# A REVIEW OF PROPERTIES AND VARIATIONS OF VORONOI DIAGRAMS

ADAM DOBRIN

## 1. Introduction

This paper is a review of Voronoi diagrams, Delaunay triangulations, and many properties of specialized Voronoi diagrams. We will also look at various algorithms for computing these diagrams. The majority of the material covered is based on research compiled by Atsuyuki Okabe in *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams* [6]. However, there will also be references to research and results presented in many papers. Multiple algorithms for computing different diagrams were found and translated from, among others, *Centroidal Voronoi Tessellations: Applications and Algorithms* [3] and *A Sweepline Algorithm for Voronoi Diagrams* [4].

Section 2 will introduce Voronoi diagrams and provide examples of where they can be seen and how they are applied. Sections 3 and 4 will discuss basic properties associated with Voronoi diagrams and their duals: Delaunay triangulations. These building blocks will allow the progression into discussing more complex ideas regarding Voronoi diagrams. In Section 5, there will be an exploration of weighted Voronoi diagrams, followed by a study of different methods for constructing Voronoi diagrams in Section 6. The last topic covered, in Section 7, will be the idea of centroidal Voronoi diagrams and different algorithms for their construction.

## 2. What is a Voronoi Diagram?

First, it should be noted that for any positive integer $n$, there are $n$-dimensional Voronoi diagrams, but this paper will only be dealing with two-dimensional Voronoi diagrams. The Voronoi diagram of a set of "sites" or "generators" (points) is a collection of regions that divide up the plane. Each region corresponds to one of the sites or generators, and all of the points in one region are closer to the corresponding site than to any other site. Where there is not one closest point, there is a boundary. Note that in Figure 1, the point $p$ is closer to $p_1$ than to any other enumerated points. Also note that $p'$, which is on the boundary between $p_1$ and $p_3$, is equidistant from both of those points.

As an analogy, imagine a Voronoi diagram in $\mathbb{R}^2$ to contain a series of islands (our generator points). Suppose that each of these islands has a boat, with each boat capable of going the same speed. Let every point in $\mathbb{R}$ that can be reached from the boat from island $x$ before any

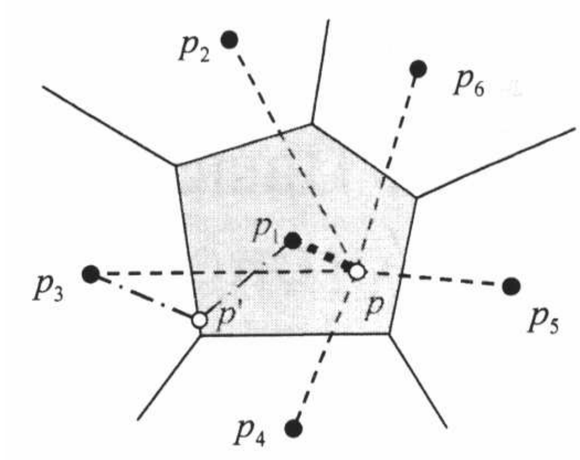



FIGURE 1. A basic Voronoi diagram [6].

other boat be associated with island $x$. The region of points associated with island $x$ is called a Voronoi region.

The basic idea of Voronoi diagrams has many applications in fields both within and outside the math world. Voronoi diagrams can be used as both a method of solving problems or as a model for examples that already exist. They are very useful in computational geometry, particularly for representation or quantization problems, and are used in the field of robotics for creating a protocol for avoiding detected obstacles. For modeling natural occurrences, they are helpful in the studies of plant competition (ecology and forestry), territories of animals (zoology) and neolithic clans and tribes (anthropology and archaelogy), and patterns of urban settlements (geography) [2].

## 3. Basic Properties of the Voronoi Diagram

3.1. **Formal Definition of the Voronoi Diagram.** We have defined a Voronoi diagram informally. Since we are going to be dealing with mathematical problems associated with and algorithms for computing the Voronoi diagram, we must formally define the two-dimensional ordinary Voronoi diagram.

First, we shall denote the location of a point $p_i$ as $(x_{i1}, x_{i2})$, and the corresponding vector will be $\vec{x}$. Let $P = \{p_1, p_2, .., p_n\} \in \mathbb{R}^2$, where $2 \leq n < \infty$ and $p_i \neq p_j$, $i \neq j$ and $\forall\ i,\ j = 1, 2, \ldots, n$ be the set of generator points, or *generators*. We call the region given by

$$V(p_i) = \{\vec{x} \mid ||\vec{x} - \vec{x}_i|| \leq ||\vec{x} - \vec{x}_j||\ \forall j \ni i \neq j\}$$

the *Voronoi region* of $p_i$, where $||\cdot||$ is the usual Euclidean distance. $V(p_i)$ may also be referred to as $V_i$. All Voronoi regions in an ordinary Voronoi are connected and convex. We call the set given by

$$\mathbb{V} = \{V(p_1), V(p_2), \ldots, V(p_n)\}$$

the *Voronoi diagram* of $P$. Different notation for $\mathbb{V}$ is $\bigcup_{i=1}^{n} V_i$.

3.2. **Basic Components of the Voronoi Diagram.** The Voronoi diagram is composed of three elements: generators, edges, and vertices.

$P$ is the set of generators. Every point on the plane that is not a vertex or part of an edge is a point in a distinct Voronoi region.

An *edge* between the Voronoi regions $V_i$ and $V_j$ is $V_i \bigcap V_j = e(p_i, p_j)$. If $e(p_i, p_j) \neq \emptyset$, $V_i$ and $V_j$ are considered *adjacent*. Any point $\vec{x}$ on $e(p_i, p_j)$ has the property that $||\vec{x} - \vec{x}_i|| = ||\vec{x} - \vec{x}_j||$. An edge can be denoted as $e_i$, where $i$ is an index for the edges and does not have to be related to the index of generator points. For example, we can label our edges from 1 to $n$, $n$ being the total number of edges, going top to bottom left to right. The labelling of edges is merely a convenience, and does not have to follow a pre-determined algorithm. It should be decided per Voronoi diagram. Also, the set of edges surrounding a Voronoi region $V_i$ can be referred to as $\partial V(p_i)$, $\partial V_i$, or the boundary of $V_i$.

A *vertex* is located at any point that is equidistant from the three (or more) nearest generator points on the plane. Vertices are denoted $q_i$, and are the endpoints of edges. The number of edges that meet at a vertex is called the *degree* of the vertex. If $\forall q_i \in \mathbb{V}$, degree$(q_i) = 3$, then $\mathbb{V}$ is considered to be *non-degenerate*. Otherwise, $\mathbb{V}$ is considered to be *degenerate*. Throughout this paper, we will, for the most part, assume non-degeneracy in our Voronoi diagrams.

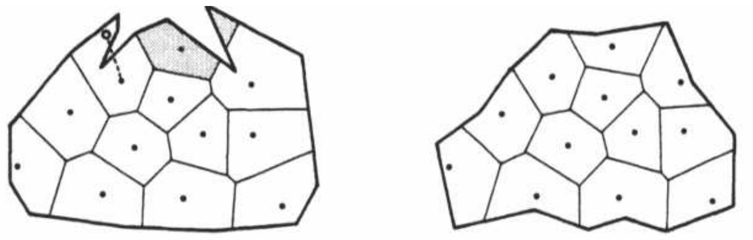


FIGURE 2. Voronoi Diagrams on Bounded Subsets of $\mathbb{R}^2$ [6].

3.3. **Voronoi Diagrams on a Bounded Subset of $\mathbb{R}^2$.** While most of the time we will consider Voronoi diagrams on $\mathbb{R}^2$, we can also have them on any set $S \subseteq \mathbb{R}^2$. We will assume $S$ to be non-empty, for a Voronoi diagram on an empty set would be trivial. The *bounded Voronoi diagram* is defined by $\mathbb{V}_{\cap S} = \{V_1 \cap S, V_2 \cap S, \ldots, V_n \cap S\}$. If, for any $i$, $V_i$ shares the boundary of $S$, we call $V_i \cap S$ a *boundary Voronoi region*. Unlike ordinary Voronoi regions, boundary Voronoi regions need not be connected or convex.

In Figure 2, we see two Voronoi diagrams generated by the same set $P$, but on different subsets of $\mathbb{R}^2$. In the left diagram, the shaded region is not connected, and in both diagrams, many of the regions are not convex. Note that the non-convex regions are boundary regions.

