hat{p}, \hat{q})$ are the edges of G. □

The skeleton of a cluster gives a fundamental structure of its domain. The following lemma 5.21 and its proof is a restatement of the argument in [8].

Lemma 5.21 *Let S be a set of clusters $\hat{p_1}, \ldots, \hat{p_n}$. Any point in $\mathcal{V}_i(S)$ can be connected to $T(\hat{p_i})$ by a straight line segment that lies entirely in $\mathcal{V}_i(S)$.*

Proof. Let p be a point in $\mathcal{V}_i(S)$, and let C_p be the smallest circle with center p that contains entire $\hat{p_i}$. Since p is in $\mathcal{V}_i(S)$, $\hat{p_i}$ is the only cluster contained in C_p.

Let v be the point at which C_p touches $\hat{p_i}$. Consider a point q starts from p, and moving toward v along the segment $\overline{pv}$. Until C_q touches other vertex of $\hat{p_i}$, C_q contains only $\hat{p_i}$ since C_q is in C_p. When other vertex of $\hat{p_i}$ appears on C_q, q is on $T(\hat{p_i})$ by the definition of skeleton. If no other vertex appears on C_q, the circle vanishes; which implies v is the only point in $\hat{p_i}$. □

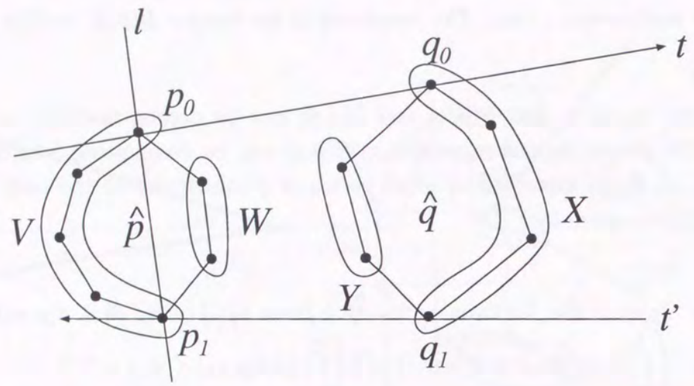

Figure 5.12: Vertices of two clusters are grouped into four sets V, W, X and Y

Theorem 5.22 *Let S be a set of clusters $\hat{p}_1, \ldots, \hat{p}_n$, and let C be a curve of $\partial\mathcal{V}_i(S)$. The edges on C to which a vertex p of $\hat{p}_i$ contributes can be sorted according to their polar angle around p.*

Proof. We show that any distinct points on C can not have the same polar angle around p.

Let q and q' be the points on C that have the same polar angle around p. We assume that q is farther from p than q'. Let C_q be the smallest circle centered at q that contains entire $\hat{p}_i$, and let $C_{q'}$ the circle defined similarly. Since q' is on the segment $\overline{pq}$, $C_{q'}$ is properly contained in C_q. The existence of q' implies that there is another cluster entirely contained in C_q; which is a contradiction. □

Theorem 5.23 *If clusters $\hat{p}$ and $\hat{q}$ has exactly two common right tangents, $T(\hat{p})$ intersects $R(\hat{p}, \hat{q})$ in a connected tree.*

Proof. We split the vertices of $\hat{p}$ into two sets V and W. A vertex of $\hat{p}$ is in V if it is a vertex of $conv(\hat{p} \cup \hat{q})$, or in W if not. Similarly, we split the vertices of $\hat{q}$ into X and Y.

Let t and t' be the common right tangents, and let p_0 and p_1 be vertices of $\hat{p}$ at which t and t' touch $\hat{p}$ respectively. Since $\hat{p}$ is a convex polygon, vertices of V and W are separated by a line l that passes through p_0 and p_1 (see Fig. 5.12).

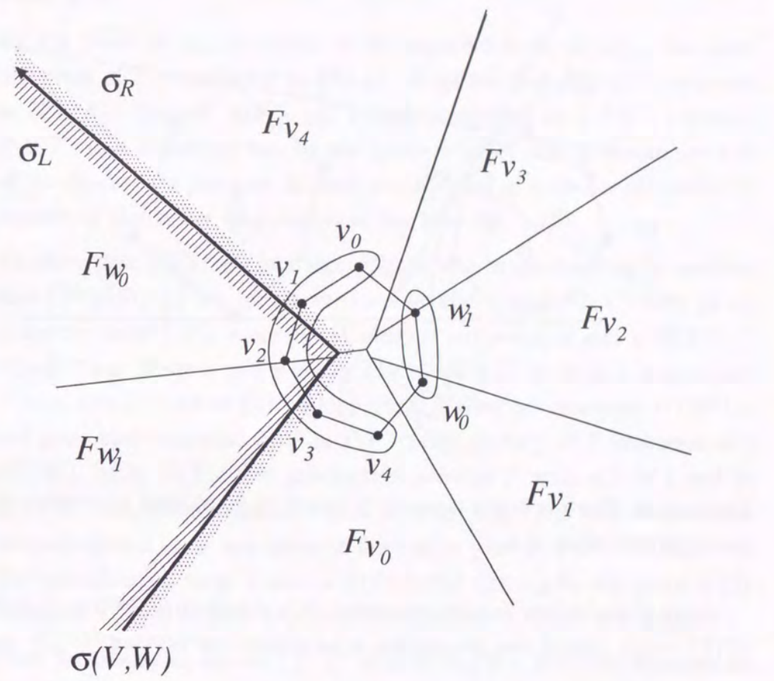

Figure 5.13: For linearly separated sites, the furthest Voronoi diagram is separated by the merge chain

Let $\sigma(V, W)$ be the set of Voronoi edges that are shared by pairs of $F_v(\hat{p})$ and $F_w(\hat{p})$ for $v \in V, w \in W$. Since V and W are linearly separated by l, $\sigma(V, W)$ is a connected chain of segments by the property of a furthest Voronoi diagram, and all Voronoi regions of vertices in V appears in the same side of $\sigma(V, W)$. We call the side of $\sigma(V, W)$ in which Voronoi regions of vertices in W appears *left side* σ_L, and the other side σ_R (see Fig. 5.13). We include $\sigma(V, W)$ in σ_L and σ_R.

To prove the proposition, we show the following:

(1) $T(W) \cap \sigma_L$ intersects $R(\hat{p}, \hat{q})$ in a connected tree

(2) $\sigma(V, W)$ is in $R(\hat{p}, \hat{q})$

(3) $T(V) \cap \sigma_R$ intersects $R(\hat{p}, \hat{q})$ in a connected tree

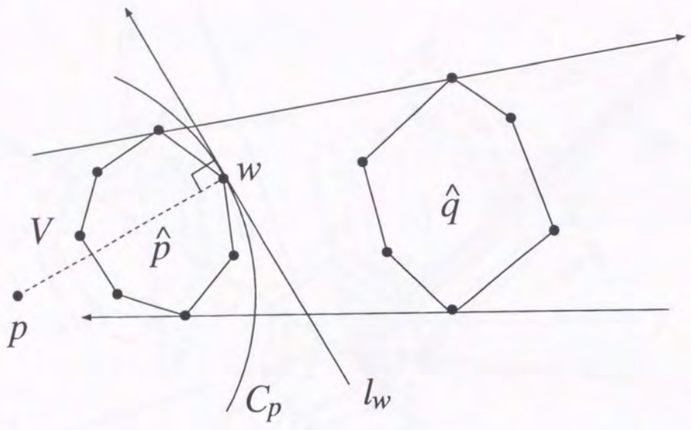

Figure 5.14: Since w is not a vertex of $conv(\hat{p} \cup \hat{q})$, at least one vertex of $\hat{q}$ is in the right half plane of l_w

Since V and W are linearly separated, $T(\hat{p})$ equals to $(T(W) \cap \sigma_L) \cup \sigma(V,W) \cup (T(V) \cap \sigma_R)$. Hence, the proposition is established by proving (1) ,(2), and (3).

