Voronoi Diagrams and Applications

M. Melkemi and D. Vandorpe
LIGIA-LISPI
Université Claude Bernard Lyon.
43Bd 11 Novembre 1918, Bat. 710
69100 Villeurbanne.
e-mail melkemi@ligia.univ-lyon1.fr

Abstract

This paper presents two applications of digital and continuous Voronoi diagrams in computer vision. In fact, we compute the shape of discrete points and we present the relationship between Voronoi diagrams and skeletons. Our propositions are organized in three parts: 1) we propose an efficient algorithm which computes the digital Voronoi tessellation and is applied to compute the shape of a set of discrete points. 2) We compute the error of an approximate skeleton of a polygonal shape. This approximation is obtained from the Voronoi diagram of sampled points. 3) We use the Voronoi diagram of line segments for computing an exact and almost complete skeleton of a polygonal shape.

1 Introduction

The Voronoi diagrams are powerful tools in computer vision. Indeed, several applications and efficient solutions using the ordinary Voronoi diagram and its generalized versions are widely discussed in the literature [1] [7] [10] [11] [12] [17]. In fact, the Voronoi diagram is used for image compression, image segmentation, texture segmentation, skeletonization, shape computation, shape decomposition, three dimensional reconstruction problems and so on. The present paper is organized in two levels, the first deals with the digital Voronoi diagram and its application to the shape computation. The second one presents a study of the relationship between the continuous Voronoi diagram and the skeletonization problem.

Computing the shape of points using digital Voronoi diagram

Several applications, as segmentation and compression problems, passe through the computation of the continuous Voronoi diagram using approaches issued from computational geometry domain. The obtained Voronoi diagram is digitized and the Voronoi regions are identified using the extraction of connected components algorithms. These steps, computation and digitization, use the most time of the algorithm. This problems can be circumvented by the development of digital geometry algorithms. In this perspective, we propose an algorithm computing the shape of digital set of points. First, we compute the true digital Voronoi tessellation of n discrete points in $O(m^2)$ where m^2 is the image size. The algorithm stores only one $m \times m$ image and the coordinates of the n initial points. The advantage is that the complexity of the algorithm is independent from the number of points n. Using this algorithm, we can then compute the Delaunay triangulation. Finally, the shape is identified as a subgraph of the computed triangulation in $O(n)$ complexity.

Voronoi diagrams and Skeletons

The concept of skeletonization denotes a process which transforms a 2D shape in 1D representation. This is a key problem in several domains as pattern recognition and robotic domains. The relationship between skeletons and Voronoi diagrams is discussed in [3]. In fact, it has been observed that the Euclidean skeleton is embedded in a generalized Voronoi diagram of a polygonal shape [9] and in the ordinary Voronoi diagram of points issued from regular sampling of polygonal shape [3]. In this part, we give, first, an error estimation of an approximate skeleton of a polygonal shape. This approximation is computed from the Voronoi diagram of points. These points are obtained from regular sampling of the boundary shape. In the second part, we present an approach which computes a skeleton of a polygonal shape from the Voronoi diagram of line segments. The computation of continuous skeletons from the generalized Voronoi diagram of polygons is not completely solved. In other words, we have not found an efficient algorithm which computes a generalized Voronoi diagram of any set of polygons, having arbitrary shapes, with less complexity. Nevertheless, we have found some algorithms which compute this generalized Voronoi in some particular conditions [8] [12] [15] [18]. We propose the use of the line segments Voronoi diagram for computing a continuous skeleton of a polygonal shape. The idea is to decompose each polygon in non connected elements (points and line segments) and to compute the Voronoi diagram of line segments. The skeletons of polygonal shapes are obtained from the computed Voronoi diagram by deletion of some useless bisectors.

In section 2, the algorithm which computes the digital Voronoi diagram and the shape of discrete points are given. In section 3, we give the error analysis of an approximate skeleton of a polygonal shape and we present the algorithm which computes a continuous skeleton of a polygonal shape from the Voronoi diagram of line segments.

2 Computing the shape of discrete points

2.1 Definitions

Let $S = \{P_1, P_2, \cdots, P_n\}$ be a set of points (called *seeds*) in digital image I of size $m \times m$ where $P_i = (x_i, y_i)$ and x_i, y_i are the coordinates in I.
A matrix V of size $m \times m$ is defined as follows. For any point P with coordinates (x, y), $V(x,y) = i$ if $d_e(P_i, P) < d_e(P_j, P)$ for all $j \neq i$ and $d_e(P_i, P) < d_e(P_j, P)$ for all $j < i$. d_e denotes the Euclidean distance.