3.4. **Dominance Regions.** Given any two generators, $p_i$ and $p_j$, the perpendicular bisector of the line connecting $p_i$ and $p_j$ is $b(p_i, p_j) = \{\vec{x} \mid ||\vec{x} - \vec{x}_i|| = ||\vec{x} - \vec{x}_j||\}, i \neq j$. $H(p_i, p_j) = \{\vec{x} \mid ||\vec{x} - \vec{x}_i|| \leq ||\vec{x} - \vec{x}_j||\}, i \neq j$ is the *dominance region* of $p_i$ over $p_j$, and consists of

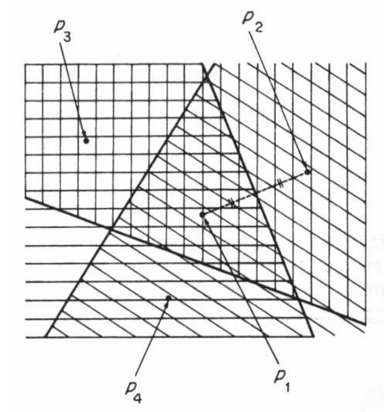



FIGURE 3. Construction of a Voronoi Region Using Half-Planes [6].

every point of the plane that is closer to $p_i$ than $p_j$ or equidistant from the two. $H(p_j, p_i)$, or $\mathrm{Dom}(p_j, p_i)$, is the dominance region of $p_j$ over $p_i$. In the basic Voronoi diagram, $H(p_j, p_i)$ is a half-plane.

From our definition of dominance regions, we can define Voronoi regions in yet another way. Let $P = \{p_1, p_2, \ldots, p_n\} \in \mathbb{R}^2$ be a set of generator points. $V_i = \bigcap_{j \in \mathbb{Z}^+ \leq n} H(p_i, p_j)$ is the ordinary Voronoi region associated with $p_i$. The set $\mathbb{V} = \{V_1, V_2, \ldots, V_n\}$ is the Voronoi diagram on $\mathbb{R}^2$ generated by $P$.

In Figure 3, we can see the construction of a Voronoi region using dominance regions. By drawing in the half-planes associated with $p_1$, we can see how a Voronoi region is created using the method of finding $\bigcap_{j \in \mathbb{Z}^+ \leq n} H(p_i, p_j)$. Using this method for all of the points in a Voronoi

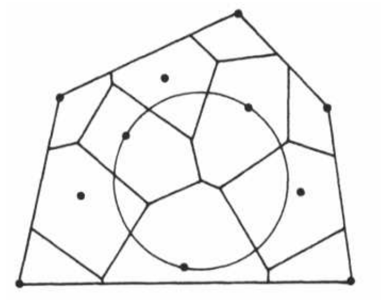


FIGURE 4. A largest empty circle in a basic Voronoi diagram [6].

diagram becomes overly complicated, and is generally bypassed in lieu of other algorithms, which will be discussed in Section 6.

3.5. **The Convex Hull of $P$.** The *convex hull* of $P$ is the smallest convex set containing the generator set $P$, and is denoted as $\mathrm{CH}(P)$. The boundary of this region is referred to as $\partial\,\mathrm{CH}(P)$. Given a Voronoi diagram $\mathbb{V}(P)$, $V_i$ is unbounded iff $p_i \in \partial\,\mathrm{CH}(P)$. With knowledge of the $\mathrm{CH}(P)$, we know the following about a Voronoi diagram $\mathbb{V}(P)$:

(i) A Voronoi edge $e(p_i, p_j)(\neq \emptyset)$ is a line segment iff the line connecting $p_i$ and $p_j$ ($\overline{p_i p_j}$) is not on $\partial\,\mathrm{CH}(P)$.

(ii) A Voronoi edge $e(p_i, p_j)(\neq \emptyset)$ is a half-line or ray iff $P$ is non-collinear and $p_i$ and $p_j$ are consecutive generator points on $\partial\,\mathrm{CH}(P)$.

(iii) All Voronoi edges are lines iff $P$ is collinear.

3.6. **Empty Circles.** With a Voronoi diagram $\mathbb{V}(P)$, it is helpful to know about empty circles. An *empty circle* is one with no generator points within its boundary. For each vertex $q_i \in \mathbb{V}(P)$, there exists a unique empty circle $C_i$ centered at $q_i$ that passes through at least three generator points, and it is the largest empty circle centered at $q_i$. If we assume non-degeneracy, then $C_i$ passes through exactly three generator points. Note that the largest empty circle represented in Figure 4 has three generator points on its boundary.

3.7. **Graph Theory and Voronoi Diagrams.** Also interesting to note are the correlations between basic graph theory and Voronoi diagrams. If we let $n$, $n_e$, and $n_v$ be the number of generator points, Voronoi edges, and Voronoi vertices of a Voronoi diagram, respectively. We find that

$$n_v - n_e + n = 1. \tag{1}$$

If $P$ has more than 3 elements, then

$$n_e \leq 3n - 6 \quad \text{and} \quad n_v \leq 2n - 5$$

Furthermore, if we let $n_c$ be the number of unbounded Voronoi polygons and continue to assume that $P$ has more than 3 elements, then

$$n_v \geq \frac{1}{2}(n - n_c) + 1 \quad \text{and} \quad n_e \leq 3n_v - n_c - 3$$

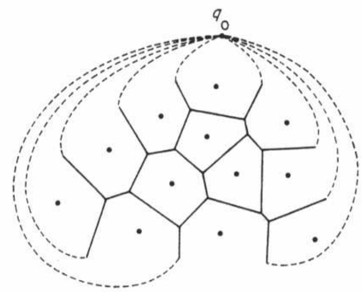



FIGURE 5. A Voronoi graph induced from a Voronoi diagram [6].

Another interesting theorem is that for any Voronoi diagram, the average number of Voronoi edges per Voronoi polygon does not exceed 6. More accurately, this number is less than or equal to $\frac{2(3n-6)}{n}$.

In graph theory, let $G$ be a connected planar graph. Let $v$, $e$, and $f$ be the number of vertices, edges, and faces in $G$, respectively. Euler's Theorem states that $v - e + f = 2$, and that $2e \geq 3f$ and $e \leq 3v - 6$ [7].

We can find a direct correspondence between Euler's Theorem for connected planar graphs and Eq. 1. Take a Voronoi diagram on $\mathbb{R}^2$. Let $n$, $n_b$, $n_c$, $n_e$, and $n_v$ be the number of Voronoi regions, bounded Voronoi regions, unbounded Voronoi regions, Voronoi edges, and Voronoi vertices, respectively. We know that $n_v - n_e + n = 1$. Make any half-line edges into line segments by capping them with a vertex. All new vertices added are added to $n_v$. Simultaneously, all but one of the unbounded Voronoi regions are subtracted from $n_c$; one remains because of the infinite region. (Note: If we had no unbounded Voronoi regions to begin with, we still have an infinite region not previously defined; thus $n_c = 1$.) Since we have included an extra Voronoi region, we've altered our equation to be $n_v - n_e + n = 2$. Disregard our generator set $P$. Our manipulated Voronoi diagram $\mathbb{V}$ has become a connected planar graph $G$. The number of regions $n$ becomes the number of faces $f$ in $G$. The number of vertices $n_v$ becomes the number of vertices $v$ in $G$. The number of edges $n_e$ becomes the number of edges $e$ in $G$. Our equation becomes $v - e + f = 2$, Euler's Theorem.

## 4. Delaunay Tessellations

4.1. **Introduction.** Continuing the theme of graph theory, we will now discuss Delaunay tessellations, which are considered to be dual to Voronoi diagrams [7]. For all Delaunay tessellations, we will assume non-collinearity. This means that for our generator set $P$, the points in $P$ are not all on the same line. Given a Voronoi diagram $\mathbb{V}(P)$, join all pairs of generator points whose Voronoi regions share an edge. Thus, in the Delaunay tessellation of $P$, or $\mathbb{D}(P)$, there exists the edge $\overline{p_i p_j}$ if and only if $e(p_i, p_j) \in \mathbb{V}(P) \neq \emptyset$. If this tessellation consists of only triangles, we call it a *Delaunay triangulation*. If not, we call it a *Delaunay pretriangulation*. A Delaunay tessellation will be a triangulation if

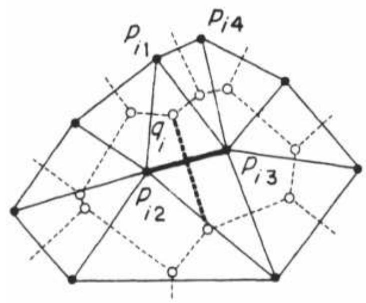



FIGURE 6. A Voronoi diagram and a Delaunay triangulation [6].

and only if all vertices of $\mathbb{V}(P)$ are non-degenerate. See Figure 6 for an illustration of the relation between a Voronoi diagram (dashed lines) and a Delaunay triangulation (solid lines) of the same generator set $P$ (solid dots).