- Proof of (1)

  We show that for each vertex w in W, its Voronoi region $F_w(V \cup W)$ in the furthest Voronoi diagram $F(V \cup W)$ is in $R(\hat{p}, \hat{q})$.

  Consider a point p in $F_w(V \cup W)$. Let C_p be the smallest circle centered at p that contains entire $\hat{p}$, and let l_w be the tangent of C_p at w (see Fig. 5.14). We assume that l_w is directed so that C_p is in $H_{\mathcal{L}}(l_w)$.

  Since w is not a vertex of $conv(\hat{p} \cup \hat{q})$, there is at least one vertex of $\hat{q}$ in $H_{\mathcal{R}}^{-}(l_w)$. It means that $\delta(p,\hat{p}) < \delta(p,\hat{q})$; which implies $p \in R(\hat{p},\hat{q})$. The argument is true for all points in $F_w(V \cup W)$. We conclude that $F_w(V \cup W) \subseteq R(\hat{p},\hat{q})$ for any w in W; which implies $T(V \cup W) \cap \sigma_L$ is in $R(\hat{p},\hat{q})$. Since $T(W) \cap \sigma_L$ corresponds to $T(V \cup W) \cap \sigma_L$, $T(W) \cap \sigma_L$ is in $R(\hat{p},\hat{q})$.

- Proof of (2)

  By the proof of (1), $\bigcup_{w \in W} \partial F_w(V \cup W)$ is in $R(\hat{p},\hat{q})$. Since $\sigma(V,W) \subseteq \bigcup_{w \in W} \partial F_w(V \cup W)$, $\sigma(V,W)$ is in $R(\hat{p},\hat{q})$.

- Proof of (3)

  By the proof of (1), no vertex of W contributes to $B(\hat{p},\hat{q})$. Similarly, no vertex of Y contributes to $B(\hat{p},\hat{q})$. It means that $R(\hat{p},\hat{q})$ corresponds to $R(V,X)$. Since V and X are linearly separated by l, $T(V)$ intersects $R(V,X)$ in a connected tree by the theorem in [8] that *if the convex hulls of the clusters are pairwise disjoint, the skeleton of a cluster intersects the domain of the cluster in a connected tree* (see Fig. 5.15).

  We show that $T(V) \cap \sigma_R$ intersects $R(\hat{p},\hat{q})$ in a connected tree by verifying that $T(V) \cap R(\hat{p},\hat{q}) \cap \sigma_R$ is connected and acyclic. Clearly, $T(V) \cap R(\hat{p},\hat{q}) \cap \sigma_R$ is acyclic since $T(V)$ is acyclic. Consider two points x and y in $T(V) \cap R(\hat{p},\hat{q}) \cap \sigma_R$. Since x and y are in $T(V) \cap R(\hat{p},\hat{q})$, there is a unique path P from x to y in a tree $T(V) \cap R(\hat{p},\hat{q})$. If P does not intersect $\sigma(V,W)$, x and y are also connected by P in $T(V) \cap R(\hat{p},\hat{q}) \cap \sigma_R$. If P intersects with $\sigma(V,W)$, let x_0 be the first intersection point of P with $\sigma(V,W)$, and let y_0 be the last. Then, consider a path $x \to x_0 \to y_0 \to y$. It is obvious that the path from x to x_0 and the path from y_0 to y are in $T(V) \cap R(\hat{p},\hat{q}) \cap \sigma_R$. The path from x_0 to y_0 is also in $T(V) \cap R(\hat{p},\hat{q}) \cap \sigma_R$ by the proof of (2). Hence, in $T(V) \cap R(\hat{p},\hat{q}) \cap \sigma_R$, there is a path that connects x and y.

  From the discussion above, $T(V) \cap R(\hat{p},\hat{q}) \cap \sigma_R$ is a tree (For example, see Fig. 5.16). □

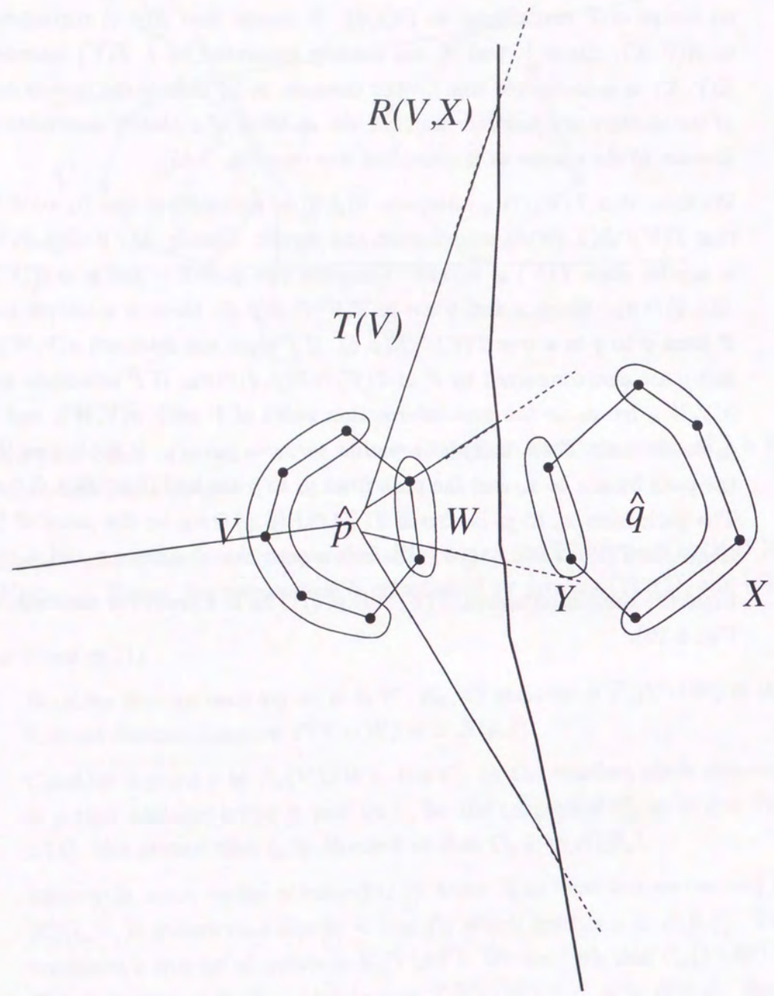

Figure 5.15: Since V and X are linearly separated, $T(V)$ intersects $R(V, X)$ in a connected tree

