ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol.2, Issue 12, December 2013

Fortune's Method: An Efficient Method For Voronoi Diagram Construction

Ms.Punam Marbate[1], Ms.Reetu Gupta[2]

Department of Computer Science & Engineering, Rajiv Gandhi College of Engineering & Research, Nagpur, India[1]

Department of Computer Science & Engineering, Priyadarshani Indira Gandhi College of Engineering, Nagpur, India[2]

Abstract: This Paper briefly describes the Fortune's method i.e. Fortune's sweepline algorithm for constructing voronoi diagram. There are a variety of algorithms available to construct Voronoi diagrams. One popular method is the incremental algorithm that adds a new site to an already existing diagram. In 1985, Steve Fortune developed a plane-sweep algorithm which is more efficient in time than any incremental algorithm. There are many approaches to constructing Voronoi diagrams. Some methods are more efficient in terms of time than others.
Here, we have tried to explore the aspects regarding fortune's method.

Keywords: Voronoi Diagram, Dirichlet Tesselation, Delaunay Triangulation

1. INTRODUCTION

The Voronoi diagram is a very versatile and well-studied geometric construct in computational geometry [1, 2, 3]. A traditional distance measurement of Voronoi diagram is the Euclidean distance. Voronoi Diagram (VD) is a type of versatile geometric data structure. It has been widely used in physics, astronomy, geographical information systems, computer graphics, image processing, and robotics. Basically, Voronoi diagram is the graph theoretic approach in Robot motion planning. Some of the limitations that might be attributed to this approach are:

- Time complexity of constructing such connected network (graph) as the robot's field of operation expands.
- Need of global map of the environment for which the graph is constructed.
- Uncertainty introduced by the application of moving/movable objects.

In this paper, the different methods for construction of voronoi diagram described, In second section, the history and current application of voronoi diagram described. Section three explores the basic method for constructing voronoi diagram and the Fortune's method for constructing voronoi diagram.

1.1 Introduction to Voronoi Diagrams / Dirichlet Tesselations:

A Voronoi diagram is a geometric structure that represents proximity information about a set of points or objects. Given a set of sites or objects, the plane is partitioned by assigning to each point its nearest site. The points, whose nearest site is not unique, form the Voronoi diagram. That is, the points on the Voronoi diagram are equidistant to two or more sites. So for a set S of n sites:

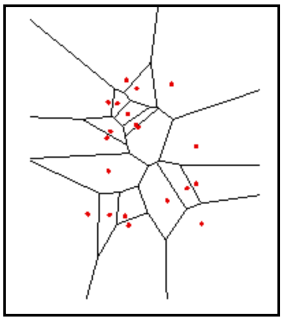

Fig 1: Example of a Voronoi Diagram

The partition of the plane into blocks of points with the same nearest site or sites. Voronoi diagrams were first discussed by Peter Lejeune-Dirichlet in 1850. But it was more than a half of a century later in 1908 that these diagrams were written about in a paper by Voronoi, hence the name *Voronoi Diagrams*. The Voronoi cells/polygons are sometimes also called *Thiessen Polytopes or Dirichlet Regions*.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol.2, Issue 12, December 2013

1.2 Delaunay Triangulation:

The Delaunay triangulation of a point set is a collection of edges satisfying an "empty circle" property: for each edge we can find a circle containing the edge's endpoints but not containing any other points. The Delaunay triangulation is the dual structure of the Voronoi diagram in R^2. By dual, we mean to draw a line segment between two Voronoi vertices if their Voronoi polygons have a common edge, or in more mathematical terminology: there is a natural bisection between the two which reverses the face inclusions. The circumcircle of a Delaunay triangle is called a Delaunay circle.

