Open source implementation of the Multiplicatively Weighted Voronoi Diagram as a TerraView plugin

Eng. Maurício Carvalho Mathias de Paulo[1,2]
Dr. Antônio Miguel Vieira Monteiro[2]
Dr. Eduardo Gerbi Camargo[2]

[1]Diretoria de Serviço Geográfico – DSG
Quartel General do Exército, Bloco "F", 2o Piso, Ala Norte
CEP:70630-901 – SMU – Brasília – DF, Brasil

[2]National Institute for Space Research – INPE
Caixa Postal 515 – 12245-970 – São José dos Campos - SP, Brasil

{mauricio, miguel, eduardo}@dpi.inpe.br

Abstract. *Given a point set the Voronoi diagram associates to each point all the locations in a plane that are closer to it . This diagram is often used in spatial analysis to divide an area among points. In the ordinary Voronoi diagram the points are treated as equals and the division is done in a purely geometrical way. A weighted Voronoi diagram is defined as an extension of the original diagram. The weight given usually relates to some variable property of the phenomenon represented by each point. The weighted distance is then computed as a function that depends both on the weight and on the euclidean distance. This article describes a multiplicatively weighted Voronoi diagram implementation as an open source plugin for TerraView. The algorithm used computes an approximation of the diagram using multipolygons to represent each point's area. This choice avoids the voids that might appear in most of the implementations that focus on finding the intersections and scales well in memory.*

1. Introduction

The Epidemiology and Information Coordination (Coordenação de Epidemiologia e Informação - CEInfo) of São Paulo's Municipal Health Bureau (Secretaria Municipal de Saúde de São Paulo-SP/Brazil) is researching methods to estimate supply and demand of public health care in the city. One of the parameters in the analysis is the accessibility, translated as the distance each person has to walk until reaching the nearest health center.

The health centers are represented as points and the number of treatments in each specific area is measured yearly. Because of the lack of information of where each person come from to be treated it's only possible to estimate an area from where the majority probably came. In this case a spatial partitioning of the study area among the points is a viable alternative.

The Voronoi Diagram is a method to divide the space among a set of points assigning an area to each. The main property of this division is that each area represent the space where the point is the nearest neighbor [Boots 1986]. There are a few extensions on the Voronoi Diagram that can assign areas based on some property of the points,

thus providing a way to distinguish each point by their representativeness. One of these extensions is the Multiplicatively Weighted Voronoi Diagram.

The Multiplicatively Weighted Voronoi Diagrams have been used to evaluate student allocation in educational centers [Karimi et al. 2009], logistic district attribution [Novaes et al. 2009] and computation of dominance area of health centers [Rezende et al. 2000]. The later suggests that the diagram can be used in problems similar to the estimation that CEInfo is researching.

This article first presents the theory behind Multiplicatively Weighted Voronoi Diagrams and the improvements over the Ordinary Diagram (Section 3). In Section 4 the algorithm, user interface and data structure chosen are presented. In Section 5 some preliminary results are presented suggesting its application in geographic analysis.

2. Related work

The Ordinary Voronoi Diagram has some well known algorithms that most implementations are based on. The main algorithms are the incremental insertion, divide and conquer, plane-sweep construction and embedding in a three dimensional space [Aurenhammer 1991]. Many of these implementations are available in open source libraries.

The Multiplicatively Weighted Voronoi Diagram is not necessarily convex nor continuous [Gahegan and Lee 2000]. This makes most of the popular algorithms used to build the Ordinary Voronoi to be impractical for the weighted Voronoi. Some researchers used grid approximation to build the weighted regions [Dong 2008], usually reaching slow results when compared to the Ordinary Voronoi vector-based algorithms. Some approaches uses the bisector defined by two points and finds every possible arc intersection and then reconstructs the diagram evaluating which ones are dominance area's boundaries [Lee and Gahegan 2002]. The optimal algorithm was designed using an spherical inversion in the three dimensional space [Aurenhammer and Edelsbrunner 1984].