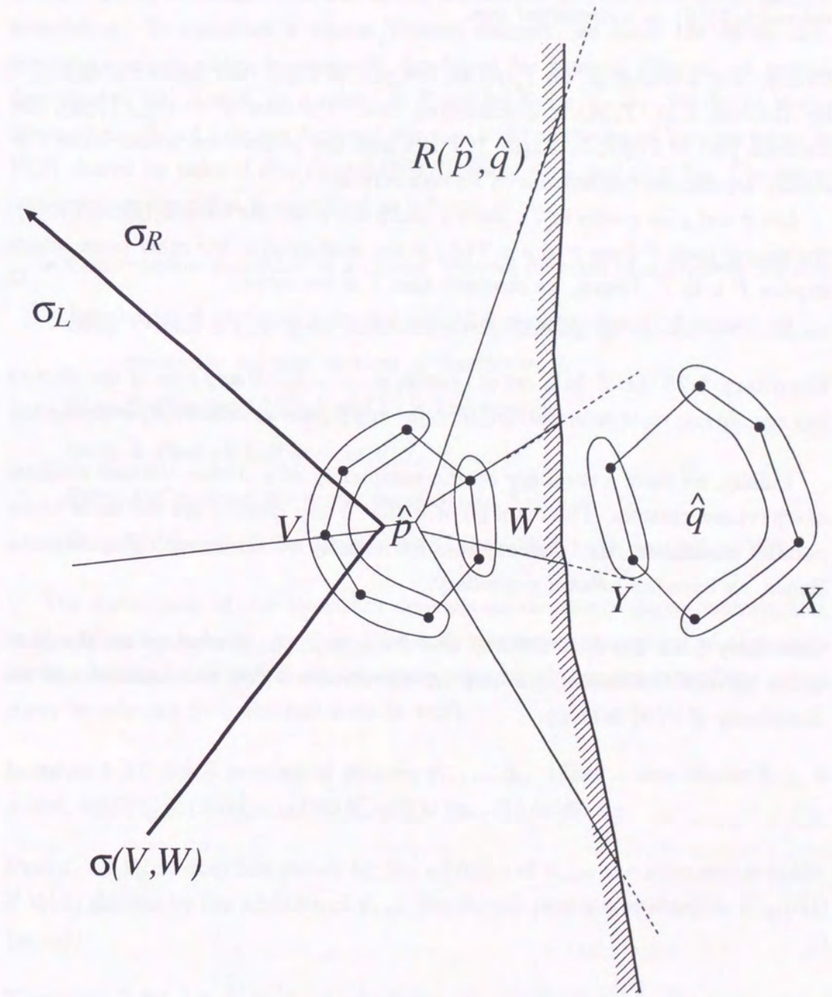

Figure 5.16: Since $\hat{p}$ and $\hat{q}$ has exactly two common right tangents,$T(V \cup W)$ intersects $R(\hat{p}, \hat{q})$ in a connected tree

Theorem 5.24 *Let S be a set of clusters $\hat{p_1}, \ldots, \hat{p_n}$. If any pair of the clusters has exactly two right common tangents, then, for any $\hat{p_i}$, the skeleton $T(\hat{p_i})$ intersects $\mathcal{V}_i(S)$ in a connected tree.*

Proof. For a cluster $\hat{p_i}$, let $T_j(\hat{p_i})$ be the part of $T(\hat{p_i})$ that intersects $R(\hat{p_i}, \hat{p_j})$. By theorem 5.23, $T_j(\hat{p_i})$ is a connected tree. Consider $T = \bigcap_{j=1}^{n} T_j(\hat{p_i})$, the common part of $T_1(\hat{p}), \ldots, T_n(\hat{p})$. If T is a tree, the proposition holds. Since T is clearly acyclic, we concentrate on its connectivity.

Let x and y be points in T. Since x and y are in all the trees $T_1(\hat{p}), \ldots, T_n(\hat{p})$, the unique path P from x to y in $T(\hat{p_i})$ is also contained in the other trees; which implies P is in T. Hence, we conclude that T is connected. □

By lemma 5.21 and theorem 5.24, the following corollary is introduced.

Corollary 5.25 *Let S be a set of clusters $\hat{p_1}, \ldots, \hat{p_n}$. If any pair of the clusters has exactly two right common tangents, for any $\hat{p_i}$, the domain $\mathcal{V}_i(S)$ is connected.*

Finally, we have a corollary on the complexity of a cluster Voronoi diagram of equivalent clusters. The main point is that if any clusters are the same under parallel translation, any pair of them has exactly two common right tangents. Hence, we have the following corollary.

Corollary 5.26 *Let S be a set of clusters $\hat{p_1}, \ldots, \hat{p_n}$. If clusters are the same under parallel translation, for any $\hat{p_i}$, the domain $\mathcal{V}_i(S)$ is connected and the complexity of $\mathcal{V}(S)$ is $O(n)$.*

Construction of a Cluster Voronoi Diagram

In this section, we assume that all clusters are distinct and the same under parallel translation. To construct a cluster Voronoi diagram, we apply the divide-and-conquer scheme, which is originally developed for Voronoi diagram of points, described in [31]. Let S_1 be a subset of S and let S_2 be $S - S_1$. We define *merge curve* $M(S_1, S_2)$ of a cluster Voronoi diagram $\mathcal{V}(S)$ as the set of Voronoi edges in $\mathcal{V}(S)$ shared by pairs of $\partial\mathcal{V}_i(S)$ and $\partial\mathcal{V}_j(S)$, for $p_i \in S_L$ and $p_j \in S_R$. The entire construction algorithm is described as follows.

- Construction algorithm of a cluster Voronoi diagram of equivalent clusters

  Step 1 *Split S into equal sized subsets S_L and S_R by median x-coordinate among the leftmost vertices of clusters in S.*

  Step 2 *Construct $\mathcal{V}(S_L)$ and $\mathcal{V}(S_R)$ recursively.*

  Step 3 *Find all half lines in $\mathcal{V}(S)$.*

  Step 4 *Construct the merge curve $M(S_L, S_R)$.*

  Step 5 *Discard all the edges of $\mathcal{V}(S_L)$ and $\mathcal{V}(S_R)$ that are not in $\mathcal{V}(S)$.*

The correctness of the algorithm depends on the merge curve construction. In the following lemmas, we show that each connected component of the merge curve contains a half line in $\mathcal{V}(S)$; we can traverse all components of the merge curve by starting from the half lines in $\mathcal{V}(S)$.

Lemma 5.27 *Let S be a set of clusters $\hat{p_1}, \ldots, \hat{p_n}$. When a new cluster $\hat{p}_{n+1}$ is added, $\mathcal{V}_i(S) \subseteq \mathcal{V}_i(S \cup \{p_{n+1}\}) \cup \mathcal{V}_{n+1}(S \cup \{p_{n+1}\})$ holds.*

Proof. If $\mathcal{V}_i(S)$ does not shrink by the addition of $\hat{p}_{n+1}$, the proposition holds. If $\mathcal{V}_i(S)$ shrinks by the addition of $\hat{p}_{n+1}$, the shrunk part is contained in $\mathcal{V}_{n+1}(S \cup \{p_{n+1}\})$. □

Theorem 5.28 *Let S be a set of clusters $\hat{p_1}, \ldots, \hat{p_n}$ that are the same under parallel translation, and sorted according to their leftmost x-coordinates. For $S_L = \{\hat{p_1}, \ldots, \hat{p}_{\lfloor n/2 \rfloor}\}$ and $S_R = \{\hat{p}_{\lfloor n/2 \rfloor + 1}, \ldots, \hat{p_n}\}$, no closed curve is in the merge curve $M(S_L, S_R)$.*

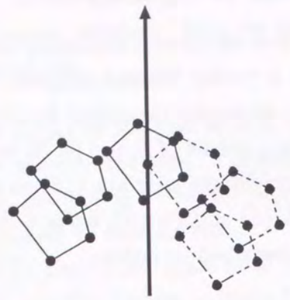

Figure 5.17: A right tangent of a cluster in S_R covers all clusters in S_L

Proof. Consider a case that a closed curve C of $M(S_L, S_R)$ contains parts of domains of clusters in S_R.