The *digital Voronoi region* R_i of a seed P_i is a set of pixels (x, y) in V such that $V(x,y) = i$.

Two Voronoi regions R_i and R_j are neighbors if and only if exist two 4-neighboring pixels (i_1, j_1) and (i_2, j_2) such that $V(i_1, j_1) = i$ and $V(i_2, j_2) = j$.

The *digital Voronoi tessellation* of S is the set of all digital Voronoi region $R_i (i = 1, \cdots, n)$.

The digital Voronoi diagram consists of pixels which lie on the boundary of the Voronoi regions such that it is 8-connected and it has unit thickness.

The *digital Voronoi diagram* of S consists of pixels (i,j) such that there is a 4-neighbor (i_1, j_1) of (i,j) and $V(i,j) < V(i_1, j_1)$.

The Delaunay triangulation is the geometrical dual of the Voronoi diagram, obtained by linking seeds whose Voronoi regions are adjacent across a common face.

If no four points of S are cocircular then by linking the neighboring seeds we obtain the correct Delaunay triangulation.

The Delaunay triangles P_i, P_j, P_k are determined by the following manner: for every pixel $M = (x,y)$ in I, we assume $V(x,y) = i$. If there are two 4-neighbors $M' = (x', y')$ and $M" = (x", y")$ such that M' and $M"$ are 8-neighbors of each other and $V(x', y') = j$ and $V(x", y") = k$ then P_i, P_j, P_k is a Delaunay triangle.

2.2 Computing digital Voronoi tessellation

The algorithm is based on the following idea: the image which contains the digital Voronoi tessellation can be obtained by propagating locally (8-neighborhood) the labels of the nearest seeds. The distance between a seed and a given pixel is not propagated approximately, as in [4] [5], but rather computed correctly. The algorithm stores, only, one image V and the coordinates of seeds $P_i (i = 1, \cdots, n)$. In the beginning, V consists of seed pixels P_i with the initial value i (the label of the seed) and the other pixels with initial value zero. The algorithm is sequential and described as follow:

First passes

```
For j = 2,···,m − 1 do
Begin
  For i = 2,···,m − 1 do
  Begin
    k0 = V(i,j), k1 = V(i − 1,j − 1)
    k2 = V(i,j − 1), k3 = V(i + 1,j − 1)
    k4 = V(i − 1,j)
    V(i,j) = k such that:
    k is the smallest label defined as follows:
    de((i,j),Pk) = minimum (de((i,j),Pkr))
                       r = 0,···,4 and kr ≠ 0
  End
  For i = m − 1,···,2 do
  Begin
    k5 = V(i + 1,j), k0 = V(i,j)
    if ( k5 > 0) then
    if (de((i,j),Pk5) < de((i,j),Pk0))
      then V(i,j) = k5
    else if (de((i,j),Pk5) = de((i,j),Pk0) )
        then V(i,j) = minimum (k0,k5)
  End
End
```

Second passes

For $j = m-1, \cdots, 2$ do
Begin
  For $i = 2, \cdots, m-1$ do
  Begin
    $k_0 = V(i,j), k_1 = V(i-1, j+1)$
    $k_2 = V(i, j+1), k_3 = V(i+1, j+1)$
    $k_4 = V(i-1, j)$
    $V(i,j) = k$ such that:
    k is the smallest label defined as follows:
    $d_e((i,j), P_k) =$ minimum $(d_e((i,j), P_{k_r}))$
      $r = 0, \cdots, 4$ and $k_r \neq 0$
  End
  For $i = m-1, \cdots, 2$ do
  Begin
    $k_5 = V(i+1, j), k_0 = V(i,j)$
    if ($k_5 > 0$) then
    if $(d_e((i,j), P_{k_5}) < d_e((i,j), P_{k_0}))$
      then $V(i,j) = k_5$
    else if $(d_e((i,j), P_{k_5}) = d_e((i,j), P_{k_0}))$
      then $V(i,j) =$ minimum (k_0, k_5)
  End
End

Figures 1,2 show an example of a digital Voronoi tessellation computed using the previous algorithm.