As this paper is written, there are many open source libraries with Ordinary Voronoi Diagram implementations, but there is none for the Multiplicatively Weighted Voronoi Diagram. *CGAL* (Computational Geometry Algorithms Library) has an open source implementation of the Additively Weighted Voronoi Diagrams [Karavelas and Yvinec 2002] but as the diagram works with parabolas instead of circles, the data structures and algorithms are not reusable. Some algorithms and implementations are available to compute gridded approximations of the algorithm [Ohyama 2011] but this limits the applications when analyzing geographic vector data.

TerraView (An open source geographic information system application based on TerraLib [Câmara et al. 2008]) has an Ordinary Voronoi implementation. This article describes a vector-based Multiplicatively Weighted Voronoi implementation that is based on the incremental construction algorithm. To solve the problem of the circular arcs that are used in the diagram two approaches are suggested, one for compatibility with polygon implementations and one that extends the concept of polygons using the CurvePolygon data type [Stolze 2003].

3. Weighted Voronoi Diagrams

3.1. Ordinary Voronoi Diagram

Given a set of generator points $S = \{p_1, p_2, ..., p_n\}$, the Voronoi diagram built from the set represents the regions of the plane where each point is closer than all of the others in the set, as formalized in the Equation 1 [Boots 1986]. Therefore, the diagram assigns to each point an area where it is the nearest neighbor [Aurenhammer and Klein 2000].

The boundary of each area is defined by finding the equation for x where the distance to both generator points are equal (Equation 1). Thus the boundary of each dominance area is a line that represents the end of one generator's dominance and the beginning of the other. Figure 1 shows an Ordinary Voronoi diagram example when applied to a given set of generator points.

$$P_i = x | d(x, p_i) \leq d(x, p_j); j \in S, j \neq i \tag{1}$$

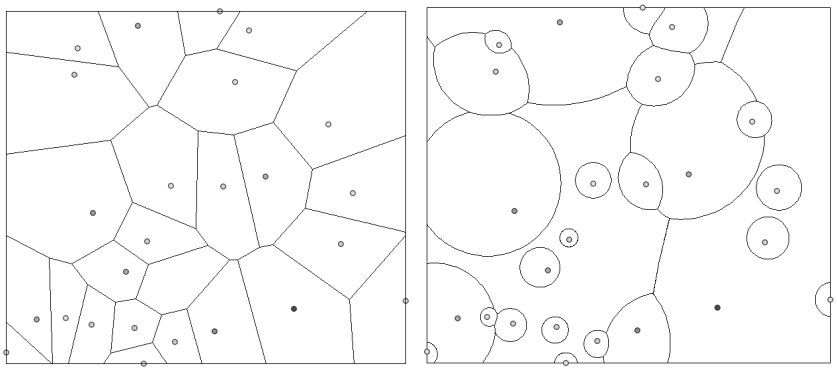

Figure 1. Ordinary Voronoi Diagram and Multiplicatively Weighted Voronoi Diagram

3.2. Multiplicatively Weighted Voronoi Diagram

As an extension, a Weighted Voronoi diagram is defined by analogy, by changing the euclidean distance $d(x, p_i)$ for the weighted distance $d_w(x, p_i)$ [Aurenhammer and Edelsbrunner 1984]. Equation 2 shows the definition of the region of dominance WP_i of a given generator p_i [Boots 1986]. In geographic applications the weight should be a property of the phenomenon mapped by the point p_i, which is considered in the spatial partitioning process [Boots 1986].

$$WP_i = x | d_w(x, p_i) \leq d_w(x, p_j); j \in S, j \neq i \tag{2}$$

Using the distance function as the ratio of the euclidean distance by the generator's weight (Equation 3) a Multiplicatively Weighted Voronoi Diagram is obtained.

$$d(x, p_i) = \frac{|x - p_i|}{w_i} \tag{3}$$

As an example, consider the time taken to reach a coordinate x in a plane. Consider the generator point p_i as a vehicle with constant speed. The time taken to reach x can be computed as $\frac{|x-p_i|}{speed}$. The higher the speed of the vehicle the lower the time distance. This function would then consider a vehicle with distance 100 and speed 10 as nearer than a vehicle with distance 50 and speed 1.