Since all domains are connected by corollary 5.26, it implies that the inside of C is partitioned into domains of clusters in S_R. Consider a cluster $\hat{p} \in S_R$ whose domain is in C. By lemma 5.27, the domain of $\hat{p}$ in $\mathcal{V}(S_L \cup \{\hat{p}\})$ is bounded by C and surrounded by domains of clusters in S_L.

On the other hand, the domain of $\hat{p}$ in $\mathcal{V}(S_L \cup \{\hat{p}\})$ is unbounded by lemma 5.16 since the leftmost vertex of $\hat{p}$ is to the right of the leftmost vertices of clusters in S_L; the right tangent of $\hat{p}$ parallel to y-axis covers all clusters in S_L (see Fig. 5.17). This implies a contradiction. Hence, there is no closed curve that contains parts of domains of clusters in S_R.

By a similar argument, we can prove that there is no closed curve C' of $M(S_L, S_R)$ that contains regions of clusters in S_L since all clusters are the same under parallel translation and sorted according to leftmost vertices; the rightmost vertex of a cluster in S_R is to the right of the rightmost vertices of clusters in S_L. □

Find All Half Lines in a Cluster Voronoi Diagram

By corollary 5.17, we can find all half lines in $\mathcal{V}(S)$ by the same method described in section 3.3.3 that finds all boundary lines of $R_{in}(S)$ for angular equivalent polygons. Hence, we can find all half lines in $\mathcal{V}(S)$ in $O(n \log n)$ time.

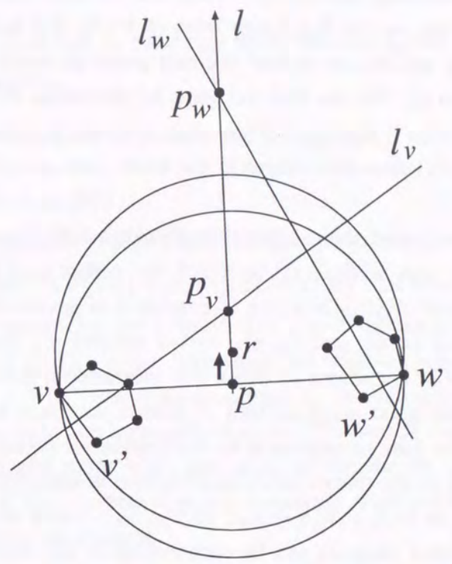

Figure 5.18: A new edge appears on the merge curve at the closer of p_v and p_w

Construction of a Merge Curve

Consider we are on a point p of $M(S_L, S_R)$, and going along a line l emanating from p (remember that we already have half lines in $M(S_L, S_R)$). We can find edges of $M(S_L, S_R)$ as follows.

Assume that p is on the bisector $B(\hat{p}_i, \hat{p}_j)$ of $\hat{p}_i$ in S_L and $\hat{p}_j$ in S_R, and let v be the farthest vertex of $\hat{p}_i$ from p, and let w be the farthest vertex of $\hat{p}_j$ from p (see Fig. 5.18).

Consider a point r that starts from p and goes along $M(S_L, S_R)$. By the proof of lemma 5.18, a new edge appears when the shrinking part of C_r touches a vertex of $\hat{p}_i$ or $\hat{p}_j$. The point at which the new edges appears can be calculated as follows.

Let v' be the vertex of $\hat{p}_i$ that is adjacent to v in the shrinking part of C_r, and let w' be the vertex of $\hat{p}_j$ that is adjacent to w and in the shrinking part of C_r. Consider the perpendicular bisectors l_v of the segment $\overline{vv'}$, and l_w of the segment $\overline{ww'}$. Let p_v and p_w be the points at which l_v and l_w intersect l respectively.

Apparently, the shrinking part of C_r touches v' or w' when r reaches at the closer of p_v and p_w. Hence, we can find a new edge of $M(S_L, S_R)$ in constant time.

The remaining problem is to find the exit point at which $M(S_L, S_R)$ leaves the domain of $\hat{p}_i$ or $\hat{p}_j$. We can find the point by traversing $\partial V_i(S_L)$ and $\partial V_j(S_R)$ every time a new edge is found on $M(S_L, S_R)$; however, it takes $O(n^2)$ time since $\partial V_i(S_L)$ and $\partial V_j(S_R)$ have $O(n)$ edges in the worst case and $M(S_L, S_R)$ can have $O(n)$ edges.

We can efficiently find the exit point by theorem 5.22. Consider a vertex v of $\hat{p}_i$. Let C_1 be the part of $B(\hat{p}_i, \hat{p}_j)$ to which the vertex v of $\hat{p}_i$ contributes, and let C_2 be the part of $\partial V_i(S_L)$ to which the vertex v of $\hat{p}_i$ contributes. By theorem 5.22, the segments on C_1 and C_2 are sorted around v. To find intersections between C_1 and C_2, it suffices to check the intersection between the segments that have the same polar angle around v. Hence, intersections between C_1 and C_2 can be found in time proportional to the number of segments in C_1 and C_2.

The time used in the merge curve construction is bounded by the sum of the number of edges in $\mathcal{V}(S_L)$, $\mathcal{V}(S_R)$ and $M(S_L, S_R)$; which is $O(n)$. Hence, the entire cluster Voronoi diagram can be constructed in $O(n \log n)$ time.

5.2.3 Construction of Weak Voronoi Regions

In this section, we show how to construct weak Voronoi regions with the aid of a cluster Voronoi diagram.

- Construction algorithm of a weak Voronoi region r_i

  Step1 Locate $\hat{p}_i$ in $D(S)$

  Find a region R of $D(S)$ in which the center p of $\hat{p}_i$ locates.

  Step2 Traverse $D(S)$ and visit regions of $D(S)$ that intersect r_i

  In the traverse, we use breadth first search on the adjacency graph of regions of $D(S)$. While traversing, we examine whether the visiting region r of $D(S)$ shares a region with r_i. In that case, we compute the components of the boundary of $r \cap r_i$. The search starts from R found in Step 1 and stops when any adjacent regions do not share a region with r_i. If the traverse does not terminate after visiting $O(n)$ edges of $D(S)$, stop the traverse.

  Step3 Construct ∂r_i

  If the traverse was stopped in Step 2, construct ∂r_i by computing $\cap_{j \neq i} r_{ij}$. Otherwise, connect the components of ∂r_i computed in Step 2.

In step 1, we can use an existing *point location* method such as in [31]. In step 2, we construct an adjacency graph whose vertices correspond with the regions of $D(S)$ which share a region with r_i. Vertices are connected by an edge only if their corresponding regions are adjacent in $D(S)$.

By lemma 5.6, the region R found in Step 1 is ensured to share a region with r_i. In Step 2, by traversing the boundary of r, we can check whether it shares a region with r_i and ∂r_i passes through the inside of it. If ∂r_i is turned out to pass through the inside of a region of $D_j(S)$, we store the intersection between ∂r_{ij} and $\partial D_j(S)$ in the data structure corresponding to the edge of $\partial D_j(S)$ at which they intersect. Also, for each intersection between ∂r_i and $\partial D_j(S)$, we store the information about which component of ∂r_{ij} is connected to it. The components of ∂r_{ij} computed in Step 2 correspond to the components of ∂r_i.