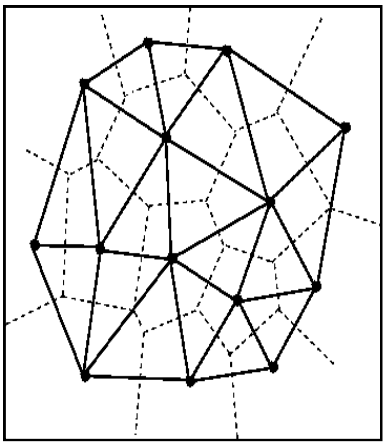

Fig 2: Delaunay triangulation, on top of the Voronoi diagram (in dotted lines)

1.3 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 3, the point p is closer to p1 than to any other enumerated points. Also note that p0, which is on the boundary between p1 and p3, is equidistant from both of those points. As an analogy, imagine a Voronoi diagram in R2 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 R that can be reached from the boat from island x before any other boat be associated with island x. The region of points associated with island x is called a Voronoi region.

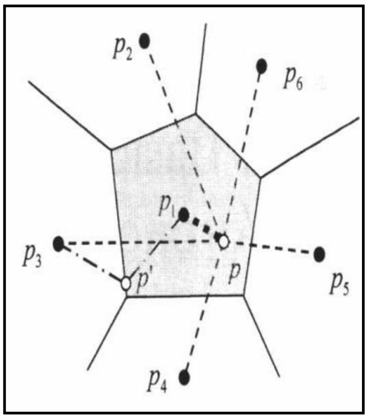

Fig 3: Voronoi Diagram

For modelling 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 archaeology), and patterns of urban settlements (geography).

2. HISTORY OF VORONOI DIAGRAM

2.1 History - The recorded history of Voronoi-like diagrams goes back to the 17th century, when Descartes drew Voronoi-like diagrams to describe fragmentation of space by the stars, although he does not describe rules that would allow us to credit him with inventing the diagram outright. [4, 6][2,203-204] That honour goes to Dirichlet, who described the diagram formally in 1850. Since then the diagram has been independently rediscovered many times across many fields of research, and as a result has acquired a number of names.Despite their long history, research into Voronoi diagrams are still very much ongoing, spurred on by the influence of computational geometry, which began exploring them in 1975. [7]

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol.2, Issue 12, December 2013

2.2 Pre-history - Voronoi diagrams in nature Voronoi-like forms occur in nature, most often in growth patterns, for example figure 3 shows orets growing from sites in the head of a daisy and meeting to form Voronoi regions. [4, p. 470] The sites are arranged in a spiral pattern, which is accentuated by the Voronoi edges. Voronoi-like forms may also be found in spherules growing patterns, sand parapets around mouth-breeder fish territories [4, p480], giraffe and turtle markings, mud crack patterns and in large scale rock formations such as the Giant's Causeway. These occurrences make it probable that their study extends earlier than that documented, even before Descartes, as early philosophers spent some time looking at patterns in nature. [4, p6].

2.3 Current applications:

Voronoi diagrams are a truly cross-disciplinary area of research. This makes the absence of literature about Voronoi diagrams in music conspicuous – they also described in some papers about how they are applied within other fields of study, which lead to some ideas about how they may be applied to music.

2.3.1 Topological analysis in medicine - cancer diagnosis Computational diagnostic tools allow automated identification of cancer cells, with the possibility of faster and more objective decision making than by human pathologists. As described by Demir & Yener [8], Voronoi diagrams may aid this diagnosis of cancer by describing the topology of cells within tissue samples. Figure 4 shows such a Voronoi diagram applied to cells found through automatic feature extraction. Similar techniques could be applied in the field of musical analysis, to examine the topology of a piece of music. Without wanting to be crass, an analogy might even be drawn between the relationships between benign and malignant tumours and dissonant and resonant musical structures.

2.3.2 Data visualisation – treemaps -The treemap was invented by Ben Shneiderman in 1990 as a compact, spatial visualisation of a tree structure. Each rectangle within a sector represents a company, coloured according to rising or falling value. We can gain some insights into the comparative performance and makeup of the sectors, but such a treemap has some problems. The aspect ratios of the individual rectangles vary enormously, leading to difficulty of comparison by the human eye.