This diagram's dominance areas are bounded by pieces of Apollonius' circles [Aurenhammer and Edelsbrunner 1984]. This is a straight result since the ratio of the distances to the two generator points is constant.

$$d(x,p_1) = d(x,p_2) \rightarrow \quad \frac{|x-p_1|}{|x-p_2|} = \frac{w_1}{w_2} \tag{4}$$

A simple proof arises from setting $p_1 = (0,0)$ and $p_2 = (0,a)$ and $\frac{w_1}{w_2} = c$. Equation 5 is a circle's equation. This result was generalized for coordinates in the plane as is used to compute the boundaries of the diagram [Aurenhammer and Edelsbrunner 1984].

$$\frac{|(x_i,x_j)-(0,0)|}{|(x_i,x_j)-(0,a)|} = c \rightarrow x_i^2 + x_j^2 = c \cdot (x_i^2 + (x_j-a)^2)$$
$$(1-c)x_i^2 + (1-c)x_j^2 - 2acx_j + ca^2 = 0 \tag{5}$$

In the example of the vehicles in the plane, the vehicles are the generator points p_i. The coordinates x where the distance is equal are the places where two vehicles reach the same place at the same time. Therefore this diagram's dominance areas represent the area where each vehicle is the one that reaches every place faster than any other. This could be called the Quickest Neighbor Diagram and might have applications on search and rescue for emergency situations. This is a simple application where diagram is used as an event modeling technique.

There are also applications where the diagram is used as an area assignment method [Boots 1986]. Due to the fact that the distance's ratio is dimensionless so is the weight's ratio. Thus the weights can be any attribute in any unit. This is particularly useful in geographic applications in which the weights usually represent a non spatial parameter that is rarely in units compatible with the reference system of the point set.

The diagram can be interpreted both as a weighted area assignment, where generator points with bigger weights area assigned to bigger areas, or as a growth model, where the areas grow in different ratios starting from the generator points [Boots 1986]. Figure 1 shows a Multiplicatively Weighted Voronoi Diagram compared to an Ordinary Voronoi Diagram.

4. Implementation

4.1. Interface

The parameters necessary to compute a Multiplicatively Weighted Voronoi Diagram are the coordinates of each point and it's weights. As the project aims to be easy for future researchers, an user interface was created and integrated as a plugin on TerraView. Figure 2 shows the interface designed with *QtDesigner* (a cross-platform tool for designing

graphical user interfaces which is part of the Qt SDK) for this purpose. The code used the library *QT* 3.3.8 [Jasmin and Summerfield 2005] for compatibility with TerraView.

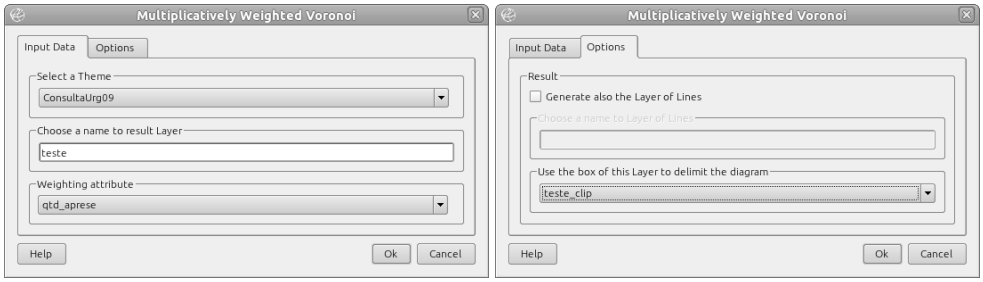

Figure 2. Interface developed for the TerraView's plugin

The first tab shows the main parameters that the user is required to enter in order to build the diagram. TerraView works tightly coupled with many database engines so the plugin lists the available options of input point layers from the currently used database connection. When the input theme is chosen the plugin lists the available attributes. An output layer name is also required for the polygon layer that is created to store the dominance areas.

4.2. Algorithm

Given two generator points p_1 and p_2 and their respective weights w_1 and w_2 the space is divided in two areas. As described in Equation 5, the weighted distance is used to divide the plane using the boundary of an Apollonius Circle. Therefore, one point receives the inner area of the circle and the other one the outer area. Figure 3 shows an example where $w_1 > w_2$.