In Step 3, we construct ∂r_i by connecting the components of it found in Step 2. We can traverse the components in the order they appear on ∂r_i since

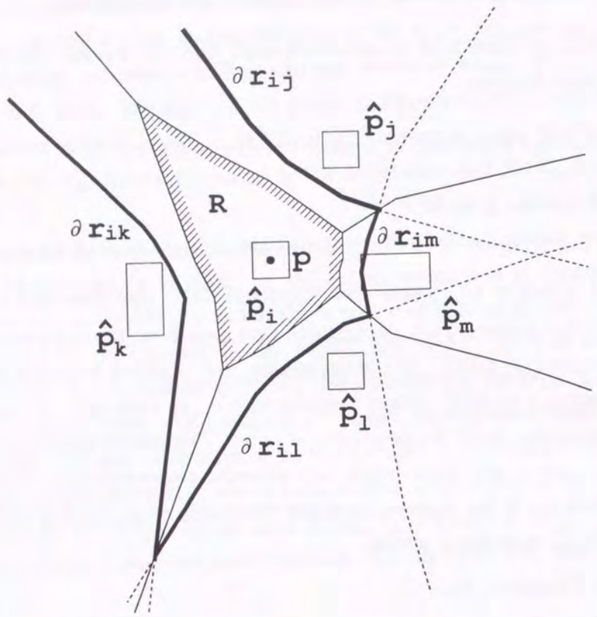

Figure 5.19: A weak Voronoi region r_i is constructed from a diagram $D(S)$

components adjacent on ∂r_i share the same endpoint in $D(S)$; which is recorded in Step 2.

Since a weak Voronoi region is simply connected by theorem 5.8, the breadth first search from R finds all the regions of $D(S)$ that contribute to r_i; which ensures that the entire ∂r_i is constructed. For the example in Fig. 5.19, we can construct ∂r_i as follows:

(1) locate the center p of $\hat{p}_i$, then we find the region R.

(2) Start breadth first search from R, then we find the components ∂r_{ij}, ∂r_{ik}, ∂r_{il}, and ∂r_{im}.

(3) Connect these components, then we get ∂r_i.

We analyze the time complexity to construct n weak Voronoi regions $r_1, \ldots, r_n$. We denote the complexity of $D(S)$ as N.

In Step 1, one point location query can be processed in $O(\log N)$ time with $O(N \log N)$ pre-process time. Since n is not more than N, $O(N \log N + n \log N) = O(N \log N)$ time is used in Step 1.

In Step 2, the computation time depends on the input. First consider a worst case that some ε-Point $\hat{p}_i$ is so large that contains most of other ε-Points. In this case, we need to traverse most of $D(S)$ to construct r_i; which results in $O(N)$ time. Without cut off in the traverse, we need overall $O(nN)$ time. To prevent this, we cut off the traverse after visiting $O(n)$ edges in $D(S)$; which results in overall $O(n^2)$ time. On the other hand, consider a case that the size of each ε-Points is sufficiently small compared with their relative distances. In the case, we need to traverse only adjacent regions of $D_r(\hat{p}_i)$. In addition, since ε-Points can be considered to be distributed uniformly in the plane, each region of $D(S)$ has at most constant number of adjacent regions on average. Hence, each segment of $D(S)$ is traversed at most constant times; which results in overall $O(n)$ time in Step 2.

In Step 3, we traverse the components of ∂r_i; the time to construct ∂r_i depends on the complexity of ∂r_i itself. By a similar discussion in Step 2, it takes $O(n)$ time in best case. By theorem [9], the complexity of ∂r_i is $O(n \log^* n)$; which implies worst case $O(n^2 \log^* n)$ time. On the other hand, if all traverses were cut off in Step 2, we need $O(n^2 \log^* n \log n)$ time.

From the discussion above, once we have the diagram $D(S)$, we can construct all weak Voronoi regions in $O(N \log N)$ time (best case) or in $O(n^2 \log^* n \log n)$ time (worst case). The best case occurs when ε-Points are distributed uniformly.

Let us summarize our method. $D(S)$ can be constructed in $O(n^2 \alpha(n))$ by the method in [8]. For a special case that all ε-Points have the same size of input error, $D(S)$ can be constructed in $O(n \log n)$ time from the discussion in section 5.2.2. Hence, the overall construction time including the construction of $D(S)$ is:

1. If all ε-Points have the same size of input error,

   $O(n \log n)$ time (best case) or $O(n^2 \log^* n \log n)$ time (worst case)

2. Otherwise,

   $O(n^2 \alpha(n))$ time (best case) or $O(n^2 \log^* n \log n)$ time (worst case)

5.3 Strong Voronoi Regions

In this section, we introduce geometric properties of a strong Voronoi region R_i. Then we describe a construction algorithm of strong Voronoi regions.

5.3.1 Geometric Properties of a Strong Voronoi Region

First, consider an elementary strong Voronoi region R_{ij} determined by two ε-Points $\hat{p}_i$ and $\hat{p}_j$.

Definition 5.29 (Strong Voronoi Region R_{ij})

$$R_{ij} = \{p | M(p, \hat{p}_i) \leq m(p, \hat{p}_j)\}$$

A strong Voronoi region R_{ij} is the complement of r_{ji}; which is convex as shown in Fig. 5.1. By definition 5.3, R_i is given by $\cap_{j \neq i} R_{ij}$. Since R_{ij} is convex, R_i is a simple connected region.

Next, we describe a condition that a point is contained in R_i.

Theorem 5.30 *Let S be a set of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$, and let $m_i^*(p)$ be the piecewise minimum function $\min\{m(p, \hat{p}_j) | j \neq i, 1 \leq j \leq n\}$. The necessary and sufficient condition for a point p in the plane to be contained in R_i is $M(p, \hat{p}_i) \leq m_i^*(p)$.*

Proof. By the definition 5.29 of R_{ij}, a point p is in R_{ij} iff $M(p, \hat{p}_i) \leq m(p, \hat{p}_j)$. Since R_i is given by $R_i = \cap_{j \neq i} R_{ij}$, the proposition holds. □

From the above theorem, we can introduce a simple sufficient condition that a strong Voronoi region becomes empty.

Theorem 5.31 *Let S be a set of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$. If $\hat{p}_i$ and $\hat{p}_j$ intersect, both R_i and R_j is empty.*

Proof. Since $\hat{p}_i$ and $\hat{p}_j$ intersect, for any point p in the plane, $m(p, \hat{p}_j) < M(p, \hat{p}_i)$ holds. Hence by theorem 5.30, no point is contained in R_i. By the same argument, R_j is also empty. □

Based on the same idea for weak Voronoi regions, we consider a partition of the plane according to m_i^*. This partition corresponds to the Voronoi diagram of a set of *polygons* $S \setminus \{\hat{p}_i\}$ if we see an ε-Point as a polygon.

Next, we show a relationship between a strong Voronoi region and the diagram.

Theorem 5.32 (Components of a Strong Voronoi Region R_i)

Let S be a set of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$, and let $V(S)$ be the Voronoi diagram of a set S of polygons, and let $V_j(S)$ be the Voronoi region of $\hat{p}_j$ in $V(S)$. Then, R_i is expressed as:

$$R_i = \cup_j \left(R_{ij} \cap V_j(S \setminus \{\hat{p}_i\})\right)$$

Proof.