0 0 0 1 1 1 1 1 2 2 2 2 2 2 2 2 4 4 4
0 0 0 **1** 1 1 1 1 2 2 2 2 2 2 2 4 4 4 4
0 0 0 1 1 1 1 1 2 2 2 2 2 2 4 4 4 4 4
0 0 0 1 1 1 1 1 2 2 2 2 2 4 4 4 4 4 4
0 0 0 1 1 1 1 1 **2** 2 2 2 2 4 4 4 4 4 4
0 0 0 1 3 3 3 3 2 2 2 2 2 4 4 4 4 4 4
0 0 0 1 3 3 3 3 2 2 2 2 2 4 4 4 4 4 4
0 0 0 1 3 3 3 3 2 2 2 2 2 4 4 4 4 4 4
0 0 0 1 **3** 3 3 3 2 5 5 5 5 **4** 4 4 4 4 4
0 0 0 1 3 3 3 3 3 5 5 5 5 4 4 4 4 4 4
0 0 0 1 3 3 3 3 3 5 5 5 5 4 4 4 4 4 4
0 0 0 1 3 3 3 6 6 **5** 5 5 5 4 4 4 4 4 4
0 0 0 1 3 3 6 6 6 5 5 5 5 4 4 4 4 4 4
0 0 0 1 3 3 6 6 6 5 5 5 5 5 4 4 4 4 4
0 0 0 1 3 3 6 6 6 5 5 5 5 5 5 4 4 4 4
0 0 0 1 3 3 **6** 6 6 6 6 5 5 5 5 5 4 4 4
0 0 0 1 3 3 6 6 6 6 6 6 5 5 5 5 4 4 4
0 0 0 1 3 3 6 6 6 6 6 6 6 5 5 5 5 4 4
0 0 0 1 3 3 6 6 6 6 6 6 6 6 6 5 5 5 4

Figure 1. The matrix V after the first passes. S is a set of six pixels (indicated by underlined labels). Labels 'i' mark the i^{th} digital Voronoi region $i = 1, \cdots, 6$

1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 4 4 4
1 1 1 **1** 1 1 1 2 2 2 2 2 2 2 2 4 4 4 4
1 1 1 1 1 1 2 2 2 2 2 2 2 2 4 4 4 4 4
1 1 1 1 1 1 2 2 2 2 2 2 2 4 4 4 4 4 4
1 1 1 1 1 2 2 2 **2** 2 2 2 2 4 4 4 4 4 4
1 3 3 3 3 2 2 2 2 2 2 2 4 4 4 4 4 4 4
3 3 3 3 3 3 2 2 2 2 2 4 4 4 4 4 4 4 4
3 3 3 3 3 3 3 2 2 2 4 4 4 4 4 4 4 4 4
3 3 3 3 **3** 3 3 3 5 5 4 4 4 **4** 4 4 4 4 4
3 3 3 3 3 3 3 5 5 5 5 4 4 4 4 4 4 4 4
3 3 3 3 3 3 3 5 5 5 5 5 4 4 4 4 4 4 4
3 3 3 3 3 3 5 5 5 **5** 5 5 5 4 4 4 4 4 4
3 3 3 3 6 6 6 5 5 5 5 5 5 4 4 4 4 4 4
6 6 6 6 6 6 6 6 5 5 5 5 5 5 4 4 4 4 4
6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 4 4 4 4
6 6 6 6 6 6 **6** 6 6 6 6 5 5 5 5 5 4 4 4
6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 4 4 4
6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 4 4
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 4

Figure 2. The digital Voronoi tessellation (the matrix V after the second passes)

This algorithm computes the digital Voronoi diagram and the Delaunay triangulation with $O(m^2)$ complexity. The complexity of this algorithm does not depend on the number of the seeds but it depends only on the size of the image. Table 1 indicates the dependence between the size of the image and the CPU time (sec) for computing the digital Voronoi diagram.

Image size	64^2	128^2	256^2	512^2
CPU time (sec)	0,13	0,49	2,40	18,5

Table 1. Dependence between CPU time (sec) and image size ($m \times m$)

An example of the Delaunay triangulation, computed from digital and continuous Voronoi diagrams, are respectively illustrated in Figure 3(c) and Figure 3(d).

Parui et al.[14] have computed the shape of a dot pattern using the Delaunay triangulation, where the most time consuming step is to find the digital Voronoi diagram and the Delaunay triangulation. The complexity of their algorithm is $O(n \times m^2)$ where n is the number of seeds. It is clear that the complexity of the presented algorithm ($O(m^2)$) is less than the complexity of the algorithm described in [14].