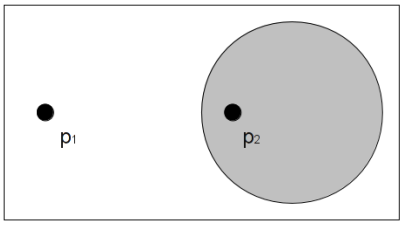

Figure 3. Dominance areas of P_1 and P_2 where $w_1 > w_2$

Without loss of generality, let $w_1 > w_2$ therefore p_1 dominates p_2 and p_1's dominance is the outer area while p_2's is the inner area of the circle. The center of the circle that represents the boundary of the dominance areas has center c_1 and the radius r_1 (Equation 6) calculated using the two points' coordinates and weights [Aurenhammer and Edelsbrunner 1984].

$$\vec{c_1} = \frac{w_2^2.\vec{p_1} - w_1^2.\vec{p_2}}{w_1^2 - w_2^2} \quad and \quad \vec{r} = \frac{w_1.w_2.|\vec{p_1} - \vec{p_2}|}{w_1^2 - w_2^2} \tag{6}$$

A Multiplicatively Weighted Voronoi Diagram represents each generator's dominance area. Therefore to compute a single point's dominance every Apollonius circle over

each other generator is computed. Figure 4 shows the dominance areas of P_1 over P_2 and P_3. In this example, P_1 dominates P_2 and is dominated by P_3. Thus the dominance area of P_1 is the intersection of the two areas, shown in gray.

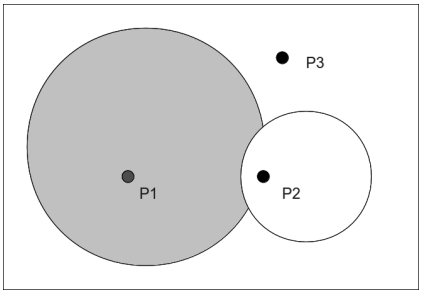

Figure 4. Intersection of the two dominance areas of the point P_1

The implemented code computes each point's dominance by intersecting every Apollonius circle created by combining the point with every other point in the set. Algorithm 1 shows the main steps to find the dominance area of a point $p[i]$ given the array p that stores every point in the set. It uses a straight forward bisector computation storing each generator point's dominance area.

Algorithm 1: Compute dominance area of the point pointList[i]

```
Input: i, pointList, RegionOfInterest
Output: dominanceArea (Area of dominance of the point i)
dominanceArea=RegionOfInterest;
for j de 0 a n do
    if i!=j then
        circleIJ=ApolloniusCircle(i,j,pointList);
        dominanceArea=Intersection(circleIJ,dominanceArea);
```

The function *ApolloniusCircle(i,j,pointList)* returns a regular polygon with 360 sides centered on the circle's center and with radius calculated using Equation 6. Thus, the circle is approximated using a polygon before computing the intersections.

The function *Intersection(circleIJ,dominanceArea)* computes polygon intersections, returning a new polygon representing the intersection area. This function's computational time is $O(a \cdot b)$ where a and b, represents how many vertexes are used in each polygon. This computation is performed by TerraLib's polygon overlay functions.

4.3. Complexity

The main iteration, represented by Algorithm 1 is $O(n^2)$ where n is the number of generating points. Each dominance intersection is computed by intersecting two non-convex polygons. The polygon *circleIJ* has 360 sides, so $a = 360$. The number of edges in a region is bounded by n [Aurenhammer and Edelsbrunner 1984] so the complexity of the method *Intersection(circleIJ,dominanceArea)* is bounded by $O(n \cdot 360)$. This method is called $O(n^2)$ times so the whole computational complexity of the algorithm is bounded by $360 \cdot O(n^3)$.

For better performance the data structure chosen to represent the dominance areas should be based on circular arcs instead of MultiPolygons. The SQL multimedia standard (ISO/IEC 13249 SQL/MM) aims to standardize the CurvePolygon and MultiSurface geometry types, which represents, respectively, a closed area defined by pieces of circular arcs and a collection of CurvePolygon [Stolze 2003].