First, we show $R_i \supseteq \cup_j \left(R_{ij} \cap V_j(S \setminus \{\hat{p}_i\})\right)$ by proving $R_i \supseteq R_{ij} \cap V_j(S \setminus \{\hat{p}_i\})$ holds for any j.

By the definition of a Voronoi region, any point p in $V_j(S \setminus \{\hat{p}_i\})$ satisfies $m(p, \hat{p}_j) = m_i^*(p)$. Hence, any point p in $R_{ij} \cap V_j(S \setminus \{\hat{p}_i\})$ is in R_i by theorem 5.30; which implies $R_{ij} \cap V_j(S \setminus \{\hat{p}_i\}) \subseteq R_i$. Since the same argument applies for any j, we can conclude that $R_i \supseteq R_{ij} \cap V_j(S \setminus \{\hat{p}_i\})$.

Next, we show $R_i \subseteq \cup_j \left(R_{ij} \cap V_j(S \setminus \{\hat{p}_i\})\right)$ is true.

Consider an arbitrary point p in R_i. By theorem 5.30, p satisfies $M(p, \hat{p}_i) \leq m_i^*(p)$. In addition, for $\hat{p}_j$ that satisfies $m(p, \hat{p}_j) = m_i^*(p)$, $M(p, \hat{p}_i) \leq m(p, \hat{p}_j)$ holds; which implies $p \in R_{ij}$. On the other hand, by the definition of $V_j(S \setminus \{\hat{p}_i\})$, p is in $V_j(S \setminus \{\hat{p}_i\})$. Since there is an ε-Point $\hat{p} \in S$ that satisfies $m(p, \hat{p}) = m_i^*(p)$ for any point p in R_i, we can conclude that $R_i \subseteq \cup_j (R_{ij} \cap V_j(S \setminus \{\hat{p}_i\}))$.

From the discussion above, we can conclude $R_i = \cup_j (R_{ij} \cap V_j(S \setminus \{\hat{p}_i\}))$ is true. □

By theorem 5.32, once we have the Voronoi diagram $V(S \setminus \{\hat{p}_i\})$, we can construct the boundary of R_i by a similar method in the weak Voronoi region construction. However, we need a Voronoi diagram for each i; it is too slow to construct n Voronoi diagrams.

The purpose we use the Voronoi diagram $V(S \setminus \{\hat{p}_i\})$ is to exclude redundant ε-Points that does not contribute to R_i; which leads to an efficient construction of ∂R_i. In fact, we do not need the entire diagram for the purpose; we show it in the next theorem 5.33.

Theorem 5.33 (The Maximum Size of a Strong Voronoi Region)

Let S be a set of ε-Points $\hat{p}_1, \ldots, \hat{p}_n$, and R_i be the strong Voronoi region of $\hat{p}_i$, and $V(S)$ be the Voronoi diagram of a set S of polygons. For the Voronoi region $V_i(S)$ of $\hat{p}_i$, R_i satisfies $R_i \subseteq V_i(S)$.

Proof. Consider a point p in R_i. By the definition of a strong Voronoi region, p satisfies $d(p, f_A(\hat{p}_i)) \leq d(p, f_A(\hat{p}_j))$ $(1 \leq j \leq n)$ for any possible arrangement A in $A(S)$. Since there is a possible arrangement B that satisfies $d(p, f_B(\hat{p}_j)) = m(p, \hat{p}_j)$ $(1 \leq j \leq n)$, $m(p, \hat{p}_i) = \min\{m(p, \hat{p}_j) | 1 \leq j \leq n\}$ holds; which implies $p \in V_i(S)$ by the definition of Voronoi region. Since the same argument applies for any point p in R_i, we can conclude that $R_i \subseteq V_i(S)$ □

Based on the theorem, to construct R_i, we need only the part of $V(S \setminus \{\hat{p}_i\})$ that overlaps $V_i(S)$. Moreover, by the property of Voronoi diagram, $\hat{p}_j$ satisfies $V_i(S) \cap V_j(S \setminus \{\hat{p}_i\}) \neq \emptyset$ only if its Voronoi region $V_j(S)$ is adjacent to $V_i(S)$. Hence, to find all the ε-Points that contributes to R_i, it is sufficient to find all the adjacent regions of $V_i(S)$ by traversing $\partial V_i(S)$.

5.3.2 Construction of Strong Voronoi Regions

From the discussion in the previous section, strong Voronoi regions can be constructed as follows.

- Construction algorithm for strong Voronoi regions

  Step1 For a set S of ε-Points, find intersections between ε-Points, and remove ones that intersects others.

  Step2 Consider an ε-Point as an polygon and construct the Voronoi diagram $V(S)$ of polygons.

  Step3 For each ε-Points $\hat{p}_1, \ldots, \hat{p}_n$, do the following:

  Step3-1 In $V(S)$, find all adjacent regions $V_{i_1}, \ldots, V_{i_{m_i}}$ of $V_i(S)$.

  Step3-2 Construct the boundary of $\cap_{j=1}^{m_i} R_{ii_j}$.

In Step 1, the intersection can be detected in $O(n \log n + K)$ time (K is the number of intersection) [6]. By theorem 5.31, we know that strong Voronoi regions of removed ε-Points are empty.

In Step 2, the existing algorithm for Voronoi diagram of segments [35] can be used. In the algorithm, a polygon is treated as a set of segments. $V(S)$ can be constructed in $O(n \log n)$ time.

In Step 3-1, we traverse $\partial V_i(S)$ in time proportional to the number m_i of adjacent regions of $V_i(S)$. In Step 3-2, we construct ∂R_i by computing the intersection of m_i convex regions. Since ∂R_i is a curve whose component contains

parabolas, we need to care about the time complexity of construction of ∂R_i. Observe that the boundary of elementary strong Voronoi regions R_{ij} and R_{ik} intersect at most constant times. Hence, by theorem 2 in [9], the complexity of the boundary of the intersection $\cap_{j=1}^{k} R_{ij}$ of k curves is $O(k \log^* k)$. This ensures that the intersection between $\cap_{j=1}^{k} R_{ij}$ and $\cap_{j=k+1}^{2k} R_{ij}$ can be constructed in $O(k \log^* k)$ time by a method in [7]; which implies that ∂R_i can be recursively constructed in $O(m_i \log^* m_i \log m_i)$ time. From the discussion, the time used in Step 2 is $O(m_i \log^* m_i \log m_i)$. By theorem 3 in [20], the number of edges in the adjacency graph of $V(S)$ is $O(n)$; $\sum_{i=1}^{n} O(m_i \log^* m_i \log m_i) = O(n \log^* n \log n)$.

Finally, we can conclude that all strong Voronoi regions can be constructed in $O(n \log^* n \log n)$ time.

The value of K ranges from $O(n)$ to $O(n^2)$ depending on the arrangement of ε-Points. In a best case that ε-Points are disjoint, $K = O(n)$;strong Voronoi regions can be constructed in $O(n \log^* n \log n)$ time. In a worst case that any pair of ε-Points intersects, $K = O(n^2)$; strong Voronoi regions can be constructed in $O(n^2)$ time.