In a Delaunay triangulation, regions are called *Delaunay triangles*. Edges in Delaunay tessellations are called *Delaunay edges*. If a Delaunay edge is shared by two Delaunay triangles, then we call it an *internal Delaunay edge*. Otherwise, we call it an *external Delaunay edge*. The external Delaunay edges in $\mathbb{D}(P)$ constitute $\partial\,\mathrm{CH}(P)$.

Interesting to note is the fact that (assuming non-degeneracy), Voronoi edges and Delaunay edges are orthogonal. With a little thought, this is fairly obvious, because a Voronoi edge $e(p_i, p_j)$ lies on the perpendicular bisector of $\overline{p_ip_j}$, a Delaunay edge. This means that Voronoi diagrams and their Delaunay tessellations are not only dual graphs, but are reciprocal figures [7].

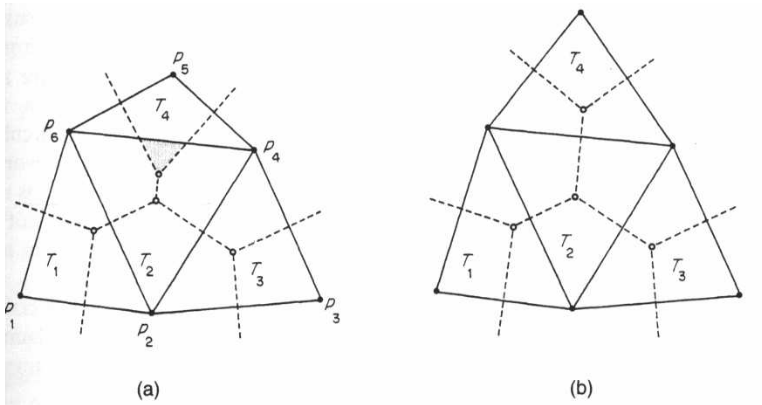



FIGURE 7. Delaunay triangulations: (a) is not a Pitteway triangulation, while (b) is a Pitteway triangulation [6].

4.2. **The Pitteway Triangulation.** The Delaunay triangulation of P is a Pitteway triangulation of $P$ if and only if every internal Delaunay edge $\overline{p_i p_j}$ crosses the associated Voronoi edge $e(p_i, p_j)$ of $\mathbb{V}(P)$.

4.3. **Correspondence Between Voronoi and Delaunay.** We mentioned that Delaunay triangulations are non-degenerate duals to Voronoi diagrams, but have not yet discussed the meaning and results of this claim. Given a non-degenerate Voronoi diagram $\mathbb{V}(P)$ and a Delaunay triangulation $\mathbb{D}(P)$, let $Q$ and $Q_d$ be the sets of Voronoi vertices and Delaunay vertices, respectively. Let $E$ and $E_d$ be the sets of Voronoi edges and Delaunay edges, respectively. Let $C_d$ be the set of circumcenters of Delaunay triangles. Then the following are true:

(i) $Q_d = P$

(ii) $C_d = Q$

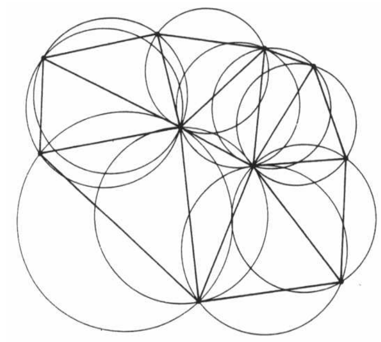


FIGURE 8. Empty circumcircles of Delaunay triangles (i.e. Delaunay circles) in a Delaunay triangulation [6].

(iii)$|E_d| = |E|$

Statement (i) says that each generator point $p_i$ is a vertex of a Delaunay triangle. Statement (ii) says that the circumcenter of each Delaunay triangle corresponds to a Voronoi vertex. Statement (iii) says that the number of Delaunay edges is equal to the number of Voronoi edges.

On a side note, the circumcenter of a Delaunay triangle, is the point equidistant from the three vertices. It is the center of the circumcircle, or *Delaunay circle*, which is the largest empty circle contained in a Delaunay triangle. The circumcircle of the Delaunay triangle is an empty circle only if the triangulation of $P$ is a Delaunay triangulation. The points on a Delaunay circle, which are Delaunay vertices, are called *natural neighbors.* Two vertices are natural neighbors if and only if they are connected by a Delaunay edge.

To close, we will look at another interesting property of Delaunay triangulations. Let $D(p_i, p_j)$ be the shortest path along the Delaunay edges of $\mathbb{D}(P)$ from $p_i$ to $p_j$. Then $D(p_i, p_j) \leq c * d(p_i, p_j)$, where $d(p_i, p_j)$ is the Euclidean distance from $p_i$ to $p_j$ and $c = \frac{2\pi}{3cos(\frac{\pi}{6})} \approx 2.42$.

## 5. The Weighted Voronoi Diagram

5.1. **Introduction.** So far, in our discussion of Voronoi diagrams, we have assumed that our generator points, besides their location, have equal value, or weight. The idea of assigning distinct weight to generator points can be more useful than having uniformly weighted points in some scenarios. Weighted generator points are sometimes more applicable when looking at, for example, the population size of a settlement, the number of stores in a shopping center, or the size of an atom in a crystal structure [6].

Recall from our definition of the Voronoi diagram from Section 3.4. that a Voronoi region $\mathbb{V}_i$ is the intersection of the dominance regions of $p_i$ over every other generator point in $P$. While we will have different formulae for dominance regions in weighted voronoi diagrams, the idea remains the same. The dominance region of a generator point $p_i$ over another, $p_j$, where $i \neq j$ and $d_w(p_x, p_y)$ is the weighted distance between points $x$ and $y$, is written as

$$\mathrm{Dom}_w(p_i, p_j) = \{p | d_w(p, p_i) \leq d_w(p, p_j)\}.$$

Let

$$V_w(p_i) = \bigcap_{p_j \in P \setminus \{p_i\}} \mathrm{Dom}_w(p_i, p_j).$$

$V_w(p_i)$, or $V_w(i)$, is called a weighted Voronoi region, and $\mathbb{V}_w = \{V_w(p_1), V_w(p_2), \ldots, V_w(p_n)\}$ is called the weighted Voronoi diagram. Another way to denote $\mathbb{V}_w$ is $\mathbb{V}(P, d_w)$, where $P$ is the generator set with weights $W = \{W_1, W_2, \ldots, W_n\}$ and $d_w$ is the weighted distance. We will discuss $d_w$ more in depth in the following sections.

5.2. **The Multiplicatively Weighted Voronoi Diagram.** Recall the analogy of generator points to islands with boats from Section 2. The multiplicatively-weighted Voronoi diagram assigns a speed to each boat. Therefore, a faster boat will reach boats previously outside of its region.

This weighted Voronoi diagram has its weighted distance given by

$$d_{mw}(p, p_i) = \frac{||\vec{x} - \vec{x}_i||}{w_i}, \text{ where } w_i > 0.$$

This is called the *multiplicatively weighted distance* or *MW-distance.* There are many names for the associated Voronoi diagram: $\mathbb{V}_{mw}$, the multiplicatively-weighted Voronoi diagram, the MW-Voronoi diagram, the circular Dirichlet tessellation, or the Apollonius model. We will not be using the last two, as they are highly specialized and this paper is a more general survey of ideas and concepts associated with Voronoi diagrams.

A MW-Voronoi region is a non-empty set, it does not have to be convex or connected, and it can have a hole or holes. A MW-Voronoi region $V_w(p_i)$ is convex if and only if the weights of adjacent MW-Voronoi regions are not smaller than the weight of $p_i$. Another way to denote "the weight of $p_i$" is $w_i$. Also, two MW-Voronoi regions may share disconnected edges. Edges in $\mathbb{V}_{mw}$ are circular arcs if and if only if the weights of the two affective regions are not equal. Edges in $\mathbb{V}_{mw}$ are straight lines if and only if the weights of the two affective regions are equal. See Figure 9 for a diagram of the bisectors between points $p_i$ and $p_j$ with multiplicatively weighted distance for several ratios $\alpha = \frac{w_i}{w_j} = 1,\ 2,\ 3,\ 4,\ 5.$

Let $w_{max} = \max\{w_j, p_j \in P\}$ and $P_{max} = \{p_j | w_j = w_{max}\}$. A MW-Voronoi region $V_{mw}(p_i)$ is unbounded if and only if $p_i \in P_{max}$ and $p_i$ is on $\partial\,\mathrm{CH}(P_{max})$.