Using this kind of data structure the method *Intersection(circleIJ,dominanceArea)* drops it's complexity to $O(n)$ as it's the computational time of intersecting n pieces of circles with a circle. Therefore the algorithm is bounded by $O(n^3)$. This result is better than most approximations that achieved $O(n^4)$ [Lee and Gahegan 2002] or worse [Dong 2008] but worse than the optimal algorithm [Aurenhammer and Edelsbrunner 1984].

Both by using polygon approximation or CurvePolygon datatype the algorithm scales well in memory. This is a straight result of the iteration process that computes only one dominance area at each iteration. The implementation stores each computed area in the database, so at each step only one dominance area and one circle are in memory. Implementations that compute the intersections first then reconstruct the dominance areas need to store all the intersections in memory during processing time [Gahegan and Lee 2000].

5. Results

Figure 5 shows the result of using the implemented plugin as a tool for geographic spatial partitioning. The polygon that limits the map is São Paulo city's boundary. The points represent the health care institutes in which the number of emergency treatments done in the year of 2009 were measured. The circular arcs are the Multiplicatively Weighted Voronoi area boundaries.

Figure 5. Weighted Voronoi Diagram of São Paulo's Health Centers

In this application the diagram was used as an weighted area assignment tool.

Thus health centers that treated more people than it's neighbors were assigned bigger areas. Some areas with much lesser number of treatments received areas so small that they can be seen as with insignificant influence. This happened because there are health centers with 2 treatments per year near other centers with more than 200.000 treatments per year. This is a major improvement from the Ordinary Voronoi in this application since the later would assign areas without any regard to the relevance of each point in the space.

In the southern area of the map there is a district called Grajaú that is populated mostly by familys with low income. The health center there was assigned the biggest area in the map since it is responsible for treating more people than any other health care in the city and there is no other health center of the same size near. This is a good example of how a weighted Voronoi area can grow large if the generator point is in fact dominating it's neighbors.

Since the implementation uses an approximation of the circles as polygons instead of the proposed CurvePolygon data type a topology error was created on every intersection. This happens because every point that describes the end of a circular arc in the diagram is the result of the intersection of three circles as shown in Figure 4 [Lee and Gahegan 2002]. When the circles are approximated by polygons, the intersection of the areas is not computed using circular arcs but using lines. Figure 6 shows the spaces that are produced between the areas where there should be an intersection point.

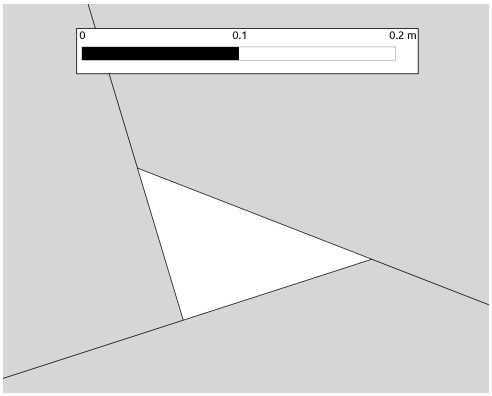

Figure 6. Approximation error in the intersections

The percent error (e)on the approximation can be found using Equation 7 as shown in Figure 7. Applying this equation to the 360 sides used by default in the implementation, the estimated percent error in the boundary of each area is 0.0001523.

$$e = \frac{R \cdot (1 - cos\theta)}{R} \rightarrow e = 1 - cos(\frac{2 \cdot \pi}{360}) \approx 0.0001523 \quad (7)$$

6. Conclusion

As the algorithm's main effort is to find each point's dominance area in each step, the void areas without dominance found in some previous researches [Aurenhammer and Edelsbrunner 1984] [Rezende et al. 2000] does not happen. From the

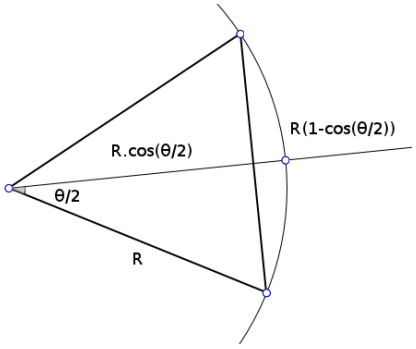

Figure 7. Circle as 360 side Polygon approximation

mathematical definition in Equation 2 there should be no area where the inequation is unsolvable. This algorithm builds by brute force the dominance areas, preserving multi part polygons, therefore areas can be discontinuous, but not empty. The application of the diagram in a geographic environment is enhanced by this improvement as empty spaces in the area partitioning are an undesired effect.