5.4 Applications

- Nearest neighbor search with an imprecise input

  We can use strong Voronoi regions to verify the result of a nearest neighbor search in a possible arrangement. Let $f_A(\hat{p}_i)$ be the nearest neighbor of p in a possible arrangement of A. If p is contained in R_i, the result is valid; the only nearest neighbor of p is $\hat{p}_i$. On the other hand, if p is not in R_i, we can find all the candidates of nearest neighbor of p by finding all weak Voronoi regions that contains p. To find such regions, we can use a method for *point enclosure* problem in [19].

- Verification of the topological structure of a Voronoi diagram

  Once we have weak Voronoi regions, we can check whether the topological structure of the Voronoi diagram of a possible arrangement is stable against input errors. To check the stability of a Voronoi vertex v of a Voronoi diagram V, it is enough to count the number of weak Voronoi regions that contain v. We can prove that if there is only three weak Voronoi regions that contain v, the Voronoi vertex v always exists and its adjacent Voronoi

regions are given by the same set of ε-Points in all possible arrangements; if there are more than three weak Voronoi regions that contain v, v will degenerate to other Voronoi vertex in some possible arrangement. The argument is formalized as follows:

Theorem 5.34 *Let S be a set of ε-Points $\hat{p_1}, \ldots, \hat{p_n}$. If r_i,r_j and r_k are the only weak Voronoi regions that intersects with the region $r_i \cap r_j \cap r_k$, then in the Voronoi diagram of any possible arrangement of A, there is a Voronoi vertex whose nearest sites are only $f_A(\hat{p_i}), f_A(\hat{p_j})$ and $f_A(\hat{p_k})$.*

Proof.

First, we show that the intersection R of r_i, r_j, and r_k corresponds to $\cap r_{ji} \cap r_{ik} \cap r_{ki} \cap r_{jk} \cap r_{kj}$. Since the only weak Voronoi regions that contain a point p in R are r_i,r_j and r_k, p satisfies

$$\left\{ \begin{array}{l} M(p,\hat{p_i}) \\ M(p,\hat{p_j}) \\ M(p,\hat{p_k}) \end{array} \right\} < m(p,p_l) \leq M(p,p_l) \ , l \neq i,j,k$$

Hence, the ε-Point that gives $M^*(p)$ for p in R is one of the $\hat{p_i}$,$\hat{p_j}$ and $\hat{p_k}$. By theorem 5.10, the part of r_i, r_j and r_k around R is determined only by $\hat{p_i}$,$\hat{p_j}$ and $\hat{p_k}$; which implies $R = (r_{ij} \cap r_{ik}) \cap (r_{ji} \cap r_{jk}) \cap (r_{ki} \cap r_{kj})$.

Next, let A be a possible arrangement, and let p_{ijk} be the equidistant point of $f_A(\hat{p_i})$,$f_A(\hat{p_j})$ and $f_A(\hat{p_k})$. Observe that in the arrangement A, p_{ijk} satisfies $p_{ijk} \in r_{ij} \cap r_{ji} \cap r_{ik} \cap r_{ki} \cap r_{jk} \cap r_{kj}$; which implies $p_{ijk} \in R$. Since only r_i,r_j and r_k are the weak Voronoi regions that intersect with R, only $f_A(\hat{p_i}), f_A(\hat{p_j})$ and $f_A(\hat{p_k})$ can be the nearest sites of p_{ijk}. Since a Voronoi vertex has at least three nearest sites, $f_A(\hat{p_i}), f_A(\hat{p_j})$ and $f_A(\hat{p_k})$ are the nearest sites of p_{ijk} in A, and no other site can be the nearest site of p_{ijk}. The above argument is true for any possible arrangement; the proposition is proved. □

Under the existence of input error, we can not determine the coordinates of each Voronoi vertex exactly, however, theorem 5.34 gives a sufficient condition to test whether a Voronoi vertex do exist and its nearest sites can be determined exactly. Once we have weak Voronoi regions, we can verify which part of a Voronoi diagram is unstable or stable against input error.

5.5 Conclusion

In this chapter, we consider a Voronoi diagram with imprecise input and propose a method to construct Voronoi regions which show the effect of input error. We define two types of Voronoi regions: *weak Voronoi region* and *strong Voronoi region*, and introduced geometric properties of them. Especially, we show that cluster Voronoi diagram relates to weak Voronoi regions and that Voronoi diagram for polygons relates to strong Voronoi regions.

Next, we consider a construction algorithm for weak Voronoi regions. For n ε-Points, let $f(n)$ be the time to construct the cluster Voronoi diagram. Then weak Voronoi regions can be constructed in $O(f(n)+n \log n)$ time (in best case) or in $O(f(n)+n^2 \log^* n \log n)$ time (in worst case). A best case occurs when ε-Points are distributed uniformly in the plane, and the size of them is sufficiently small compared to their distances. A worst case occurs when ε-Points with large size gather in a small area. By theorem in [8], $f(n)$ is $O(n^2\alpha(n))$; we can construct weak Voronoi regions in $O(n^2\alpha(n))$ time (in best case) or in $O(n^2 \log^* n \log n)$ time (in worst case). For a special case that all ε-Points have the same size of input error, we showed that weak Voronoi regions can be constructed in $O(n \log n)$ time (best case) or in $O(n^2 \log^* n \log n)$ time (worst case).

Finally, we consider a construction of strong Voronoi regions. We showed that all strong Voronoi regions can be constructed in $O(n \log^* n \log n + K)$ time. In the previous statement, K denotes the number of intersections between ε-Points. Since K ranges from $O(1)$ to $O(n^2)$, they can be constructed in $O(n \log^* n \log n)$ time (best case) or in $O(n^2)$ time (worst case). A best case occurs when no ε-Point intersects each other, and a worst case occurs when any pair of ε-Points intersects.

We also introduce applications of our method: nearest neighbor search and verification of a topological structure of a Voronoi diagram.

Chapter 6

Conclusions

In the dissertation, we considered geometric problems with imprecise input. Our primary concern is to compute accuracy guaranteed results from such input data. Based on the idea that the accuracy of an output is guaranteed by the trace of the boundaries of all possible output figures, we developed geometric algorithms for three geometric problems: convex hull, diameter of a set of points and Voronoi diagram.

Our results are summarized as follows:

- convex hull

  For n ε-Points in the plane, the inner and the outer convex hull can be constructed in $O(n \log n)$ time. Also we show that if a non-empty inner convex hull is given, we can update the inner convex hull in $O(\log k) \sim O(k)$ time (k is the number of already processed ε-Points); which leads to $O(n \log n) \sim O(n^2)$ time incremental construction.

- diameter of a set of points

  For n ε-Points in the plane, the minimum of possible diameters can be calculated in $O(n \log n)$ time, and the maximum of them can be calculated in $O(n \log^* n \log n)$ expected time.

- Voronoi diagram

  For a set of n ε-Points in the plane, weak Voronoi regions can be constructed in $O(n^2 \alpha(n)) \sim O(n^2 \log^* n \log n)$ time ($\alpha(n)$ denotes the inverse of Ackermann's function) and strong Voronoi regions can be constructed in $O(n \log^* n \log n) \sim O(n^2 \log^* n \log n)$ time.

We showed that accuracy guaranteed solutions are available for the basic geometric problems in reasonable computational costs. For a future work, the accuracy guaranteed computation with imprecise arithmetic as well as imprecise input is considered.

