Finding Constrained And Weighted Voronoi Diagrams In The Plane

Cao An Wang [1] Peter Y. Tsin [2]

Abstract: In this paper, we define the Voronoi diagram of a set of points (called sites) in the presence of obstacle line segments in R^2, which is called a *constrained Voronoi diagram*. When each site of the constrained Voronoi diagram is assigned a different weight, the diagram is called *constrained* and *weighted* Voronoi diagram. We show that the complexity of the constrained and weighted Voronoi diagram of n sites and m obstacles for $m = cn$ and constant c in R^2 is at least $\Omega(n^4)$ in the worst case. We also present an $O(n^4)$ algorithm to construct the diagram and the algorithm is worst case optimal in both time and space.

1 Introduction

The Voronoi diagram is an important geometric structure in computational geometry which has attracted a lot of attention [Aur84, Chew85, Fort86, Kirk79, Lee79, Sh78, Yap85]. Given a set of points (called sites) S in the plane, the *standard* Voronoi diagram of the set consists of a set of Voronoi cells, $\{V(s_i) \mid s_i \in S\}$, such that for any point $x \in V(s_i)$, $d(x,s_i) \leq d(x,s_j)$ for all $s_j \in S$, where $d(x,y)$ denotes the Euclidean distance between x and y.

Many variations of the standard Voronoi diagrams have been investigated. One of them is to consider the diagram in the presence of obstacles in the plane. When the distances of such a Voronoi diagram are measured by geodesic [Aro87, Tsin89], the diagram is called *geodesic Voronoi diagram*, when the distances of such a diagram are measured by straight-line, the diagram is called *constrained Voronoi diagram* [Chew87, Wang87, Wang89]. Another variation is to assign a different weight to each site of the standard Voronoi diagram. This diagram is called *weighted Voronoi diagram* [Aur84]. In this paper, we consider the Voronoi diagram of a set of weighted sites restricted by a set of obstacle line segments in the plane (called *constrained* and *weighted* Voronoi diagram). In this diagram, all points in a Voronoi cell must be 'visible' from the site associated with that cell. Consequently, the points of some subspaces may not be visible from any site due to the blockage of obstacles, and these subspaces do not belong to any Voronoi cell by definition. Hence, the diagram may not cover the entire space. Moreover, since each site is assigned a different weight, the boundary of a Voronoi cell may contain circular segment, and the cell itself may not be connected. These characteristics of a constrained and weighted Voronoi diagram complicate its structure. In particular, a Voronoi cell may contain several disjoint sub-Voronoi cells, which greatly increase the complexity of the diagram in terms of the number of edges and vertices.

The practical applications of the weighted Voronoi diagram were reported in several papers [Aur84, Bro78, Rhy73, Blu73,Bow81], and the disciplines of the applications include economics, geographics, communications, and biology. In the previous works, the plane is assumed without restrictions. However, the plane may contain obstacles in the real world. For instance, modeling a set of short-wave (microwave or laser) transmitters with verying strength, it is desirable to determine the regions in which a certain transmitter received best among the others. If the area contains buildings and mountains as obstacle, then the obstacles may block the waves of a transmitter and form several 'blank area'. For determining the active territory of an animal in a certain species (for example, lions) in biology, lakes, rivers, and higher mountains can be regarded as obstacles. Thus, our diagram is a better geometric model than the one in the previous papers.

The existence of obstacles makes a simple divide-and-conquer method and a sweep-line method inefficiency. We observe that there exists a close relationship between the diagram and the arrangement

[1] Department of Computer Science, Memorial University of Newfoundland, Canada.
[2] School of Computer Science, University of Windsor, Canada.

of a set of lines, where the lines are completely determined by the sites and the obstacles. By constructing a closest-site ordering list and a visible-site list for each edge in the corresponding arrangement, we are able to construct the diagram within the worst-case optimal time and space bounds.

This paper is organized as follows. For ease to describe our method, we first consider the diagram with the equi-weighted sites in Section 4. We then consider the sites of the diagram with different weights in Section 5. The proofs of all the lemmas are omitted in this version.

2 Preliminaries

We shall give several definitions and show some properties of constrained and weighted Voronoi diagrams in the plane. Let the obstacles be a set of line segments.