Further, rectangle edges either side of a boundary often meet, leading to false intersections and therefore unclear segmentations. Efforts to correct the former problem tend to exasperate the\ latter. [10, p. 2] Balzer et al [10] introduce a method of constructing treemaps as hierarchical Voronoi diagrams, consisting of polygons rather than rectangles. In order to properly reject the data spatially, their Voronoi diagrams have sites that are weighted and centroidal 5. Each site is weighted so that it produces an area of the correct proportion to the value mapped to it and centroidal so that the polygons are easily compared with one another by the human eye. Their results are visually stunning and overcome the problems they identify with rectangular treemaps. Producing a weighted centroidal Voronoi diagram is computationally expensive, involving much iteration.

Lerdahl & Jackendo_'s General Theory of Tonal Music [11] describes a tree structure behind tonal music of groups within groups, for example notes within phrases within themes within musical works. Perhaps a Voronoi treemap could be constructed around such groups. More generally, the idea of well spaced sites of central gravity within their Voronoi region could translate to musical structure.

3. Construction of a Voronoi Diagram

There are a variety of algorithms available to construct Voronoi diagrams. One popular method is the incremental algorithm that adds a new site to an already existing diagram. In 1985, Steve Fortune developed a plane-sweep algorithm which is more efficient in time than any incremental algorithm. There are many approaches to constructing Voronoi diagrams. Some methods are more efficient in terms of time than others.

3.1 Basic method for voronoi construction

The simplest method for construction of voronoi diagram is a geometric construction algorithm using perpendicular bisectors. In terms of time efficiency,

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol.2, Issue 12, December 2013

it is VERY inefficient but it is easy to understand and helps to solidify the understanding of voronoi diagrams. There are some more common and MUCH MORE efficient algorithms are available for construction of voronoi diagram.

We know that a voronoi edge represents points that are equidistant from the nearest two sites. So let's start with just two points on the graph as seen in figures below. The voronoi edge for these two points would be the perpendicular bisector of the segment joining the two sites.

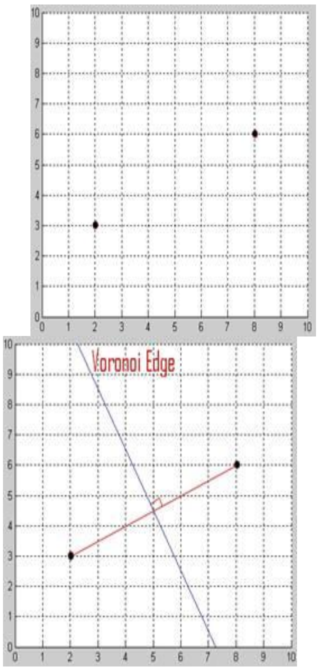

Fig 4: Step 1- Basic method of voronoi construction

If you add a third point to the diagram, you must calculate the perpendicular bisector between each of the three points: point1 to point 2, point 1 to point 3, and point 2 to point 3. The next step is to remove each section of the perpendicular bisector lines that violate the equidistant rule. You should now be left with the voronoi diagram for the three sites as seen in figure 5 below.

Adding yet another point to the diagram requires following the same method described above. As you might be able to tell, as the number of sites increase, the time it takes to compute the voronoi diagram using this method increases significantly.

