

## Representation of Water Resources in Satellite Images using Voronoi Diagrams

Ahmed M. Zeki

*Dept. of Computer Sci., Fac. of Information and Communication Tech.  
Inter. Islamic Univ., Kuala Lumpur, Malaysia*

### Abstract

Voronoi Diagrams (VD) is an important and useful concept in computational geometry. It weights great significance not only in computational geometry but in other fields of science. VD is used in several applications such as segmentation, thinning, and others. It represents the image components using polygons which can be seen also as a set of vectors.

The representation of images using VD provides a number of advantages. Among them are, unlike the raster images, we will be able to deal with objects rather than pixels. Measuring some quantities (e.g. areas and perimeters) is much easier and more accurate by the coordinates of polygons rather than by counting pixels. Among the drawbacks of VD-based methods is that its construction is a time consuming process, however, the gain in later processes and analysis recovers this limitation.

This paper describes the development of a system used in representing water resources in satellite images using VD. The image is read first, VD is then constructed and used to segment the image components. Irregular shapes (e.g. lakes) are separated using Voronoi polygons. VD is also used to extract the midline of rivers.

**Keywords:** Satellite Images, Voronoi Diagrams, Water Resources, Segmentation, Thinning

### Introduction

Voronoi Diagrams (VD) is a rediscovered mathematical concept from the 19<sup>th</sup> century. It is one of the most fundamental and useful constructs defined by irregular lattices. Thus, it is widely used in geometrical construction and many other fields. Those fields include –but not limited to– crystallography, physics, astronomy, astrophysics, electronics, radiation, chemistry, image processing, medicine, microbiology, networks, imagery, geography, telecommunication and many others [1].

VD is defined as lines that bisect the lines between a centre point and its surrounding points. The bisecting lines and connections lines are perpendicular to each other. When this rule is used, the area will be completely covered by adjacent polygons. It is also defined as a collection of geometric objects that partition the plane into cells, each of which consists of the points closer to one particular object than to any others [2].

Let  $P = \{p_1, \dots, p_n\}$  be a set of points (sites or generators) in the 2D plane. Let  $d(p, q)$  be the Euclidean distance between points  $p$  and  $q$ . The Voronoi region of a point  $p_i$  is the region given by:

$$V(p_i) = \{p \mid d(p, p_i) \leq d(p, p_j), \forall j \neq i\} \quad (1)$$

The point-VD (also called ordinary VD) generated from the set of points  $P$  is given as the set of Voronoi regions:

$$V(P) = \{V(p_1), \dots, V(p_n)\} \quad (2)$$

The boundaries of Voronoi regions may consist of line segments, half lines or infinite lines, which are called Voronoi edges. A Voronoi edge is the perpendicular bisector of the segment joining the two sites. These two points are called Voronoi neighbours. A Voronoi point (or vertex) indicates a point shared by exactly three Voronoi edges. If three sites determine Voronoi regions that meet at a Voronoi point, the circle through those three sites is centred at that Voronoi point, and there are no other sites in the circle (empty circle property) [3]. All Voronoi regions are convex polygons, some are finite and some are infinite [4,5]. Figure 1 shows an example of a VD being constructed for a set of points.

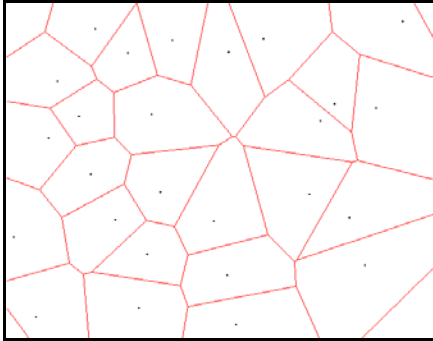


Figure 1: VD for a set of points

### Construction of Voronoi Diagrams

A variety of algorithms have been proposed to construct VD. Lafon and Blanc [4] classified the construction methods under two main categories; static and dynamic algorithms. In the static category, all generators must be known in advance. The divide-and-conquer algorithm of Shamos and Hoey [6] is a typical method of this type, in which, the set of generators is recursively divided into smaller subsets and the VDs for those subsets are merged into the final diagram. While in the dynamic algorithms generators are not known in advance. The incremental method, for instance, starts with a simple VD for a few generators and modifies the diagram by adding other generators one by one [7]. The incremental method is probably the most popular one due to its relative ease of implementation. It is one of the most practical ones from both the time complexity and the robustness points of view [3]. Another dynamic algorithm is the plane-sweep algorithm [8] in which, a vertical line is moved over the plane from (say) left to right and the VD is constructed along the line.