5.3. **The Additively Weighted Voronoi Diagram.** Continuing the boat/island analogy, imagine that boats in the additively weighted

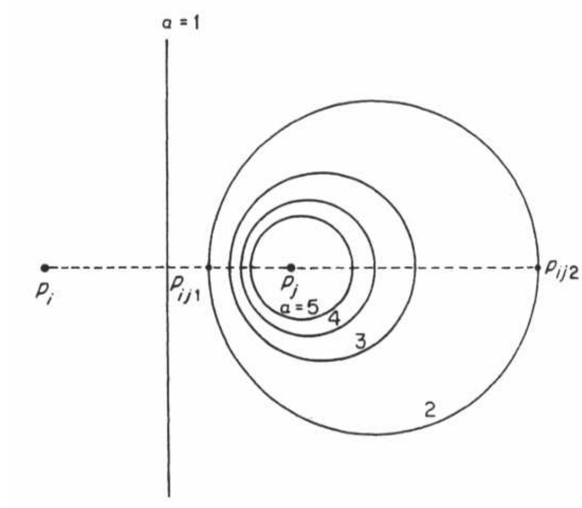



FIGURE 9. Multiplicatively weighted Voronoi diagrams for two generator points [6].

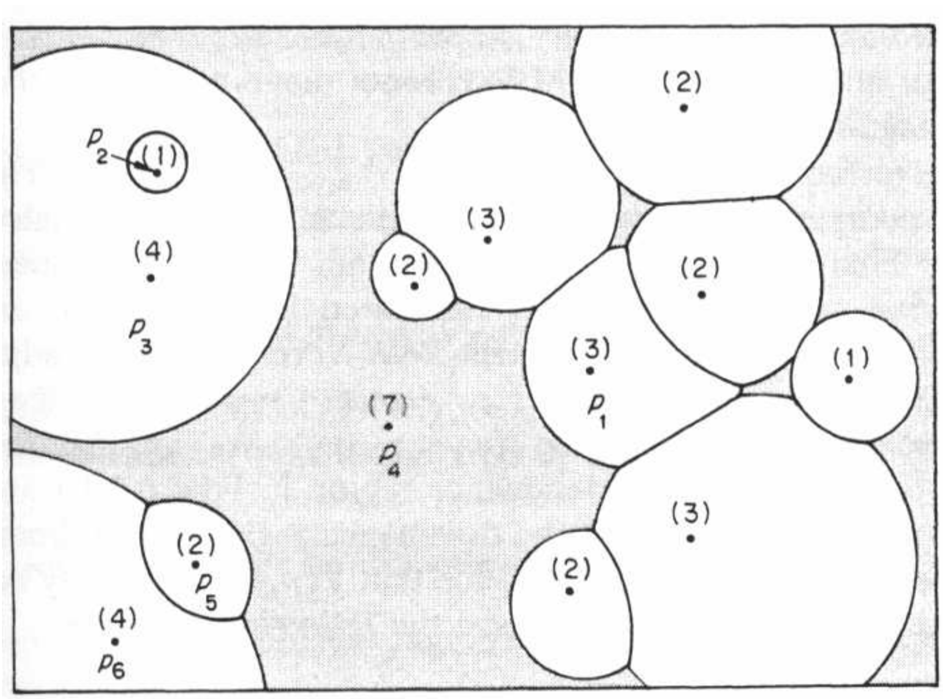



FIGURE 10. A multiplicately weighted Voronoi diagram (the numbers in parentheses represent weights associated with the generators) [6].

Voronoi diagram start a certain distance away from their respective islands. They all still travel at the same speed, but some boats begin closer to points than they would in the ordinary Voronoi diagram.

This type of weighted Voronoi diagram has its weighted distance given by

$$d_{aw}(p, p_i) = ||\vec{x} - \vec{x}_i|| - w_i.$$

The dominance region of the additively-weighted Voronoi diagram is given by

$$\mathrm{Dom}_{aw}(p_i, p_j) = \{\vec{x} \mid ||\vec{x} - \vec{x}_i|| - ||\vec{x} - \vec{x}_j|| \leq w_i - w_j\}, \text{ where } i \neq j.$$

If we let $\alpha = ||\vec{x}_i - \vec{x}_j||$, and $\beta = w_i - w_j$, we get the following results. If $\alpha = \beta$, then the dominance region of $p_j$ over $p_i$ is a half-line radiating from $p_j$ directly away from $p_i$. This result is impossible with the multiplicatively-weighted Voronoi diagram. If $0 < \alpha < \beta$, then $p_i$ completely dominates $p_j$, and $V_{aw}(p_j) = \emptyset$, another result not possible with MW-Voronoi diagrams.

With these properties in mind, we find the following statements to be true. The set $V_{aw}(p_i) = \emptyset$ if and only if

$$\min\{||x - x_i|| - w_j, \ \forall p_j \in P \ni i \neq j\} < -w_i.$$

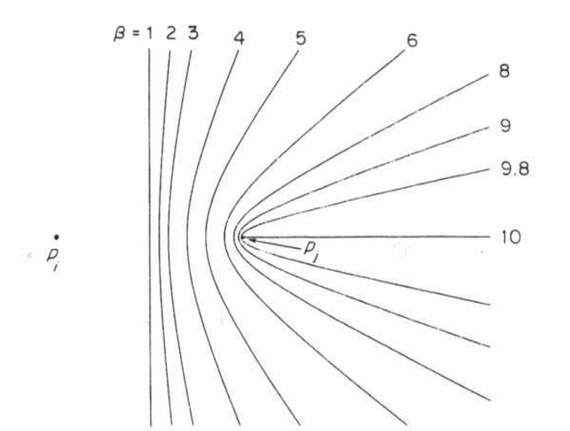



FIGURE 11. Additively weighted Voronoi diagrams for two generator points [6].

The set $V_{aw}(p_i)$ is a half-line if and only if

$$\min\{||x - x_i|| - w_j,\ \forall p_j \in P \ni i \neq j\} = -w_i.$$

The set $V_{aw}(p_i)$ has positive area if and only if

$$\min\{||x - x_i|| - w_j,\ \forall p_j \in P \ni i \neq j\} > -w_i.$$

Every edge in $\mathbb{V}_{aw}$ is either a hyperbolic arc, line segment, half-line, or infinite line. See Figure 11 for a diagram of the bisectors between points $p_i$ and $p_j$ with additively weighted distance for several parameter values $\alpha = ||\vec{x}_i - \vec{x}_j||$ and $\beta = |w_i - w_j| = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 8,\ 9,\ 9.8,\ 10.$

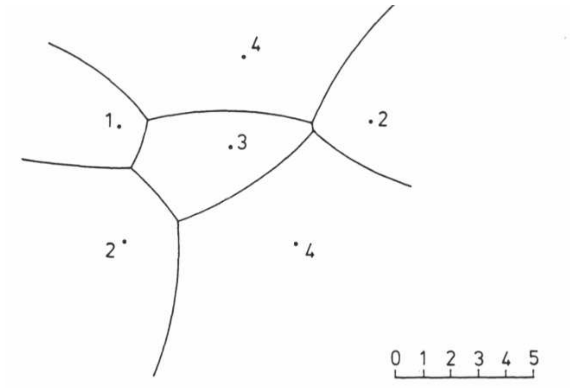



Figure 12. An additively weighted Voronoi diagram (the numbers indicate weights) [6].

If at least one weight, $w_i$, is different from another, and $V_{aw}(p_i)$ has positive area, then there exists at least one non-convex additively-weighted Voronoi region. Every non-convex AW-Voronoi region is star-shaped with respect to its generator point. This means that from $p_i$, we can draw a line to any point in the region $V_{aw}(p_i)$, and the line will be contained entirely within $V_{aw}(p_i)$.

## 6. Algorithms for Constructing the Voronoi Diagram

6.1. **Introduction.** There are many different ways to construct the ordinary Voronoi diagram (defined by $d_w = ||\vec{x} - \vec{x}_i||$, $i \neq j$). In this section, we will look at two of these: the Plane-Sweep and the Tree Expansion and Deletion methods.

6.2. **Plane-Sweep Method.** For the plane-sweep method of constructing the Voronoi diagram, we draw all generator points, Voronoi vertices, and Voronoi edges in a non-Cartesian plane. There is a one-to-one correspondence between the non-Cartesian plane and the Cartesian plane, given some initial assumptions. Therefore, we can copy our Voronoi diagram in the non-Cartesian plane to the Voronoi diagram in the Cartesian plane, which is what we ultimately want.