2.3 Computing the α-shape

During the last decades, computing the shape of a set of points (dot pattern) has been a constant research topic. This is a key problem in several image analysis applications. Such applications are the classical problem of clustering. For instance, an air traffic situation may be represented by specifying the point locations of aircraft. This representation may then be used to detect potential collisions. Research on visual perception has made extensive use of dot patterns to investigate human image understanding. In biomedical research, dot patterns analysis is used to study the cellular sociology which aims to analyze relationship that link biological function of cells to their spatial location inside a given tissue. Others applications can be found in image analysis and pattern recognition fields.

Edelsbrunner et al.[6] introduce the notion of the α-hull which is the generalization of the concept of convex hull. We consider here the case where α is an arbitrary negative number. The α-hull of a set S is defined as the intersection of all closed complements of discs (where these discs have radii $-1/\alpha$) that contain all the points of S. The discrete family of the α-hulls are called α-shapes. Edelsbrunner et al. have shown that the α-shape ($\alpha < 0$) of S is subgraph of the Delaunay triangulation of S which can be computed from the closest Voronoi diagram. It has proved also that for every Delaunay edges e there exists two real numbers α_{min} and α_{max} ($\alpha_{min} < \alpha_{max}$) such that: e is an edge of the α-shape of S if and only if $\alpha_{min} < \alpha < \alpha_{max}$. The identification of these edges from the Delaunay triangulation of S is then as follows. Let $r = -1/\alpha$ ($\alpha < 0$) be an arbitrary number. For an edge P_iP_j we compute two numbers r_{min} and r_{max}. An edge P_iP_j belongs to the α-shape, if $r_{min} < r < r_{max}$. The computation of the numbers r_{min} and r_{max} is summarized as follows:

First case: *P_iP_j is common to two Delaunay circles*
If c_1, c_2 denote two Delaunay circles stored against P_iP_j then $r_{min} = 1/2\ d_e(P_i, P_j)$, $r_{max} =$ max (d_1, d_2) where $d_1 = d_e(c_1, P_i)$, $d_2 = d_e(c_1, P_j)$
Second case: *P_iP_j is associated to one Delaunay circle passes through P_i, P_j, P_k .*

Let c be the center of the Delaunay circle. r_{min} and r_{max} are computed by: $r_{max} = +\infty$. if c and P_k fall on the same side of P_iP_j then $r_{min} = 1/2 d_e(P_i, P_j)$ else $r_{min} = d_e(P_i, c)$. The proofs of this computation are presented in [6]. An example of the α-shape, obtained using the algorithm of section 2.1 is shown in Figure 4.

3 Skeletons and continuous Voronoi diagrams

3.1 Error estimates of an approximate skeletons

The approximation of the skeleton is computed from the Voronoi diagram of points. These points are obtained from a regular sampling of the shape. The principle of the algorithm, which computes the approximation skeletons of polygonal shapes, is based on the following three steps:

Sampling the set of polygonal shapes
At this level we define the sampling step h. Sampling seeds of each shape are affected by labels. For example, seeds of a same segment are affected by a same label.

Computing Voronoi diagram of seeds
For computing Voronoi diagram, we use the iterative algorithm described in [13]. Main advantage of this algorithm is the local modification of the Voronoi tessellation when we insert new seeds.

Deletion of useless sides
At this level, we delete sides having intersection with segments composing polygonal shapes. In other words, we delete sides which separate two same labeled seeds. An example of such sides is shown in Figure 5.

The continuous skeleton of the polygonal shape is composed by straight lines, parabola curves and half lines. In other words, it consists of bisectors of line segments. Hence, the error computation of an approximate bisector is sufficient to determine the error of an approximate skeleton. An example of a two line segments bisector is shown in Figure 6.

To compute the error of an approximate bisector, we consider two cases: 1) the error on the parabola portion. 2) The error on the straight line portion.

Let A and B be vertices of a straight line (denoted AB). The points $M_0, M_1, M_2, \cdots, M_n$ are obtained from a regular sampling of AB which are defined by:

$M_0 = A, M_n = B, M_{k+1} = M_k + h. \quad (B - A)/d_e(A, B)$ with $k = 0, \cdots, n-1$. and $h/2 < d_e(B, M_{n-1}) < 3h/2$.