Optimal algorithms for Voronoi tessellation have a worst-case time complexity of  $O(n \log n)$  and  $O(n)$  on the average, where  $n$  is the number of generators [9].

### **Applications**

As mentioned earlier, VD has many applications. Drysdale [5] listed about twenty fields in which Voronoi Diagrams are commonly used. This section lists, as examples, a few applications from image processing point of view.

### **Visualization and Shape Modelling and Reconstruction**

The automatic shape reconstruction of 3D structures is one of the major goals of computer vision. Amenta et al. [10] reconstructed surfaces using 3D VD and DT. The method produces a set of triangles that are called the crust of the sample points. A simple combinational algorithm is given that computes a piece-wise linear approximation of a smooth surface from a finite set of sample points. The algorithm uses Voronoi vertices to remove triangles from the DT [11]. Automatic mesh generation is considered the main bottle-neck of the Finite Element Analysis nowadays [12]. Meshing is used for purpose of visualization, modelling, animation, and others. In addition to the applications of meshing, VD is used to create an interesting artistic effect called mosaicing [13]. A mosaic is a tiled image, where each tile has a single colour. The VD of a point set can be used as a tiling. Each Voronoi tile is coloured by the image pixel closest to the point site.

### **Motion Planning**

Motion planning is a fundamental problem in robotics and computational geometry. Hoff et al. [13] computed generalized 2D and 3D VD using interpolation-based polygon rasterization hardware. A discrete VD is computed by rendering a 3D distance mesh for each Voronoi site. They applied their method on motion planning in static and dynamic environments. The Voronoi boundaries of obstacles provide paths of maximal clearance between the obstacles. The Voronoi algorithm computes the approximate distance to the nearest obstacle. It can also be used to find the minimum-length paths between points in the Euclidean plane with polygonal obstacles [14].

Gold [15] constructed dynamic VD to provide a natural structure for dealing with adjacency-based collision-avoidance problems in marine navigation. Embedding the moving boat within the diagram permits the detection of immediate neighbors within any specified tolerance. The dynamic structure allows real-time local updating of obstacle and marker positions. In another research paper, Gold et al. [16, 17] used VD as a basic tool for Geographic Information System (GIS). They showed how VD can be used to maintain the topology of the map objects of a GIS.

### **Skeletonization**

The essential information about a shape is stored in its skeleton. Skeletonization (medial axis extraction or thinning) is the process of extracting the skeleton of a pattern. The skeleton is defined as the set of points which have semi-equal distance from two or more points of the pattern contour [18]. The relationship between VD and the extraction of the medial axis is quite clear from the definition. Hence, it is not surprising that the use of VD for skeletonization is an old problem [19]. The applications of the medial axis are tremendous, including the separation of abutting or overlapping objects, extraction of line graphs, and interpretation of road maps, just to name few [20].

### Segmentation

Segmentation and identification of various components of a document is a necessary pre-processing stage for document analysis and understanding. VD was used to extract the text-line from binary document images [21]. To represent the physical structure of a page, the method uses the neighbour graph which represents neighbours of connected components. The use of the area Voronoi Diagram enables extracting neighbours without predetermined parameters. Based on the neighbour graph, the task of text-line extraction is considered to be the selection of its sub-graphs appropriate as text-lines. The method can be applied to titled non-rectangular pages.

Kalitzin et al. [22] tessellated blurred CT-scan images using VD generated by a set of points extracted from the image. The set of centres of the Voronoi cells is the set of local extrema of the grey-scale image. This set is localized by using the winding number distribution of the gradient vector field. Finally, the segmentation is done based on the scale evolution which induces hierarchical structure of embedded segments. Papadopoulou and Lee [23] applied VD to segment VLSI components in order to predict the yield of a VLSI chip.