Let $P = \{p_1, p_2, \ldots, p_n\}$ be our generator set. For any point $p$, we define $r(p) = \min\{d(p, p_i) \mid 1 \leq i \leq n\}$. By the properties of Voronoi diagrams, $r(p)$ is the radius of the largest empty circle centered at $p$. We shall denote the location of $p$ as $(x(p), y(p))$. Let

$$\phi(p) = (x(p) + r(p), y(p)).$$

In other words, for any point $p \in V_i$, $\phi(p) = (x(p) + d(p, p_i), y(p))$. $\phi(p) = p$ if and only if $p$ is a generator point. Any $q \in \mathbb{V}$, $e \in \mathbb{V}$, $V_i \in \mathbb{V}$, and $\mathbb{V}$ have images $\phi(q)$, $\phi(e)$, $\phi(V_i)$, and $\phi(\mathbb{V})$, respectively.

Let $p_i$ and $p_j$ be two generator points such that $x(p_i) > x(p_j)$ where $e(p_i, p_j) \neq \emptyset$. $\phi$ maps $e(p_i, p_j)$ to part of a hyperbola with leftmost point $p_i$. We will cut this hyperbola at $p_i$ into $h^+(p_i, p_j)$ and $h^-(p_i, p_j)$. $h^+(p_i, p_j) = h^+(p_j, p_i)$ and $h^-(p_i, p_j) = h^-(p_j, p_i)$.

To help the reader better understand $\phi(\mathbb{R}^2)$, we can draw an analogy between it and $\mathbb{R}^2$. Recall the boat/island analogy from before. Imagine that there is a current, moving directly to the right, that is faster than the uniform-velocity abled boats. Therefore, no boat can reach a point to the left of its island. Voronoi regions in $\phi(\mathbb{R}^2)$ are contained within open-right hyperbolas.

For the plane-sweep method, we will need to make four initial assumptions. First, numerical computation will be carried out in precise arithmetic. Second, no four generator points align on a common circle. This is the same as assuming non-degeneracy. Third, no two generator points align vertically. Lastly, any generator point and any of that generator point's Voronoi vertices do not align horizontally.

In this method, we take a vertical line and move it from left to right over the plane. We update our data everytime there is an event, which will be defined as:

:the sweepline hits a generator point.

:the sweepline hits a Voronoi vertex in $\phi(\mathbb{V})$.

To represent the structure of $\phi(\mathbb{V})$ along the sweepline, we will use an alternating list $L$ of regions and boundary edges which appear on the line, from bottom to top. We will also use $Q$, a set of points in the plane. It is a list of all possible points where events may occur. To begin with, $Q = P$, the set of all generator points. Candidates for Voronoi vertices $\phi(q)$ are added to $Q$ as they are found by the sweepline.

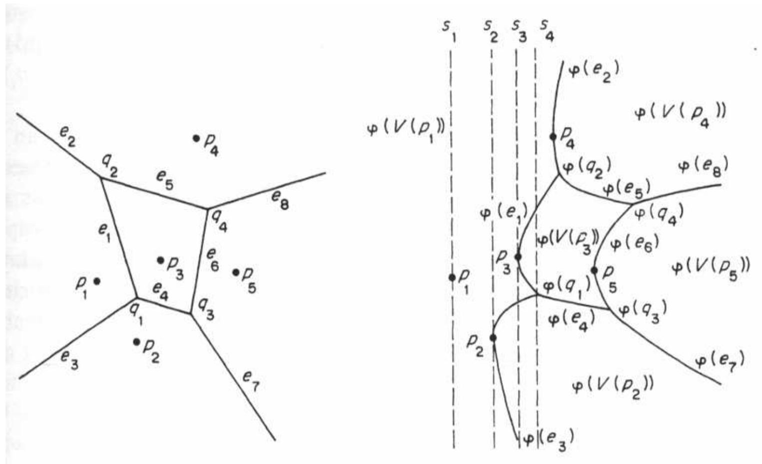


FIGURE 13. Voronoi diagram and its image: (a) Voronoi diagram $\mathbb{V}$ for five generators and (b) its image $\phi(\mathbb{V})$ [6].

Given a set $P = \{p_1, p_2, \ldots, p_n\}$ of $n$ generator points, using the following algorithm, we will end up with the transformed Voronoi diagram $\phi(\mathbb{V})$.

(1) Let $Q = P$.
(2) Choose and delete the leftmost point, say $p_i$, from $Q$.
(3) Let $L$ be a list consisting of a single region, $\phi(V(p_i))$.
(4) While $Q$ is non-empty, repeat 4.a, 4.b, and 4.c.
    (a) Choose and delete the leftmost point $w$ from $Q$.
    (b) If $w$ is a generator point, say $w = p_i$, do 4.b.i, 4.b.ii, and 4.b.iii.
        (i) Find the region $\phi(V(p_j))$ on $L$ containing $p_i$.

(ii) Replace $(\phi(V(p_j)))$ on $L$ with

$$(\phi(V(p_j)), h^-(p_i,p_j), \phi(V(p_i)), h^+(p_i,p_j), \phi(V(p_j))).$$

(iii) Add to $Q$ the intersection of $h^-(p_i,p_j)$ with the immediate lower-half hyperbola on $L$ and the intersection of $h^+(p_i,p_j)$ with the immediate upper-half hyperbola on $L$.

(c) If $w$ is an intersection, say $w = \phi(q_t)$, do 4.c.i, 4.c.ii, 4.c.iii, and 4.c.iv.

(i) Replace the subsequence $(h^\pm(p_i,p_j), \phi(V(p_j)), h^\pm(p_j,p_k))$ on L with $h = h^-(p_i,p_k)$ or $h = h^+(p_i,p_k)$.

(ii) Delete from $Q$ any intersections of $h^\pm(p_i,p_j)$ or $h^\pm(p_j,p_k)$ with other half-hyperbolas.

(iii) Add to $Q$ any intersections of $h$ with its immediate upper- and lower-half hyperbola on $L$.

(iv) Mark $\phi(q_t)$ as a Voronoi vertex incident to $h^\pm(p_i,p_j)$, $h^\pm(p_j,p_k)$, and $h$.

(5) Report all half-hyperbolas that were ever listed on $L$, all the Voronoi vertices marked in 4.c.iv, and the incidence relations among them. [6]

From this algorithm, we have all edges, vertices, and generator points of $\phi(\mathbb{V})$ on $\phi(\mathbb{R}^2)$. We now need to be able to convert a point $\phi(p)$ to $p \in \mathbb{R}^2$. We know to which generator points each edge and vertex is associated. Call one of these generator points $p_i$. We know that $\phi(p) = (x(p) + r(p), y(p))$. Let $x_i = x(p_i)$, $y_i = y(p_i)$, $x_\phi = x(\phi(p))$, and $y = \phi = y(\phi(p))$. Let $m$ be the slope of the line between $p_i$ and $\phi(p)$ and $d$ be the Euclidean distance between $p_i$ and $\phi(p)$. $m = \frac{y_\phi - y_i}{x_\phi - x_i}$ and $d = \sqrt{(y_\phi - y_i)^2 + (x_\phi - x_i)^2}$. From this, we find that

$$r(p) = \frac{d}{2\cos(\tan^{-1}(m))}. \tag{2}$$

Our point,

$$p = (x_\phi - r(p), y_\phi), \tag{3}$$

can now be written with Eq. 2 and Eq. 3:

$$p = (x_\phi - \frac{d}{2\cos(\tan^{-1}(m))}, y_\phi). \tag{4}$$

Using the sweepline algorithm and Eq. 4, we can graph a Voronoi diagram in $\phi(\mathbb{R}^2)$, then translate each point on an edge or vertex back to $\mathbb{R}^2$, giving us our Voronoi diagram $\mathbb{V}$.

6.3. **Tree Expansion and Deletion Algorithm.** While the Plane-Sweep Method is a good system to construct the Voronoi diagram, it is more useful as a programmable way to do so. Carrying out the method by hand, or even with Maple or Mathematica, is a laborious task, involving much more work than is actually necessary. For simpler Voronoi diagrams (those with a relatively small generator set $P$), we can utilize a simpler algorithm. This algorithm takes a Voronoi diagram on $l-1$ generator points and another generator point in the plane, and gives a Voronoi diagram on $l$ generator points.