Definition: Let O be a set of obstacles. Let o^o be the open line segment of $o \epsilon O$, and let $O^o = \{o^o \mid o \epsilon O\}$. Two arbitrary points x and y in the plane are *visible* from each other in the presence of O iff $(\overline{xy} - \{x, y\}) \cap o^o = \emptyset$ for all $o^o \epsilon O^o$, where $\overline{xy}$ denotes the straight line segment spanning x and y, which is also regarded as a point set.

Definition: Let s be a site in the plane and $w(s)$ be the weight of s. The distance of s to an arbitrary point x, denoted by $d_w(x,s)$, is determined by $\frac{d(x,s)}{w(s)}$, where $d(x,s)$ is the Euclidean distance between s and x. Let s be a site with weight $w(s)$ in the presence of obstacle O in the plane. The distance between s and an arbitrary point x is determined by

$$d_{wo}(x,s) = \begin{cases} d_w(x,s) & \text{if } x \text{ and } s \text{ are visible from each other} \\ \infty & \text{otherwise} \end{cases}$$

Definition: Let S be a set of weighted sites and O be a set of obstacles in the plane. The constrained and weighted Voronoi diagram, denoted by $CWVor(S,O)$, is a set of Voronoi cells $\{V(s_i) \mid s_i \epsilon S\}$ such that $V(s_i) = \{x \epsilon R^2 \mid d_{wo}(x,s_i) < d_{wo}(x,s_j) \text{ and } d_{wo}(x,s_i) \neq \infty \bigvee s_j \in S,\ s_i \neq s_j\}$

Clearly, if the weights of the sites in S are all the same, then $CWVor(S,O)$ becomes constrained Voronoi diagram $CVor(S,O)$, and if the set of obstacles is empty, $CWCor(S,O)$ becomes weighted Voronoi diagram $WVor(S)$. The *boundary* of a Voronoi cell $V(s_i)$ is the closure of the point set $V(s_i)$. A *Voronoi edge* is a maximal straight line segment or a circular arc of the boundary of a Voronoi cell. The endpoints of the Voronoi edges of $V(s_i)$ are *Voronoi vertices*.

Property 1: A Voronoi edge must be one of the following three types: (1) a segment of the bisector of two sites, (2) a section of an obstacle, and (3) a segment of the line determined by a site and an endpoint of an obstacle.

Property 2: [Aur84] Let $S = \{s_1, s_2\}$ be a set of two weighted sites, and let weights $w(s_1) < w(s_2)$. Then, the Voronoi cell $V(s_1)$ is a closed disk with center $\frac{w^2(s_1)s_1 - w^2(s_2)s_2}{w^2(s_1) - w^2(s_2)}$, and radius $\frac{w(s_1)w(s_2)d(s_1,s_2)}{w^2(s_1) - w^2(s_2)}$. The Voronoi cell $V(s_2)$ is the complement of the above disc.

Property 3: Let $V(s_i)$ for $s_i \epsilon S$ be a Voronoi cell of $WVor(S)$, then the boundary of the cell consists of circular arcs and/or line segments.

3 The lower bound of a CWVor(S,O)

We have the following lower and upper bounds of the diagram in terms of vertices and edges.

Lemma 3.1: The number of edges and vertices of $CWVor(S,O)$ may be up to $O((mn+n^2)^2)$ in the worst case, where n is the number of sites and m is the number of obstacles.

Lemma 3.2: The maximum number of edges and vertices of $CWVor(S,O)$ is bounded by $O((mn+n^2)^2)$, where n is the number of sites and m is the number of obstacles.

4 Finding the constrained Voronoi diagram CVor(S,O)

It is easy to prove that the arrangement of $A(O' \cup B \cup T)$ contains $CVor(S,O)$, where O' is the set of lines, each of which extends an obstacle (called a type-1 line), B is the set of perpendicular bisectors of n sites (type-2 lines), and T is the set of lines, each of which is determined by a site and an endpoint of an obstacle (type-3 line). The following algorithm is first given in a high-level description. Then, some important steps of it are described in detail.