The representation of the circles as polygons before intersection brought an unnecessary speed loss. Using geometries based on circle sections would reduce the computational time to find the intersections and give better topology nodes.

The implemented code was compiled both on Windows and Linux environment which allows the tool to be used by many researchers. It's released under GPL license as a TerraView Plugin and is available through the TerraLib repository. This implementation is a proof of concept that is already usefull for many purposes. The next goal in the research is to implement the MultiCurve geometry type as defined by SQL/MM in order to improve both precision and speed.

Acknowledgments

This work is the result of the final article written for Introduction to Geoprocessing course in order to fulfill the requirements for the master's degree on Remote Sensing assigned by the Brazilian Army on National Institute for Space Research (INPE). The database of health care emergency treatments in São Paulo City were kindly provided by CEInfo for testing purposes.

References

Aurenhammer, F. (1991). Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405.

Aurenhammer, F. and Edelsbrunner, H. (1984). An optimal algorithm for constructing the weighted Voronoi diagram in the plane. *Pattern Recognition*, 17(2):251–257.

Aurenhammer, F. and Klein, R. (2000). Handbook of Computational Geometry, chapter Voronoi Diagrams. *Elsevier*, 169:170.

Boots, B. (1986). *Voronoi (Thiessen) Polygons*, volume 45. Geo Books.

Câmara, G., Vinhas, L., Ferreira, K., Queiroz, G., Souza, R., Monteiro, A., Carvalho, M., Casanova, M., and Freitas, U. (2008). TerraLib: An open source GIS library for large-scale environmental and socio-economic applications. *Open Source Approaches in Spatial Data Handling*, pages 247–270.

Dong, P. (2008). Generating and updating multiplicatively weighted Voronoi diagrams for point, line and polygon features in GIS. *Computers & Geosciences*, 34(4):411–421.

Gahegan, M. and Lee, I. (2000). Data structures and algorithms to support interactive spatial analysis using dynamic Voronoi diagrams. *Computers, environment and urban systems*, 24(6):509–537.

Jasmin, B. and Summerfield, M. (2005). C++ gui programming with qt 3.

Karavelas, M. I. and Yvinec, M. (2002). Dynamic additively weighted voronoi diagrams in 2d. In *Proceedings of the 10th Annual European Symposium on Algorithms*, ESA '02, pages 586–598, London, UK, UK. Springer-Verlag.

Karimi, F., Delavar, M. R., and Mostafavi, M. A. (2009). Space allocation of educational centers using multiplicatively weighted voronoi diagram. *WG II/2, II/3, II/4: Workshop on Quality, Scale & Analysis Aspects of City Models*.

Lee, I. and Gahegan, M. (2002). Interactive analysis using Voronoi diagrams: Algorithms to support dynamic update from a generic triangle-based data structure. *Transactions in GIS*, 6(2):89–114.

Novaes, A., Souza de Cursi, J., da Silva, A., and Souza, J. (2009). Solving continuous location-districting problems with voronoi diagrams. *Computers & Operations Research*, 36(1):40–59.

Ohyama, T. (2011). Takashi ohyama's multiplicatively weighted voronoi diagram page. `http://www.nirarebakun.com/voro/emwvoro.html`.

Rezende, F., Almeida, R., and Nobre, F. (2000). Diagramas de Voronoi para a definição de áreas de abrangência de hospitais públicos no Município do Rio de Janeiro. *Cadernos de Saúde Pública*, 16(2):467–475.

Stolze, K. (2003). Sql/mm spatial: The standard to manage spatial data in relational database systems. In *Proceedings of the BTW*.