To add $p_l$, the $l^{th}$ generator point, we need to know the vertices of $\mathbb{V}_{l-1}$ that will be affected by $V(p_l)$. We can do so with the following information. Let $q_{ijk}$ denote the vertex incident to $V(p_i)$, $V(p_j)$, and $V(p_k)$ in that order. For some point $p=(x,y)$, we let $H(p_i,p_j,p_k,p) = 0$ be the circle that passes through $p_i$, $p_j$, $p_k$, and $p$. The circle contains $p$ if and only if $H(p_i,p_j,p_k,p)<0$. Consequently, $q_{ijk}$ is in $V(p_l)$ if and only if $H(p_i,p_j,p_k,p_l)<0$. We will let

$$H(p_i,p_j,p_k,p_l) = \begin{vmatrix} 1 & x_i & y_i & x_i^2+y_i^2 \\ 1 & x_j & y_j & x_j^2+y_j^2 \\ 1 & x_k & y_k & x_k^2+y_k^2 \\ 1 & x & y & x^2+y^2 \end{vmatrix}. \tag{5}$$

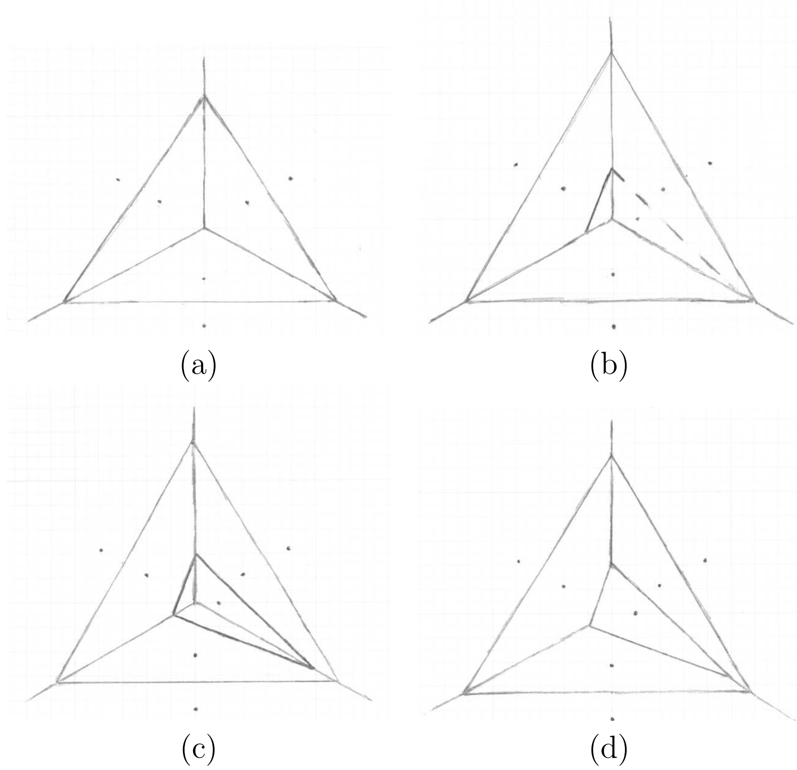


FIGURE 14. Implementation of the Tree Expansion and Deletion Algorithm.

Like the Plane-Sweep Method, we will need to make some basic assumptions to begin. We will need non-degeneracy, so we will assume that the degree of any vertex in $\mathbb{V}_{l-1}$ is 3. A problem arises, though, when $H(p_i, p_j, p_k, p_l) = 0$. When we add $p_l$ to $\mathbb{V}_{l-1}$, we will have a vertex of degree 4. To avoid this, we can change our inequality to say that $H(p_i, p_j, p_k, p_l) \geq 0$ if and only if $q_{ijk}$ is outside of $V(p_l)$. We will also assume that $\mathbb{V}_{l-1}$ divides the plane into $i + 1$ regions; call them cells. Assume that every cell, besides infinite ones, is simply connected with no holes. Also, two cells share at most one edge.

We will let $T$ be the set of vertices of $\mathbb{V}_{l-1}$ that will be in $V(p_l)$. Assume that $T$ is non-empty. Also, assume that $\mathbb{V}_{l-1}(T)$ (the components of $\mathbb{V}_{l-1}$ in $T$) is a tree. Following the aforementioned assumptions and given $P = (p_1, p_2, \ldots, p_{l-1}, p_l)$ and $\mathbb{V}_{l-1}$, we will implement the following procedure to compute a new Voronoi diagram $\mathbb{V}_l$:

(1) Find the generator $p_i (1 \leq i \leq l-1)$ such that $d(p_i, p_l)$ is minimized.
(2) Among the vertices $q_{ijk}$ on $\partial V(p_i)$, find the one that gives the smallest value of $H(p_i, p_j, p_k, p_l)$. Let $T$ be the set consisting of only this vertex.
(3) Repeat 3.a until $T$ cannot be further augmented.
    (a) For each vertex $q_{ijk}$ connected by an edge to an existing element of $T$, add $q_{ijk}$ to $T$ if $H(p_i, p_j, p_k, p_l) < 0$ and if the resultant $T$ is a tree (satisfying one of our initial assumptions).
(4) For every generator $p_i$ associated with a vertex $q_{ijk}$ that satisfies assumption 4.6.7., draw the perpendicular bisector between $p_i$ and $p_l$ from $e(p_i, p_j)$ to $e(p_i, p_k)$. Let this line segment be $e(p_i, p_l)$, the interesection of $e(p_i, p_l)$ and $e(p_i, p_j)$ be the vertex $q_{ijl}$, and the intersection of $e(p_i, p_l)$ and $e(p_i, p_k)$ be the vertex $q_{ikl}$. In the case that $q_{ijk}$ lies on the outer circuit of $\mathbb{V}_{l-1}$, draw

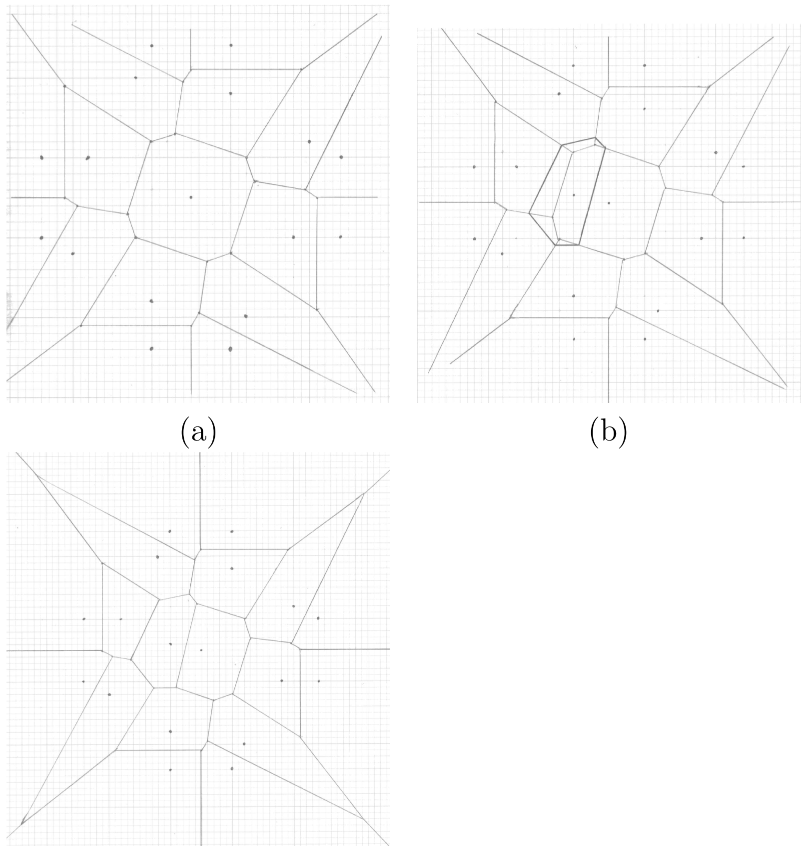
the perpendicular bisector between $p_i$ and $p_l$. When applicable, the bisector is a segment from $e(p_i, p_j)$ to $e(p_i, p_k)$. In this case, let this line segment be $e(p_i, p_l)$, the intersection of $e(p_i, p_l)$ and $e(p_i, p_j)$ be the vertex $q_{ijl}$, and the intersection of $e(p_i, p_l)$ and $e(p_i, p_k)$ be the vertex $q_{ikl}$. If the bisector does not intersect either $e(p_i, p_j)$ or $e(p_i, p_k)$, it will be a ray originating on $e(p_i, p_m)$, where $p_m$ is either $p_j$ or $p_k$, whichever is associated edge intersects with the perpendicular bisector. Let this ray be $e(p_i, p_l)$, and the intersection of $e(p_i, p_l)$ and $e(p_i, p_m)$ be the vertex $q_{iml}$.

(5) Remove all vertices in $T$, and the edges incident to them, from $\mathbb{V}_{l-1}$.

(6) Consider the interior of the edges and vertices added in 4 to be $V(p_l)$, and the resulting diagram to be $\mathbb{V}_l$.