### System Description

#### Image Reading

The System is developed using MATLAB® v.6.5.1 and tested on Dell laptop powered by Pentium 4 (2.8 GHz) and 1 GHz of RAM. The images were downloaded from the Internet from various sources. In order to process the image, any coloured or grey image is converted into a binary image. To do so, the coloured image is converted first into a grey image. The grey one is then converted into a binary image using a threshold value  $T$  which is in the range  $[0,1]$ . The binary image has values of 1 (black) for all pixels in the input image with luminance less than  $T$ , and 0 (white) for all other pixels. It was found that  $T = 2.5$  gives the best results in our case. As a result, the water resource will be shown in black colour, while other components are shown in white colour.

Figure 2 shows a coloured satellite image. Figure 3 shows the same image converted into a grey one, while Figure 4 shows it after conversion into binary image.



Figure 2: A Coloured Satellite Image



Figure 3: A Greyscale version of Figure 2



Figure 4: A binary version of Figure 3 using  $T = 2.5$

#### Noise Removal and Dilation

As shown in Figure 4, there is noise on the land. To reduce the noise, those single pixels are removed. The result is shown in Figure 5. This process is followed by a dilation operation which further enhances the image as shown in Figure 6.



Figure 5: Removing the single pixels



Figure 6: Applying Dilation

#### Edge Detection and Contour Tracing

Edges of image components will be detected in this stage. Edge detection and contour tracing is needed almost in every image processing system. Edge detection requires neighbourhood operators that are sensitive to changes and suppress areas of constant grey values [24]. The task becomes much easier when dealing with binary images. The 8-neighbourhood representation is used in this work. Contour tracing is done clockwise starting at the upper-left most pixel. Object edge points miss at least one of their neighbours. Thus, to remove the interior points and keep only those at the

boundary, the image is scanned using a  $3 \times 3$  window. The centre pixel is set to 0 if all its 8-neighbours are 1. The detected edges are shown in Figure 7.

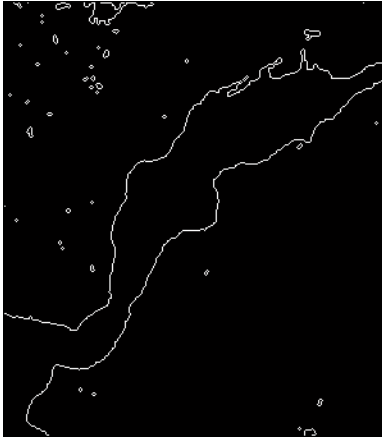


Figure 7: Edge Detection of All Components

#### Isolation of the Image Components using Area-Voronoi Diagrams

Area-VD will draw line segments between the connected components. The collection of those lines will be able to separate the connected components. The following algorithm is applied to construct area-VD:

1. Every pixel in the image is labelled with a number referring to the connected component it belongs to. Pixels of the background are labelled with 0.
2. The inner and outer contours are traced.
3. Point-VD is constructed from all points of the contour.
4. The coordinates of Voronoi edges are determined.
5. The labels of Voronoi edges (from step 1) are examined.
6. If the two generators of adjacent polygons have different labels, the edge is maintained and plotted, otherwise it is deleted.

For the details of the construction method, see [25]. Figure 8 shows area VD been constructed using the above algorithm. The construction took 137.51 seconds to be processed

#### Sampling

In fact, the sampling process is needed in the next stage. The samples are selected along the contour only (including the inner contour). To determine the starting point for sampling, the image is scanned column-wise from left to right. The first ON pixel encountered is considered as the first sample. The contour is traced clockwise and the other samples are simply selected such that every  $R^{\text{th}}$  pixel is preserved until it terminates for each connected component, where  $R$  is a sampling parameter,  $R = 1, 2, 3, \dots, n$ . When  $R = 1$ , all pixels of the contour are selected. If  $n$  is greater than the total number of pixels of the contour, only one sample will be taken.