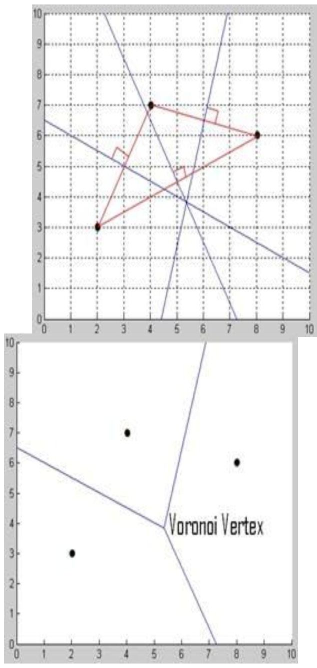

Fig 5: Step 2-Basic method of voronoi construction

3.2 Construction of Voronoi diagrams using Fortune's method

3.2.1 Introduction:

Voronoi diagram belong to classical problems of the computational geometry. Its origin dates back into 1850 when it was considered by Dirichlet. The rigid mathematical fundamental was given by Voronoi in 1908. Let us have points p_i, $0 < i <= n$ in n-dimensional space P. The set of all points with property that every point inside the cell is closer to point p_i than to any other point from P represents the Voronoi cell (Figure 1). The union of all Voronoi cells is known as Voronoi diagram.

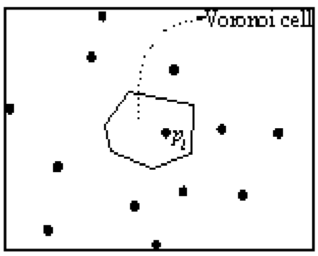

Fig 6: Voronoi cell

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol.2, Issue 12, December 2013

As seen from Figure 1, the construction of Voronoi diagram consists of repetitive subdivisions of the space determined by the input points into subspaces to meet the Voronoi criteria: In many applications, Voronoi diagrams are already the final solution. For example, study of behavior and maintenance of live creature, which are depended on number of neighbors with whom they are fighting for food and lightness, is exactly what Voronoi diagram expresses. However, having a Voronoi diagram, many important geometric features like searching for the closest neighbor, Delaunay triangulation, searching for the largest empty circle, minimum spanning tree (Euclidean tree), can be determined in the linear time.

The first reliable algorithm for construction has been published by Green and Sibson [GREE77], although the Voronoi diagrams have been known for a long time. This approach used incremental method and worked in $O(n^2)$ time. The first algorithm requiring $O(n \log_2 n)$ was suggested by Shamos and Hoey (1975) [PREP85] using divide and conquer approach. However, its implementation is complicated. In 1985, Fortune presented more elegant approach and it is described in [BERG97, O'ROU93]. In this paper, we consider primarily the data structures needed for the construction. In 1986, Edelsbruner and Seidel discovered beautiful connection between Voronoi diagram and convex hulls in one higher dimension [O'ROU93]. This method has some beautiful properties and it is becoming more and more popular.

3.2.2 Fortune's method:

In 1985, Fortune invented an interesting algorithm for construction of Voronoi diagram. He used the sweep-line strategy implemented by many algorithms of computer graphics and computational geometry. The idea of the algorithm is very simple and its time complexity is $O(n \log_2 n)$ in the worst case, where n is the number of input points. In the continuation, an idea of the algorithm is shortly given; details can be find in [BERG97].

Fortune has expanded his observation in 3D. He put coins above all points from set P. All coins are slanted for an angle of 45°. These coins are scanned with a scanning plane π, which is also slanted for an angle of 45° to the xy plane (Fig 7).