Acknowledgments

I wish to express my gratitude for guidance and encouragement received from Professor N. Tokura. I would like to thank to Professors K. Hagihara and T. Kashiwabara for their useful comments on the dissertation. I am indebted to S. Yasutome, who is now Associate Professor in Southern Osaka University, Y. Tsujino, who is now Professor in Kyoto Institute of Technology, K. Yoshida, A. Saitoh, H. Masuda and T. Nishida for their helpful discussion on my work. I also thanks to M. Kameda, S. Tanobe, T. Sumiyoshi and all the members of my laboratory for their time and kindness.

Finally, my thank goes to classmates Y. Takenaka and T. Takenaka and all of my families for their support and kindness during my life at the university.

Bibliography

[1] P. K. Agarwal, M. Berg, J. Matoušek, "Constructing Levels in Arrangements and Higher Order Voronoi Diagrams," Proc. 10th Annu. ACM Sympos. Comput. Geom. , pp. 67 - 75, 1994.

[2] M. J. Atallah, "Dynamic Computational Geometry," Proc. 24th Annu. IEEE Sympos. Found. Comput. Sci. , pp. 92 - 99, 1983.

[3] C. Burnikel, K. Mehlhorn and S. Schirra, "The LEDA class real number," Technical Report MPI-I-96-1-001,Max-Planck Institute Inform. 1996.

[4] L. Cai, J. M. Keil, Computing Visibility Information in an Inaccurate Simple Polygon. *International Journal of Computational Geometry & Applications*, vol.7, No. 6, pages 515-537, 1997.

[5] CGAL(Computational Geometry Algorithms Library), http://www.cs.ruu.nl/CGAL/.

[6] B. Chazelle and H. Edelsbrunner, "An optimal algorithm for intersecting line segments in the plane," Journal of ACM., vol. 39, no.1, pp.1-54, 1992.

[7] David P. Dobkin and Diane L. Souvaine, "Computational Geometry in a Curved World," Algorithmica 5, pp. 421-457, 1990.

[8] H. Edelsbrunner, L. Guibas and M. Sharir, "The Upper Envelope of Piecewise Linear functions: Algorithms and Applications," Discrete & Computational Geometry, vol.4, pp. 311-336 , 1989.

[9] Herbert Edelsbrunner, János Pach, Richard Pollack, Raimund Seidel and Micha Sharir, "Arrangements of curves in the plane," Theoretical Computer Science, pp. 319-336, 1992.

[10] S. Fortune, "Stable maintenance of point set triangulations in two dimensions," IEEE 30th Annual Symposium on the Foundations of Computer Science, pp. 494 - 499, 1989.

[11] P. G. Franciosa, C. Gaibisso, G. Gambosi and M. Talamo, "A convex hull algorithm for points with approximately known positions," International Journal of Computational Geometry & Applications , vol. 4, no. 2 pp. 153-163, 1994.

[12] M. F. Goodchild, "Issues of quality and uncertainty," Advances in Cartography, Elsevier Applied Science, London, pp. 113-139, 1991.

[13] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," Info. Pro. Lett., vol.1, pp. 132 - 133, 1972.

[14] L. Guibas, D. Salein and J. Stolfi, Epsilon geometry: building robust algorithm from imprecise computations. *Proceedings of 5th Annual ACM Symposium on Computational Geometry*, pages 208-217, 1989.

[15] L. Guibas, D. Salein, and J. Stolfi, "Constructing strongly convex approximate hulls with inaccurate primitives," Lecture Notes in Computer Science, vol.450, pp. 261-270, Springer-Verlag, Heidelberg, 1990.

[16] C. M. Hoffman, "The problems of accuracy and robustness in geometric computation," IEEE Computer 22 pp.31-42, Mar. 1989.

[17] M. Iri, "Quality of Computing," bit vol.28, no.9, pp.52-55, Sep. 1996, KYORITSU SHUPPAN.

[18] J. W. Jaromczyk and G. W. Wasilkowski, "Computing convex hull in a floating point arithmetic," Computational Geometry, vol.4, pp.283-292, 1994.

[19] M. J. Katz, "3-D Vertical Ray Shooting and 2-D Point Enclosure, Range Searching, and Arc Shooting Amidst Convex Fat Objects". INRIA Research Report no. 2583 (available at ftp://ftp.inria.fr/INRIA/publication/RR), 1995.

[20] D. T. Lee and R. L. Drysdale III, "Generalization of Voronoi Diagrams in the plane," SIAM Journal of Computation vol.10, no.1, pp. 73-87, 1981.

[21] K. Mehlhorn and S. Näher, *The LEDA Platform for Combinatorial and Geometric Computing.*, Cambridge University Press, 1999.

[22] Z. Li and V. Milenkovic, "Constructing Strongly Convex Hulls Using Exact or Rounded Arithmetic," Algorithmica, vol.8, pp.345-364, 1992.

[23] V. Milenkovic, "Double precision geometry: A general technique for calculating line and segment intersections using rounded arithmetic," Proceedings of 30th Annual IEEE Symposium of Foundations of Computer Science, pp.500-505, 1989.

[24] T. Nagai, S. Yasutome and N. Tokura, "Toward exact decision of segments intersection under imprecise input data," Technical Report of IEICE COMP96-23, pp. 11-20, 1996.

[25] T. Nagai, S. Yasutome and N. Tokura, "Convex Hull problem with Imprecise input (in Japanese)," Technical Report of IEICE, COMP96-53, pp. 31-40, 1996.

[26] T. Nagai, S. Yasutome and N. Tokura, "Convex hull problem with imprecise input and its solution (in Japanese)," IEICE Transactions vol. J81-D-I, no. 6, pp.615-625, 1998.

[27] T. Nagai, S. Yasutome and N. Tokura, "Voronoi Region with Imprecise Input (in Japanese)," Technical Report of IEICE, COMP98-31, pp. 85-92, 1998.

[28] T. Nagai, S. Yasutome and N. Tokura, *Voronoi region with imprecise input*, IEICE Transactions vol. J83-D-I, no.1, pp.68-79, 2000 in Japanese.

[29] T. Nagai, S. Yasutome and N. Tokura, "Cluster Voronoi diagram for point clusters that are equivalent under parallel translation," Technical Report of IEICE, COMP99-73, pp.33-40, 2000.

[30] F. P. Preparata, "An optimal real time algorithm for planar convex hulls," Comm. ACM, vol. 22, pp. 402 - 405, 1979.

[31] F. P. Preparata and M. I. Shamos, "Computational Geometry: An Introduction," Springer-Verlag, 1985.

[32] David Rappaport, "A convex hull algorithm for discs, and applications," Computational Geometry: Theory and Applications, vol. 1, No. 3, pages 171-187, 1992.

[33] K. Sugihara, *KEISANKIKA KOUGAKU*, ADVANCED ELECTRONICS SERIES II-2, BAIFUKAN, 1994.

[34] G. T. Toussaint, "Solving geometric Problems with the Rotating Calipers," Proceedings of IEEE MELECON '83, Athens, Greece, 1983.

[35] Chee K. Yap, "An $O(n \log n)$ algorithm for the Voronoi Diagram of a Set of Simple Curve Segments," Discrete & Computational Geometry, vol. 2, pp. 365-393, 1987.

[36] C. K. Yap and T. Dubé, "The exact computation paradigm," Computing in Euclidean Geometry, Lecture Notes Series on Computing, vol.1, pp. 452-492. Workd Scientific, Singapore, 2nd edition, 1995.