Algorithm Find-CVor(S,O)

Step 1. (Generating the three types of lines to be arranged.)
Extend each line segment in O, and let O' denote these lines; Find the perpendicular bisectors of every pair of the sites in S, let B denote these bisectors; Draw a straight line passing through a site s and an endpoint p of obstacle o for all $s \in S$ and all $o \in O$. Let T denote these lines.
Step 2. Construct the arrangement $A(O' \cup B \cup T)$.
Step 3. (Mask or delete some edges in the arrangement not appearing in $CV_{or}(S,O)$.)
(a) Let o' be the line extending an obstacle $o \epsilon O$. Mask $o'' = o' - o$ from $A(O' \cup B \cup T)$ for all $o \epsilon O$. (b) Let line $t \epsilon T$ be determined by a site s and an endpoint p of an obstacle. If line segment $\overline{sp}$ is crossed by an obstacle, then delete t from $A(O' \cup B \cup T)$. Otherwise, mask $t - \overline{pp'}$, where p' is the crossover point of ray $\vec{sp}$ and its first encountered obstacle (which might not exist). Do this for all $t \epsilon T$. (c) Let $b \epsilon B$ be the perpendicular bisector of sites s and s'; b' be the portions of b not visible from at least one of s and s'. Mask b' for all $b \epsilon B$ from $A(O' \cup B \cup T)$.
Step 4. For each unmasked edge of $A(O' \cup B \cup T)$, find the sites visible to this edge.
Step 5. For each unmasked edge of $A(O' \cup B \cup T)$, determine the site(s) closest and visible to this edge. Determine if this edge appears on the boundary of the Voronoi cell of that closest site. Mask the unmasked edges if it does not.
Step 6. Delete all the masked edges of $A(O' \cup B \cup T)$ to obtain $CVor(S,O)$.

Details of the steps

Step 1 is straight-forward. Step 2 can be done by the algorithm proposed by Edelsbrunner et al. [Edel86]. Step 3 can be done by traversing the lines in $A(O' \cup B \cup T)$. The following two steps are crucial for the time complexity of the algorithm.

Step 4 finds a list of sites visible to an edge for every edge in $A(O' \cup B \cup T)$. If a brute-force method is applied, then Step 4 may take $O(n^6)$ in worst case. Let $vis(e)$ denote the set of all sites visible from edge e. Let e and e'' be two edges in $A(O' \cup B \cup T)$ such that e and e'' share a common endpoint v and e'' is *successor* of e rotating at v clockwise. Let t (respectively t') be the base line of e (respectively e''), l and s (l' and s') be the *determiners* of t (t'). l is *up* if l is in the halfplane (determined by t) not containing e, l is *down*, otherwise.

Lemma 4.1: (1) Suppose that none of t and t' is of type-3, then $vis(e) = vis(e'')$. (2) Suppose that exactly one of t and t' is of type-3. Then, (i) if t is of type-3 and the determiner l is up, then $vis(e'') = vis(e) - \{s\}$ if e is unmasked; $vis(e'') = vis(e)$ if e is masked. (ii) if t' is of type-3 and the determiner l' is down, then $vis(e'') = vis(e) \cup \{s\}$ if e is unmasked; $vis(e'') = vis(e)$ if e is masked. (iii) in the remaining cases, $vis(e'')=vis(e)$. (3) Suppose that both t and t' are of type-3. Then, (i) if both l and l' are down, then $vis(e'')=vis(e) \cup \{s'\}$ if e'' is unmasked; $vis(e'')=vis(e)$ if e'' is masked. (ii) if l is up and l' is down, then $vis(e'')=vis(e) - \{s\}$ if e'' is masked and e is unmasked; $vis(e'')=vis(e) \cup \{s'\}$ if e'' is unmasked and e is masked; $vis(e'')=vis(e) - \{s, s'\}$ if e'' and e are unmasked; $vis(e'')=vis(e)$ if both e and e'' are masked. (iii) if l is down and l' is up, then $vis(e'')=vis(e)$. (iiii) if both l and l' are up, then $vis(e'')=vis(e) - \{s\}$ if e is unmasked; $vis(e'')=vis(e)$ if e is masked.