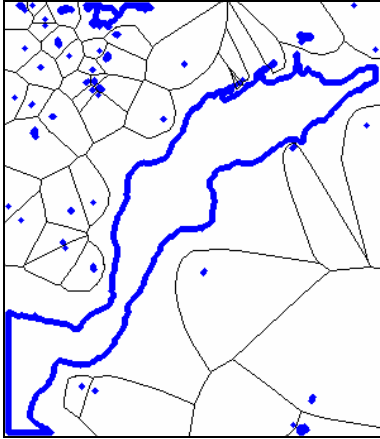


Figure 8. Area-VD for the connected components of the image.

The sampling process helps in reducing the time and memory requirement in constructing VD which highly depends on the number of Voronoi generators. The higher the number of generators, the slower the construction and the more memory it consumes. The interesting thing is that the area-VD resulted from the samples does not differ significantly from the one generated from all points of the contour. In fact, both diagrams will partition the plane into mutually exclusive regions. Figure 9 shows samples been selected from the contour using  $R = 7$ . The image in Figure 9 is processed in 4.507 seconds which means that the time is reduced to less than 1% which is a great gain as bigger images will require more time.



Figure 9. Samples from the Contours using  $R = 7$ .

#### **Extraction of the River Midline**

The chosen samples are used to construct VD. The resulted VD will produce Voronoi edges. Only those edges fall completely inside a connected component were selected and drawn. In fact, those edges represent the midline of that particular object.

Any object with no such edges will be ignored and hence deleted. Figure 10 shows the midline of a river.

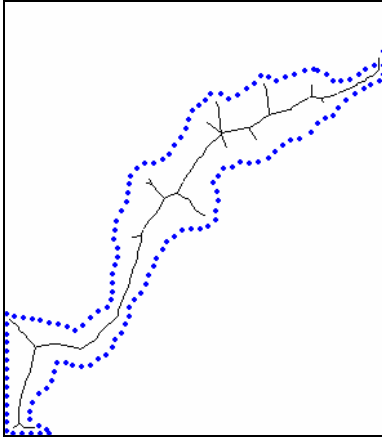


Figure 10. The Midline of the river

### Conclusions

In this paper, a preliminary research was done to explore the possibility of using Voronoi Diagrams in processing satellite images containing water resources. The study showed promising results.

The coloured satellite image is first read and converted into a binary image. The image is then cleaned by removing the noise and applying dilation process. The edges of the remaining objects were detected and traced to select samples. The samples are used to construct point-VD and Area-VD. It was found that it is better to construct VD using those samples rather than choosing all points of the contour. This helps in speeding up the construction without affecting the resulted diagram.

Area-VD is used to separate the objects of the image. It showed a high accuracy. Finally, the midline of the water resource is constructed.

Among the advantages of the VD representation of the water resources is that we will be able to deal with objects rather than pixels. Measuring some quantities (e.g. areas and perimeters) becomes much easier and more accurate by the coordinates of polygons rather than by counting the image's pixels.

However, few issues still need to be enhanced and further studied. For example, the midline for rounded objects will not be extracted properly. The midline for long objects (e.g rivers) need to be enhanced also as it contains some unnecessary branches. The use of dilation process has a serious drawback which reduces the size of the processed object. There is also a need for a proper sampling process which considers the concavity areas.