In Figure 14, we see the use of the Tree Expansion and Deletion Algorithm on a Voronoi diagram with six generator points. The new generator point affects only the central vertex. For the other three vertices, $H \geq 0$. In Figure 15, we see the implementation of the algorithm on a Voronoi diagram that, while symmetrical, much more complicated than in Figure 14. We find that $H < 0$ for four vertices.

To keep our vertices and regions organized, we have terms by which we can refer to them. We call vertices in $T$ *in*. Vertices outside of $T$



(a) (b) (c)

FIGURE 15. A Second Implementation of the Tree Expansion and Deletion Algorithm.

are *out*. During the algorithm, vertices that are as of yet unchecked are *undecided*. We call polygons with a vertex in $T$ *incident* and all other polygons *non-incident*. All vertices and polygons are initially labelled undecided and non-incident, respectively.

## 7. The Centroidal Voronoi Diagram

7.1. **Introduction.** A centroidal Voronoi diagram, or tessellation, is a Voronoi diagram of a given set such that every generator point is also the centroid, or center of mass, of its Voronoi region. Typically, centroidal Voronoi tessellations (CVT's) have an associated density function, $\rho(x, y)$, on $\Omega$, a subset of $\mathbb{R}^2$.

7.2. **Real-World Applications and Observations.** Applications for CVT's cover a multitude of disciplines from computer science to urban planning.

An interesting area in which CVT's pop up is in the territorial behavior of animals. For example, the male mouthbreeder fish, to establish domain, will spit sand away from the center of its territory. For a high enough density of fish introduced simultaneously into a body of water with a uniform sandy floor, we find an interesting result. The spitting of sand results in raised bars of sand, and, when viewed from above, creates a pattern which very closely approximates a centroidal Voronoi tessellation [3].

Also in biology, CVT's appear in the cell division of animal and plant cells. In a study of the development of starfish embryos, it was found that a view of a layer of columnar cells in an arrangement in a hollow sphere showed the cells to closely match a centroidal Voronoi tessellation [3].

Looking at urban planning ideas, we can see close correlations to those of the CVT. If we just look at the convenience of mailbox placement throughout a population, we make the following assumptions:

(1) A person will use the mailbox nearest to their home.
(2) The cost to a person of using a mailbox is a function of the distance from the person to the mailbox.
(3) The cost to the general population is measured by the distance to the nearest mailbox averaged over the population.
(4) The optimal placement of mailboxes is one that minimizes cost, or the distance to the population in general.

It makes sense, then, that the optimal placement of mailboxes is at the centroids of a CVT, using the population density as a basis for generator points. We can use these ideas in many different aspects of urban planning and the placement of resources. Schools, distribution centers, mobile vendors, bus terminals, voting stations, and service stops are some examples [3].

7.3. **Generator Sets.** The idea of the centroidal Voronoi tessellation is somewhat limiting in terms of the various diagrams that we can produce. The original Voronoi diagrams that we considered have a one-to-one correspondence between every possible set of generator points $P$ and the Voronoi diagram $\mathbb{V}(P)$. The same is not true for centroidal Voronoi tessellations: since not every Voronoi diagram is a CVT, we cannot construct a CVT from just any given generator set.

In the discussion of the optimal placement of resources, we mentioned population density, but did not explain why this was important. To create a CVT, we need a generator set whose corresponding Voronoi diagram has the property that each generator point is the centroid of its Voronoi region. This implies that the problems that we will run into when trying to find a CVT will lie solely in the generator set.

This is where the disparity between CVT's and ordinary Voronoi diagrams arises. The freedom that we have with regular Voronoi diagrams is important because we can visualize dominance regions and distances given any set of data. CVT's are limited in this sense, but, as in the applications and observations mentioned in Section 7.2, they are very useful for planning systems and noticing similarities in different fields (like animals' territorial habits and cell division).

7.4. **Constructing a Generator Set.** To create a CVT, we need to find a generator set whose points are the mass centroids of their respective Voronoi regions. There are many methods for doing so: the two that we will look at are Lloyd's method and MacQueen's method. The general idea of these methods is to take an initial set of $n$ points, $P$, then move them incrementally such that the resulting Voronoi diagram is also a CVT.

To select an initial set of points, we use a Monte Carlo method. A Monte Carlo method is any which solves a problem by generating suitable random numbers and observing only the fraction of the numbers obeying some property or properties [5]. A Monte Carlo method is not necessary to create a CVT: we could select $n$ random points on $\Omega$, then use Lloyd or MacQueen's method, and get a very similar, and for all intents and purposes, the same result. The Monte Carlo method, in effect, gives us a head start on creating the generator set for the CVT. It gives us a set of $n$ points that resembles our density function, $\rho(x, y)$. Because Lloyd and Macqueen's methods take the $n$ points and iterate them to find a generator set for a CVT, which also resembles $\rho$, the set found using a Monte Carlo method may be closer to our final set. By using a Monte Carlo method, we save time overall by reducing the number of iterations of Lloyd or MacQueen's method.

7.5. **Lloyd's Method.**

(1) Select an initial set of $n$ points ($P$) in $\Omega$, by using a Monte Carlo method.

(2) Construct a Voronoi tessellation of $\Omega$, $\mathbb{V}$, associated with $P$.

(3) Compute the mass centroids of the Voronoi regions in $\mathbb{V}$

(4) Let $P$ be the set of mass centroids computed in Step 3.

(5) If $P$ meets some predetermined criteria, implement Step 2, then terminate; otherwise, return to Step 2.

7.6. **MacQueen's Method.**

(1) Select an initial set of $n$ points, $(P)$ in $\Omega$, by using a Monte Carlo method; let $i = 1$.

(2) Determine a point $y$ in $\Omega$ by using the same Monte Carlo method as in Step 1.

(3) Find the point $p^*$ in $P$ that is closest to $y$.

(4) Set

$$p^* = \frac{ip^* + y}{i + 1}$$

and $i = i + 1$.

(5) If $P$ meets some predetermined criteria, go to Step 6; otherwise, return to Step 2.

(6) Construct a Voronoi tessellation of $\Omega$, $\mathbb{V}$, associated with $P$.

7.7. **Similarities and Differences.** Lloyd and MacQueen's methods are very similar, but also very different. They differ just in their method of resetting $P$. For Lloyd's method, every time we go through the cycle of steps (an iteration), we are constructing a new Voronoi diagram. Considering the fact that we probably must go through several hundred iterations, this is a very time-consuming method for computers. For MacQueen's method, instead of recomputing the Voronoi diagrams, we are simply moving a point, then checking our results with the "predetermined criteria." With MacQueen's method, though, we only move one point at a time, instead of many points in the set. Lloyd's method gives a better approximation of a CVT, but the difference is not notable enough to justify its use over MacQueen's method. Because of this, we will use MacQueen's method for constructing the CVT.

7.8. **Stopping Criteria.** Typically, we will have two different types of criteria, but only use one for any given construction of a CVT. One criterion will be the total number of iterations, $i$. The higher the number of iterations is, the closer $\mathbb{V}$ will be to an actual centroidal Voronoi tessellation. The other type of criterion will be an error counter, $\epsilon$. If our set $P$ remains relatively unchanged for a certain number of iterations, then we consider MacQueen's method completed. Because $y$ is chosen using a Monte Carlo method, it is possible that $p^* = y$. Then, $P$ would be unchanged. Since this could happen on any given iteration, we would want to set our error counter to cover multiple iterations. It