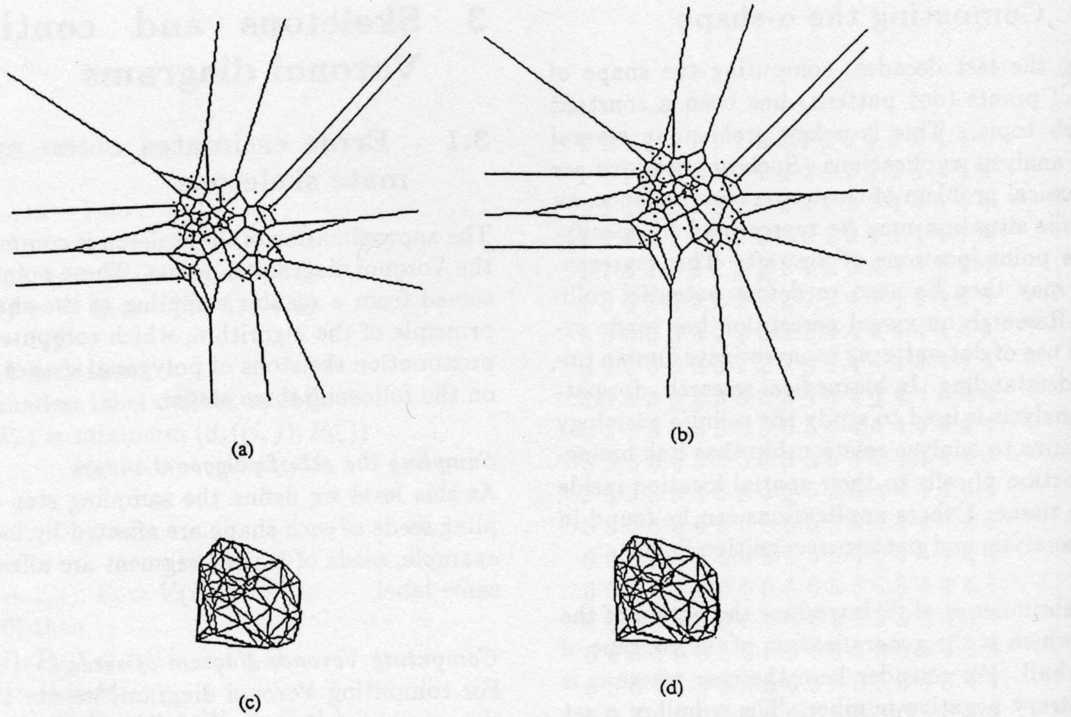

Figure 3. Continuous/Digital Voronoi diagam and Delaunay triangulation
(a) Digital Voronoi diagram of 53 seeds. (b) Continuous Voronoi diagram of the same random set of seeds. (c) Delaunay triangulation computed from the digital Voronoi diagram. (d) Delaunay triangulation computed from the continuous Voronoi diagram.

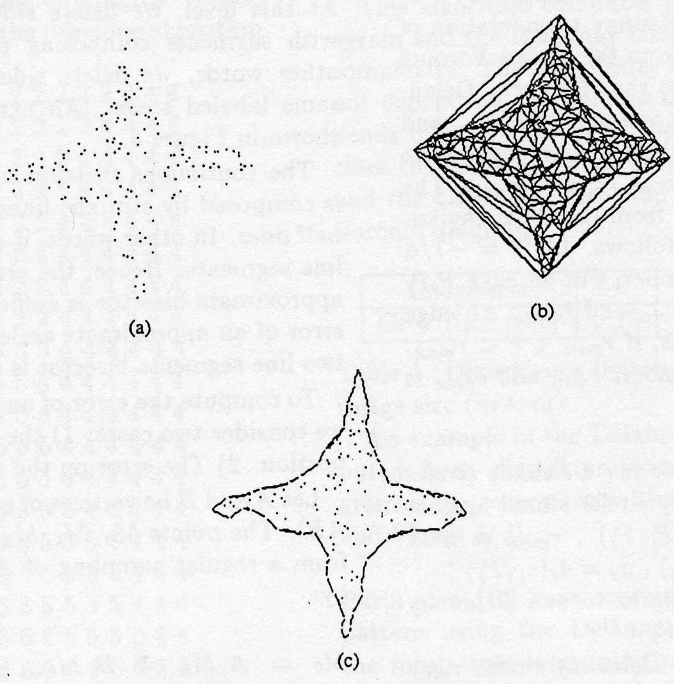

Figure 4. α-shape of a dot pattern
(a) The dot pattern. The dots are a random set of 160 seeds. (b) Edges of Delaunay triangulation. (c) The α-shape of the dot pattern for $r = -1/\alpha = 10$.