The above lemma immediately implies that given $vis(e)$ for an edge e, the list of all visible sites of the successor of e can be determined in O(1) time. It is obvious that the visible-site list of each edge on an obstacle can be determined by the type-3 edges incident at the obstacle. Therefore, the visible-site list for each edge in $A(O' \cup B \cup T)$ can be determined by first finding the visible-site list for the edges on the obstacles, then finding the visible-site list for the rest edges. The time and space complexities of the above process are $O((mn)^2)$.

Step 5 is to mask all the edges in $A(O' \cup B \cup T)$ which lie inside the Voronoi cells (thus they do not appear in $CVor(S, O)$) and to determine the Voronoi cell(s) each unmasked edge belongs to. To do so, we shall determine the visible site(s) closest to each unmasked edge. This could be easily accomplished by a brute-force method in $O((mn)^2 n)$ time. The following observation indicates a faster method.

Definition: Let S_i be a list of n sites associated with the i-th edge e_i of l. S_i is said to be in $closest - site\ ordering$, if the first site in the list is the one closest to e_i (disregard the obstacles), and the second site in the list will be the one closest to e_i if the first site is not taken into account (for example, it is not visible from e_i). In general, the k-th site for $1 \leq k \leq n$ will be the one closest to e_i if the first $k-1$ sites is not taken into account. A set of lists associated with line l is said to be in $closest - site\ ordering$ if each list in the set is in $closest - site\ ordering$.

Lemma 4.2: Let B be the set of bisectors determined by n sites. Let l be an arbitrary line crossing B. The set of closest-site ordering lists of l can be determined in $O(n^2)$ time.

Lemma 4.3: The closest site(s) for every edge of $A(O' \cup B \cup T)$ can be found in $O(n^4)$ time.

Let r be a region of $A(O' \cup B \cup T)$. Then, r belongs to $V(s_i)$ of $CVor(S, O)$ iff r is visible from s_i and the points on r are closer to s_i than to any other visible site in S. This implies that the visible-site list and the closest-site ordering list of each edge of r must contain s_i and s_i must be the closest and visible site to these edges. It can then be easily determined if an edge of r belongs $V(s_i)$.

Step 6 deletes all the masked edges in $A(O' \cup B \cup T)$, which takes at most $O((mn)^2)$ time by examining all the edges.

Theorem 4.1: Algorithm **Find-CVor(S,O)** produces $CVor(S, O)$ in $O((mn+n^2)^2)$ time, which is worst case optimal in both time and space.

5 Finding the constrained and weighted Voronoi diagram

The difference between $CVor(S, O)$ and $CWVor(s, O)$ is that the perpendicular bisectors in $CVor(S, O)$ are replaced by the circular arcs in $CWVor(S, O)$. If we can deal with this difference, then the same algorithm for constructing $CVor(S, O)$ can be directly applied to finding $CWVor(S, O)$.

After inspecting **Find-CVor(S,O)** , we find the following three key points must be dealt with: (1) how to insert a set of circles into arrangement $A(O' \cup T)$ to form $A(O' \cup C \cup T)$, (2) how to find the visible-site list for each edge in $A(O' \cup C \cup T)$, (3) how to find the closest-site ordering list for each edge in $A(O' \cup C \cup T)$. Points (1) and (2) can be easily solved. For point (3), we need the following lemma.

Lemma 5.1: Let $A(C)$ be the arrangement of $\frac{n(n-1)}{2}$ circles determined by n weighted sites in S. Let l be an arbitrary line crossing $A(C)$. Then, the closest-site ordering lists of l can be found in $O(n^2)$ time.

It is not difficult to find $CWVor(S, O)$ by an algorithm similar to **Find-CVor(S,O)**.

Theorem 5.1: Let S be a set of n weighted sites and O be a set of obstacles in the plane. $CWVor(S, O)$ can be found in $O((mn + n^2)^2)$ time and space, and this is worst-case optimal.

References

[Aur84] Aurenhammer F. and Edelsbrunner H., (1984), "An optimal algorithm for constructing the weighted Voronoi diagram in the plane", *Pattern Recognition* , 17(2), pp.251-257.

[Edel86] Edelsbrunner H.,O'Rourke J., Seidel R., (1986), "Constructing arrangements of lines and hyperplanes with applications", *SIAM J. Computing.* , Vol.15, No.2, pp.341-363.