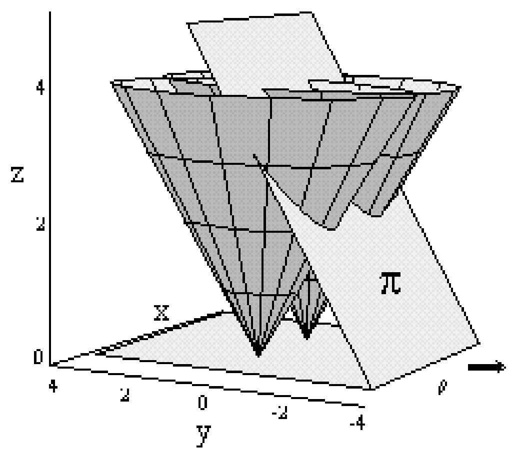

Fig 7: During the process of scanning, plane π intersects coins

In Fortune's method, we are interested in intersections of the scanning plane with coins. These intersections represent a curve of parabolic arcs or parabolic front. This curve has the property that joints of parabolic arcs lie on intersection of two neighboring coins. The parallel projection of these two coins on xy-plane represents exactly bisector on which Voronoi edge lies. At first, the method seems very complicated, but Fortune has had the fortune on his side. If the coins and sweep plane are considered in xy-plane, we get the sweep line and parabolic front. In Figure 7 it is easy to see why the curve of the parabolic front is so important. Namely, Voronoi diagram is fully constructed above the parabolic front and below we do not know anything about it.

From Fig 8, it can be seen that a topological structure of the parabolic front is changed by sliding the scanning line over xy-plane.

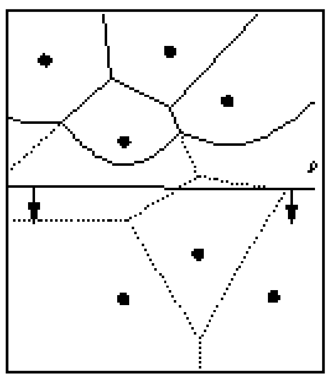

Fig 8:2D representation of Fortune's method; Voronoi diagram is constructed only above parabolic front

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol.2, Issue 12, December 2013

3.2.3 Point event

The point event is handled when a new parabolic arc appears on parabolic front. This event happens exactly when a new arc appears on the parabolic front, or more precise, when the scanning line ℓ encounters a point p_i from set *P*. Then, the scanning plane π hits the coin defined above point p_i for the first time. Because the coin and the scanning plane are slanted for the same angle, the intersection between them appears as a line on the coin surface. This line is actually a half-edge, but if it is projected onto xy-plane we can consider it a degenerated parabola with zero width. This degenerated parabola starts to open as the scanning plane π slides over the space. When a new arc comes across, it splits the existed arc into two parts and becomes a new member of the parabolic front (Fig 9).

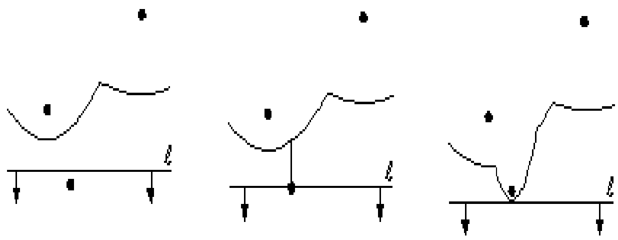

Fig 9: When point event is encountered a new arc appears on the parabolic front

A joint point of two arcs on parabolic front describes the Voronoi edge.

3.2.4 Circle event

Circle event happens when a parabolic arc shrinks to a point and disappears from the parabolic front (Fig 10). Let β_j be the disappearing arc and β_i and β_k are two neighboring arcs before β_j disappears. These arcs are then defined by three different points p_i , p_j and p_k . At the moment when β_j disappears, all three arcs pass through a common point *q* (Fig 10). The distance between *q* and the scanning line is the same as the distance from *q* to all three points p_i , p_j and p_k . These three points define a circle, with its center at point *q* which represents the Voronoi point. The lowest point of this circle touches the scanning line and represents the circle event.

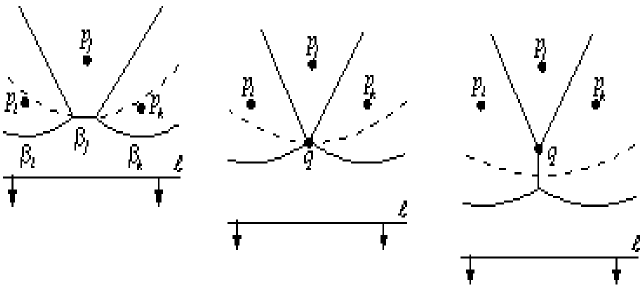

Fig 10: Parabolic arc disappears from the parabolic front

3.2.5 Fortune's algorithm

Up to now, we have met all fundamental parts of Fortune's method and so we can think about an implementation. However, let us say just a word about termination condition. When all points are behind the scanning line and all events have been handled, Voronoi diagram is not fully constructed yet. We have to construct the remaining half-lines. For this purpose, the whole diagram is surrounded by a box (Fig 11) on which border all half-edges terminate.