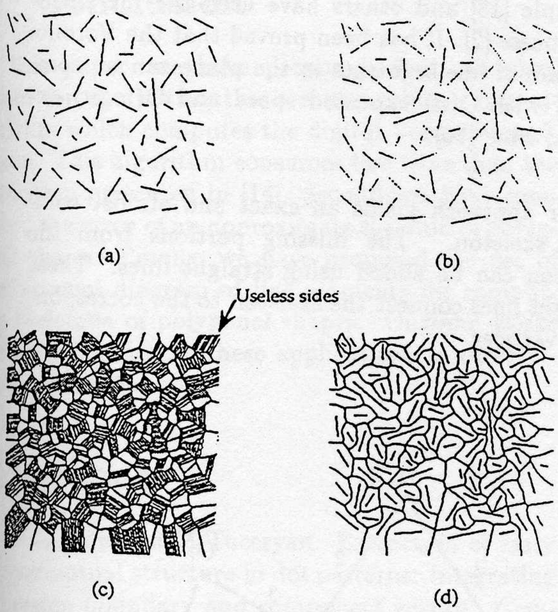

Figure 5. Approximation of Voronoi diagram of line segments (a) The set of line segments (b) Regular discretisation of the line segments (step = 2 pixels)(c) The Voronoi diagram computed from the points shown in (b). (c) The approximation of the Voronoi diagram of line segments.

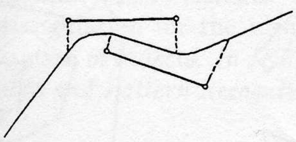

Figure 6. Bisector of two line segments

The error estimate in the parabola case

In the Voronoi diagram of line segments, the parabola portions are the locus of points equidistant from a point and a segment. Let M be a point and AB a straight line. Let us consider the same conditions illustrated in Figure 7.

$M_0 = A(0,0), M_1 ..., M_n = B(a,0)$ are the sampled seeds of AB. $B_0, B_1, ..., B_{n-1}$ are the Voronoi vertices $A_0, A_1, ..., A_{n-1}$ are the locus of intersections between the bisector of A_i and $A_{i+1} (i = 0, \cdots, n-1)$ and the line segment AB.

For each Voronoi vertex, we define the error as the difference between the approximated and the exact ordinates. The exact ordinate is computed from the parabola equation. The error is:

$$E_i = |y_{B_i} - y(x_{B_i})| = |\frac{h^2}{8y_0}|$$

The error of an approximate parabola curve by Voronoi edges behaves like $O(h^2)$. We point out that, the Voronoi vetices have the same distance to the exact bissetor and this distance decrease when the ordinate y_0 is sufficiently large.

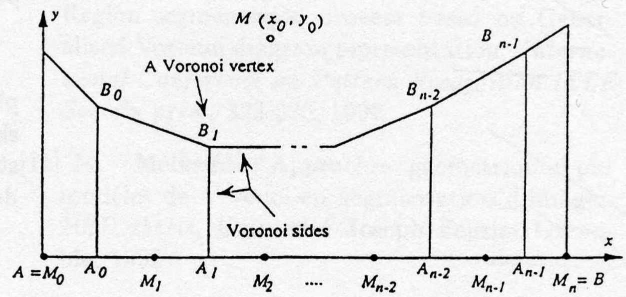

Figure 7. A portion of the bisector of a point M and a line segment AB.

The error estimate in the straight line case

Let AB and $A'B'$ two line segments. P_{ex} is an arbitrary point which lies in a straight line bisector of AB and $A'B'$. q_1, q_2 are the respective projection of P_{ex} onto the line segments AB and $A'B'$. We consider the nearest seeds M_1, M_2 to q_1 and M'_1, M'_2 to q_2. We assume that: $d_e(M_2, q_1) = \alpha.h$, $d_e(M_2, q_1) = (1-\alpha).h\ (0 \leq \alpha \leq 1)$ and $d_e(M'_2, q_2) = \beta.h$, $d_e(M'_2, q_2) = (1-\beta).h\ (0 \leq \beta \leq 1)$.

We define the error as: $E_h = d_e(P_{ex}, P_h)$. These conditions are shown in Figure 8. P_{ex} has an abscissa x_{q_2} and lies on the line perpendicular to the straight line $q_1 q_2$ and passes through the middle of $q_1 q_2$. Using simple operations, we can affirm this majoration: $E_h < C.h$, where C is a constant which depends on $\alpha, \beta, x_{q_1}, x_{q_2}, x_b, y_b$. The error of an approximate straight line bisector behaves like $O(h)$.