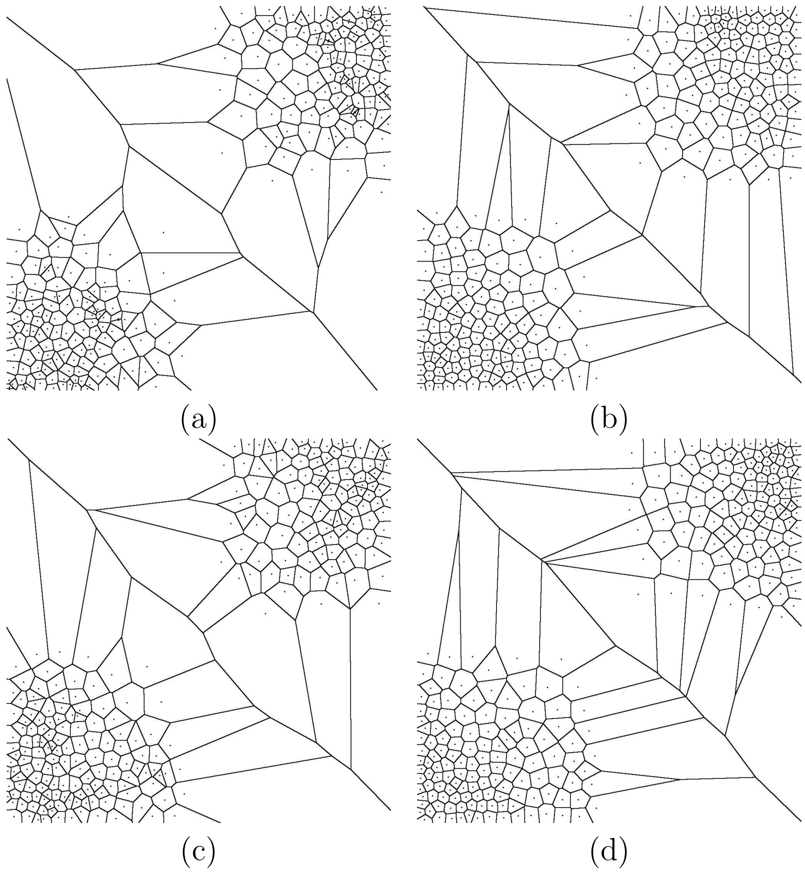



FIGURE 16. 256-Point CVT's Using a Probability Density Function of $\rho(x, y) = xy$ on $[-1, 1] \times [-1, 1]$ with Limits of: (a) $i = 10^3$, (b) $i = 10^5$, (c) $\epsilon = 10^{-4}$, and (d) $\epsilon = 10^{-8}$.

is very rare that it would happen multiple times in a row. The lower the error counter is, the closer $\mathbb{V}$ will be to an actual centroidal Voronoi tessellation.

7.9. **Imperfections in Constructing a CVT.** It is important to take note of the fact that we do not get CVT's with either Lloyd or MacQueen's methods. We consider the approximations that we receive

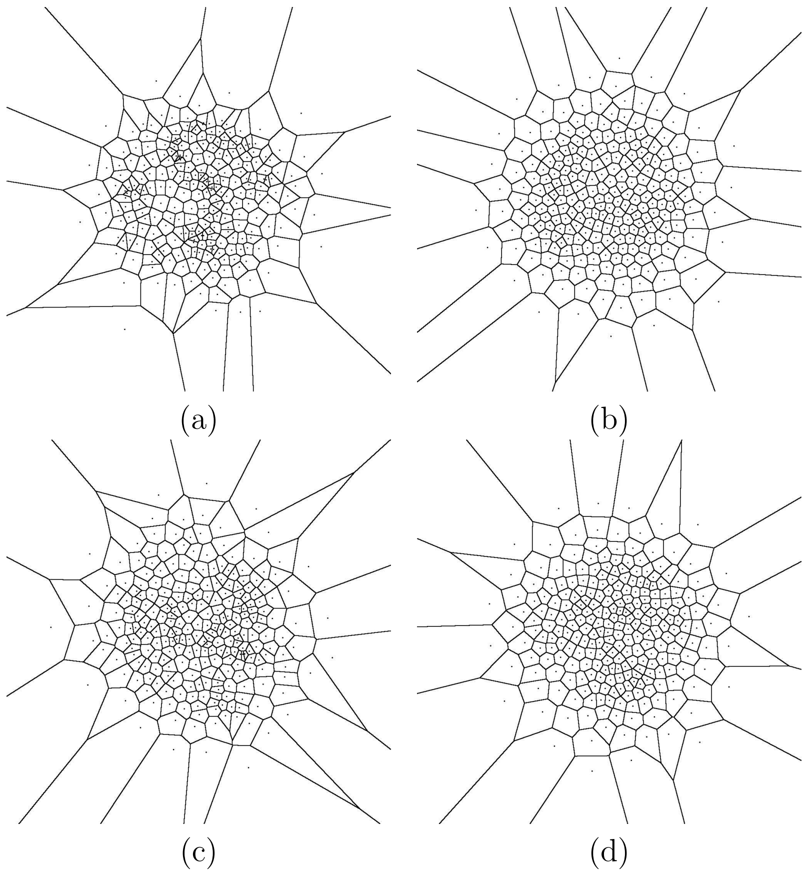


FIGURE 17. 256-Point CVT's Using a Probability Density Function of $\rho(x, y) = e^{-10x^2y^2}$ on $[-1, 1] \times [-1, 1]$ with Limits of: (a) $i = 10^3$, (b) $i = 10^5$, (c) $\epsilon = 10^{-4}$, and (d) $\epsilon = 10^{-8}$.

using these methods close enough to true CVT's because the differences between them are minimal. The only way to guarantee that the Voronoi diagram of $P$ that we get using Lloyd or MacQueen's method is truly a centroidal Voronoi diagram is by carrying the method out indefinitely. This cannot be done for all practical purposes. Thus, we accept the approximation that we make.

7.10. **Figures 16 and 17.** In both Figures 16 and 17, (a) and (b) are computed using an iteration limit, while (c) and (d) are computed using an error counter. In Figure 16, note the difference between the two diagrams on the left and the two on the right. In (a) and (c), we see Voronoi regions bunched together that are relatively far away from the points $(-1,-1)$ and $(1,1)$. Some of these problems are corrected in (b) and (d), as we can observe a better order of smaller regions closer to those points than in (a) and (c). In Figure 17, we can see the same general trends as in Figure 16. The smaller regions are closer to the origin in (b) and (d) than in (a) and (c). These results are expected because (b) uses a higher iteration limit than does (a), and (d) uses a lower error counter than does (c). These results follow directly from the ideas put forth in Section 7.8.

## 8. Conclusion

This paper is a survey of subjects closely related to Voronoi diagrams. There are many topics on Voronoi diagrams that were not covered in this paper. Future studies may include an analysis of problems about properties of the types of Voronoi diagrams that are discussed in this paper. A seemingly simple, and particularly interesting problem is, given the Voronoi edges of a non-degenerate, ordinary Voronoi diagram, to find the locations of the generator set $P$ [6]. Also fascinating are "post-office" problems, ones that, given a set of locations (generators) and mail centers, finding an optimal and efficient algorithm for which mail centers deliver to which location, and in what order. Problems involving biological modeling using Voronoi diagrams are very interesting as well. Charting patterns of crystal growth using additively- and multiplicatively-weighted Voronoi diagrams is an interesting application of Voronoi diagrams to biological studies [2].

There are also many types of Voronoi diagrams with interesting properties that are not included in this paper. For example, there are many other metrics to use when computing weighted Voronoi diagrams. Compoundly-weighted Voronoi diagrams use a combination of multiplicatively- and additively-weighted diagrams. Edges of CW-weighted diagrams are fairly complex, and are either part of a fourth-order polynomial curve, a hyperbolic arc, a circular arc, or a line segment, half-line, or line. The (additively-weighted) power distance to a point is characterized by subtracting the weight of a generator from the square of the Euclidean distance between the generator and the point. Power diagrams contain only line segments, half-lines, and/or lines. These two metrics of Voronoi diagrams are very useful in modeling real-world occurrences, as are the two covered in the paper, because the introduction of weighted distances account for properties inherent in generators. These properties are akin to those of the boats described in the boats/island analogies.

We can also apply weighted distances in the field of centroidal Voronoi diagrams. Another possible future project is to use weighted CVT's to reassociate congressional districts. Instead of creating districts based on many different subjects (including politics or geography), using population density to redistrict a state could result in a more balanced system.

## References

[1] Aurenhammer, F. and Edelsbrunner, H. “An Optimal Algorithm for Constructing the Weighted Voronoi Diagram in the Plane.” *Pattern Recognition* 17, 2 (1984): 251-257.

[2] *Geometry in Action*. Drysdale, Scot. 14 Nov 1996. Donald Bren School of Information and Computer Sciences at University of California, Irvine. `http://www.ics.uci.edu/~eppstein/gina/scot.drysdale.html`.

[3] Du, Qiang, Vance Faber, and Max Gunzburger. “Centroidal Voronoi Tessellations: Applications and Algorithms.” *SIAM Review* 21 (1999): 637-676.

[4] Fortune, Steven. “A Sweepline Algorithm for Voronoi Diagrams.” *Proceedings of the Second Annual ACM Symposium on Computational Geometry* Yorktown Heights, New York: Association for Computing Machinery, 1986: 313-322.

[5] Ju, Lili, Qiang Du, and Max Gunzburger. “Probabilistic Methods for Centroidal Voronoi Tessellations and Their Parallel Implementations.” *Parallel Computer* 28 (2002):1477-1500.

[6] Okabe, Atsuyuki, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, 2nd Ed.*. Chichester, West Sussex, England: John Wiley & Sons, 1999.

[7] Wilson, Robin J. *Introduction to Graph Theory*. Boston: Addison Wesley, 1996.