The following algorithm summaries the consideration done up to now and shows the core of the Fortune's method.

VORONOI_DIAGRAM(*P*)
Input: A set P of point sites in the plane
Output: The Voronoi diagram Vor(P) given inside a bounding box in a doubly-connected edge list structure
{
  Initialize the event tree Q with all point events
  while Q is not empty **do**
  {
    Consider the event with largest y-coordinate in Q
    if the event is a point event, occurring at site p_i
    then HANDLE_POINT_EVENT(p_i)
    else HANDLE_CIRCLE_EVENT(β_l)
    Remove the event from Q
  }
  Put Voronoi diagram into a box
}

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol.2, Issue 12, December 2013

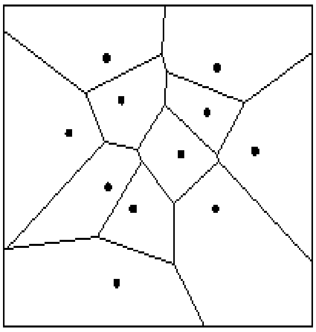

Fig 11: Completely defined computer represented Voronoi diagram does not contain any half-edges

REFERENCES

[1] O. Aichholzer, F. Aurenhammer, D. Z. Chen, D. T. Lee and E. Papadopoulou, Skew Voronoi Diagram, Int'l J. Comput. Geometry and Applications, 9(3), June 1999, pp. 235-248.
[2]H. Edelsbrunner, Algorithms in combinatorial Geometry, Springer-Verlag New York, Inc.,New York, NY, 1987.
[3]D. T. Lee,Two Dimensional Voronoi Diagram in the Lp-metric J. ACM, Oct. 1980, pp.604-618.
[4]Okabe, Boots, Sugihara & Chiu; Spatial Tessellations, Concepts and Applications of VoronoiDiagrams, 2nd Ed., 2000
[5]J.-R.Sack&J.Urrutia; Handbook of Computational Geometry, 2000
[6]Steve Fortune, A Sweep line Algorithm for Voronoi Diagrams, Algorithmic, pp. 153{174, 1987)
[7]Shamos & Hoey, Closest-point problems, In Proceedings 16th IEEE Symposium on Foundations of Computer Science, pp 151-162,197
[8] C. Demir & B. Yener, Automated cancer diagnosis based on histopathological images: a systematic survey, Technical report, Rensselaer Polytechnic Institute, Department of Computer Science, TR-05-09, 2005
[9] Ward, Rohrhuber, Olofsson, McLean, Griths, Collins & Alexander, Live Algorithm Programming
and a Temporary Organisation for its Promotion, 2004
[10] M. Balzer, O. Deussen, Voronoi Treemaps, IEEE Symposium on Information Visualization, 2005
[11] F. Lerdahl and R. Jackendo_, A Generative Theory of Tonal Music,. MIT Press, Cambridge,MA, 1983
[12] M. Balzer, O. Deussen, C. Lewerentz, Voronoi Treemaps for the Visualization of Software Metrics, ACM Symposium on Software Visualization, 2005
[13] Hamanaka & Hirata, Applying Voronoi Diagrams in the Automatic Grouping of Polyphony, 2002
[14] Franz Aurenhammer, Voronoi diagrams - a study of a fundamental geometric data structure, ACM Computing Surveys, Volume 23, Number 3, pages 345-405, 1991,14