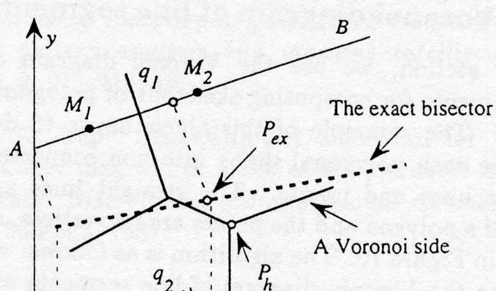

Figure 8.A straight line bisector of two line segments AB and $A'B'$. The broken dark line is the exact bisector. The solid lines are the edges of the Voronoi diagram.

Figure 9 shows the relationship between an approximate bisector and a step h.

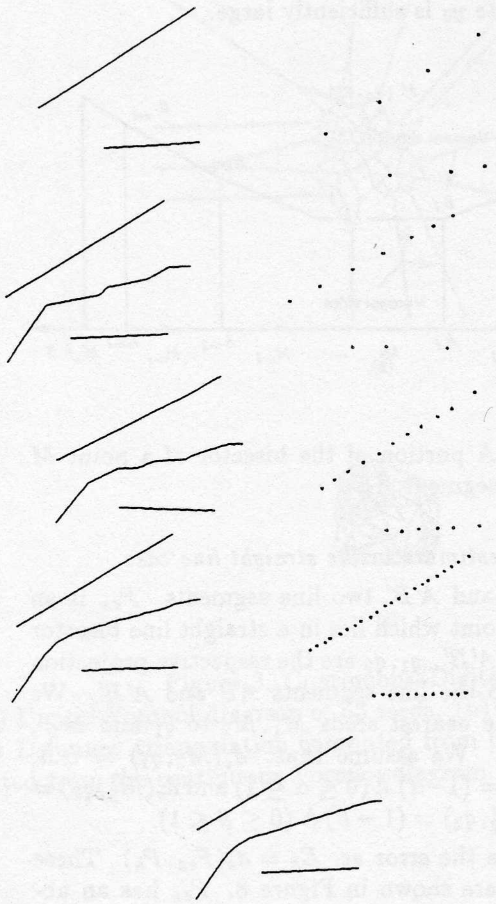

Figure 9. The approximated bisector computed from different step $h (h = 20, 15, 10, 5$ pixels).

3.2 Skeletons of polygons and Voronoi diagram of line segments

In this section, we use the Voronoi diagram of line segments for computing skeletons of polygonal shapes. The principle of this algorithm is to decompose each polygonal shape into non connected straight lines and points. The straight lines are edges of a polygon and the points are its vertices, as shown in Figure 10. The algorithm is as follows: we compute the Voronoi diagram of line segments obtained from the decomposition of the initial polygonal shapes. We then attach two labels for the points and one label for the line segments (see Figure 10). Next, we delete a bisector of elements (point and line segment) having a same label.

Several algorithms which compute the Voronoi diagram of line segments have been developed. Some algorithms are based on the divide and conquer principle [18] and others have used the incremental process [2]. It has been proved that the Voronoi diagram of line segments in the plane can be computed with $O(n)$ expected space and $O(\log n)$ expected time [16].

Our approach yields an exact and almost complete skeleton. The missing portions from the skeleton can be added using straight lines. These straight lines connect the skeleton to the correspondent vertices.