### References

- [1] **Zeki, A.M., Ghyasi, A.F., Mujahid, M. Zainul, N. Cheddad, A., Zubayr, M. and Zakaria, M.S. 2004.** Design and Implementation of a Voronoi Diagrams Generator using Java. International Arab Conference on Information Technology (ACIT'2004). Mentouri University of Constantine, Algeria. **1**:194-198, 12-15 Dec.
- [2] **Eppstein, D. 1990.** The Furthest Point Delaunay Triangulation Minimizes Angles. Computational Geometry Theory and Applications. Vol. 1. 143-148.
- [3] **Okabe, A., Boots, B. and Sugihara K. 1992.** Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. Prentice Hall.
- [4] **Lafon, J.C. & Blanc, G. 1997.** Meshing and visualization of oilfields. IEEE Conference on Information Visualization (IV'97). pp. 18-21.
- [5] **Drysdale, S. Voronoi. 1993.** Diagrams Applications from Archeology to Zoology. Regional Geometry Institute. Available at:  
<http://www.ics.uci.edu/~eppstein/gina/scot.drysdale.html>
- [6] **Shamos, M.I. & Hoey, D. 1975.** Closest-point problems. 16<sup>th</sup> IEEE Symposium on Foundations of Computer Science. pp. 151-162.
- [7] **Green, P.J. & Sibson, R. 1978.** Computing Dirichlet tessellation in the plane. The Computer Journal. **21**:168-173.
- [8] **Fortune, S. 1987.** A sweepline algorithm for Voronoi diagrams. Algorithmica **2**(2):153-174.
- [9] **Preparata, F.P. & Shamos, M.I. 1988.** Computational geometry an introduction. Springer-Verlag.
- [10] **Amenta, N., Bern, M. & Kamvysselis, M. 1998.** A New Voronoi-Based Surface Reconstruction Algorithm. SigGraph'98.
- [11] Amenta, N. & Bern, M. 1999. Surface reconstruction by Voronoi filtering. Discrete and Computational Geometry. **22**:481-504.
- [12] **Gyimothy, S., Vagvolgyi, A. & Sebestyen, I. 2000.** Boundary refinement in n-dimensional Delaunay-meshing. IEEE Transactions on Magnetics. **36**(4):1569–1573. Jul.
- [13] **Hoff, K.E., Keyser, J. Lin, M., Manocha, D. & Culver, T. 1999.** Fast computation of generalized Voronoi diagrams using graphics hardware. 26<sup>th</sup> International Conference on Computer Graphics and Interactive Techniques. pp. 277-286.

- [14] **Storer, J.A. & Reif, J.H. 1994.** Shortest paths in the plane with polygonal obstacles. *Journal of the Association for Computing Machinery (ACM)*. **41(5)**:982-1012. Sep.
- [15] **Gold, C.M. 1998.** The use of the dynamic Voronoi data structure in autonomous marine navigation. 29<sup>th</sup> International Symposium on Robotics (ISR98). Birmingham, England. pp. 217-220.
- [16] **Gold, C.M., Remmele, P.R. and Roos, T. 1997.** Voronoi methods in GIS. In *Algorithmic Foundations of GIS, Lecture Notes in Computer Science*. No. 1340, (Ed.: Van Kreveld, M., Nievergeld, J., Roos, T. and Widmeyer, P.). Springer-Verlag, Berlin, Germany. pp. 21-35.
- [17] **Gold, C.M., Remmele, P.R. & Roos, T. 1999.** Fully dynamic and kinematic Voronoi diagrams in GIS. *Algorithmica*. Special Issue on Cartography and GIS.
- [18] **Mahmoud, S., Abuhaiba, I. & Green, R. 1991.** Skeletonization of Arabic characters using clustering based skeletonization algorithm (CBSA). *Pattern Recognition*. **24(5)**:453-464.
- [19] **Montanari, U. 1969.** Continuous skeletons from digitized images. *Journal of the Association for Computing Machinery (ACM)*. **16(4)**:534-549.
- [20] **Ogniewicz, R. & Ilg, M. 1992.** Voronoi skeletons: theory and applications. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'92)*. pp. 63-69. 15-18 Jun.
- [21] **Kise, K., Iwata, M., Matsumoto, K. & Dengel. 1998.** A computational geometric approach to text-line extraction from binary document images. *Document Analysis Systems (DAS'98)*. pp. 346-355.
- [22] **Kalitzin, S.N., Romeny, B.H. & Viergever, M. 1997.** On topological deep-structure segmentation. *International Conference on Image Processing (ICIP'97)*. pp. 863-866.
- [23] **Papadopoulou, E. & Lee, D.T. 2001.** The Linfty-Voronoi diagram of segments and VLSI applications. *International Journal of Computational Geometry and Applications*. **11(5)**:503-528.
- [24] **Jähne, B. 2002.** *Digital Image Processing*. 5<sup>th</sup> edition. Springer.
- [25] **Zeki, A.M., Zakaria, M.S. & Liong, C.-Y. 2007.** The use of area-Voronoi diagram in separating Arabic text connected components. Submitted to the International Symposium on Signal Processing and its Applications, Sharjah, UAE, Feb 2007.