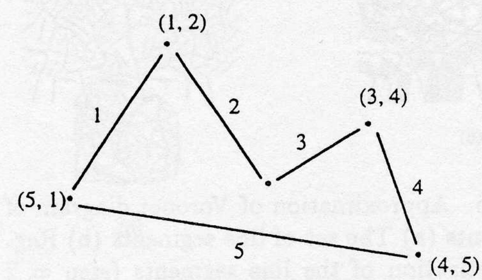

Figure 10. Elements obtained from polygonal shape

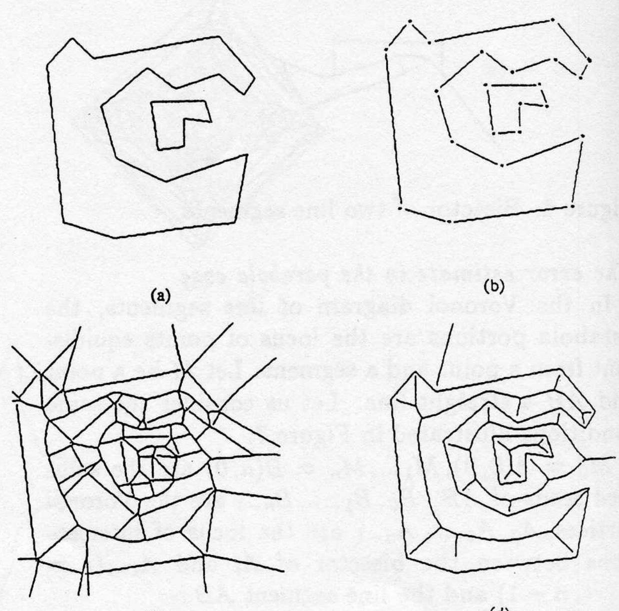

Figure 11. Continuous skeletons of two polygonal shapes (a) Two polygonal shapes (b) Elements obtained from the shapes (the step is 4 pixels) (c) The Voronoi diagram of line segments (d) The extracted skeletons.

4 Conclusion

In this paper, we have proposed, first, an algorithm for computing the α-shape using a fast algorithm which computes the digital Voronoi tesselation. This algorithm consumes less time than the algorithm presented in [14]. Second, we have computed the error of an approximate skeleton of polygonal shape. Finally, we have proposed the use of the Voronoi diagram of line segments for computing skeletons of polygonal shapes. Ongoing works are the extension of these applications to three dimensional space.

References

[1] N. Ahuja and M.Tuceryan. Extraction of early perceptual structure in dot patterns: integrating region boundary and component gestalt. *Computer Vision Graphics and Image Processing* 48, 304-356, 1989.

[2] H. Amet, J.D. Boissonnat and R. Schott. Calcul dynamique du diagramme de Voronoi d'un ensemble de segments. *Journées Géométrie Algorithmique* pages 143-153, INRIA Sophia-Antipolis, Nice, 1990.

[3] J.D. Boissonnat and P. Kofakis. Use of the Delaunay triangulation for the identification and the localization of objects. *In IEEE Conf. Computer Vision and Pattern Recognition* pages 398-401, 1985.

[4] G. Borgefors. Distance transformations in digital images. *Computer Vision Graphics and Image processing* 34, 344-371, 1986.

[5] P.E. Danielson. Euclidean distance mapping. *Computer Graphics and Image Processing* 14, 227-248, 1980.

[6] H. Edelsbrunner, D.G. Kirkpatrick and R.Seidel. On the shape of a set of points in the plane. *IEEE Trans. Inform. Theory* 29, 551-559, 1983.

[7] O.D. Faugeras, E. Le Bras-Melhman and J.D.Boissonnat. Representing stereo data with Delaunay triangulation. *Artificial Intelligence* 44,41-87,1990.

[8] H.T. Hu. Diagramme de Voronoi généralisé pour un ensemble de polygones *PHD.thesis.* Université Joseph Fourier Grenoble 1991.

[9] D.T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 4, 363-369, 1982.

[10] M. Melkemi and J.M. Chassery. Edge-Region segmentation process based on Generalized Voronoi diagram representation. *International Conference on Pattern Recognition IEEE Society press*, 323-326, 1992.

[11] M. Melkemi. Approches géometriques par modéles de Voronoi en segmentation d'images. *PHD thesis*, Université Joseph Fourier Grenoble, 1992.

[12] S.N. Meshkat and C. M. Sakkas, Voronoi diagram for multiply-connected polygonal domains II: implementation and applications *IBM J. Res. Develop.*, 31,3, 373-381 ,1987.

[13] T. Ohya, M.Iri, K. Murota. A fast Voronoi diagram algorithm with quaternary tree bucketing, *Information Processing Letters*, 18, 227-231, 1984.

[14] S.K. Parui, S. Sarkar and B. B. Chaudhuri. Computing the shape of a point set in digital images. *Pattern Recognition Letters*, 14, 89-94, 1993.

[15] V. Srinivasan and L.R.Nackman. Voronoi diagram for multiply-connected polygonal domains: Algorithm *IBM J. Res. Develop.*, 31,3, 361-372,1987.

[16] M. Teillaud. Towards dynamique randomized algorithms in computational geometry. *Technical Report INRIA*, 1727, 1992.

[17] M. Tuceryan and K.Jain. Texture segmentation using Voronoi polygons. *IEEE transactions on pattern analysis and machine intelligence*, 12, 2, 211-216, 1990.

[18] C.K. Yap. An O(*n*log*n*) algorithm for the Voronoi diagram of a set of simple curve segments *Discrete and Computational Geometry*, 2, 365-393, 1987.