Constructing Approximate Voronoi Diagrams from Digital Images of Generalized Polygons and Circular Objects

Waldir L. Roque and Dionísio Doering
Instituto de Matemática
Universidade Federal do Rio Grande do Sul
91509-900, Porto Alegre, Brazil
[roque,ddoering]@mat.ufrgs.br

ABSTRACT

In this paper we present the geometrical construction of an approximate generalized Voronoi diagram for generalized polygons and circular objects based on their minimum geometrical structure that are extracted from the object's digital image. The construction is done in $O(n)$ time complexity, where n is the number of single points defining the set of objects. An application of this technique has been done for mobile robot path planning.

Keywords

Voronoi diagrams, Generalized polygons, Digital images.

1. INTRODUCTION

The geometric construction of Voronoi diagrams has an extensive literature (see [Aur91,Oka92] and references therein). The image-based construction of Voronoi diagrams for a set of digital points has been treated in [Par93,Bor86,Mel92] and for extended digital shapes in [Arc86,Mel94,Sud99]. In these papers the main approach to compute the Voronoi diagram is based on labeling the connected components of the objects and in the application of the morphological operation of shape dilation. By this means, the Voronoi edges are found between two adjacent objects when two different labels are met. The computational cost is of the order of $O(n^2)$, where n^2 is the image size.

The construction of the Voronoi diagram for a set of digital shapes is actually an approximate diagram due to the fact that the objects are constituted of pixels, which have a discrete structure. In this paper, we discuss an alternative approach to construct the Voronoi Diagram for generalized polygons and circular objects based on their minimum geometrical structures, which are extracted from their digital images. The principle is quite simple. A simple polygon is fully characterized by its ordered sequence of vertices and an arc segment can be approximated by the set of vertices that form its polygonal line approximation. Therefore, a generalized polygon, whose edges are straight lines and arc segments, can ultimately be characterized by a set of vertices, too.

In the section 2 a brief introduction to the geometric construction of planar Voronoi diagrams is given, in the section 3 the image processing required to identify the minimum geometrical structure of generalized polygonal objects and circles is presented; section 4 briefly report the application to robot path planning and section 5 is devoted to conclusions.

WSCG SHORT PAPERS Proceedings
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

2. VORONOI DIAGRAM

In what follows, we provide a brief introduction to the geometrical construction of planar Voronoi diagrams.

2.1 Ordinary Voronoi diagrams

The planar ordinary Voronoi diagram (OVD) [Oka92] is defined as a partition of the plane into regions according to the principle of the *nearest neighbor*. More precisely, let consider

$P = \{p_1, p_2, \ldots, p_n\}$ a set of non-collinear points in the plane and let consider $d(p, p_i)$ the Euclidean distance from a point $p \neq p_i$ to a point p_i. The Voronoi region $R(p_i) \equiv R_i$ generated by the point p_i is defined as

$$R_i = \{p; d(p, p_i) \leq d(p, p_j), \forall p_i \neq p_j\} \quad (1)$$

.

The Voronoi diagram $V(P)$ for a set of points $P = \{p_1, p_2, \ldots, p_n\}$, is defined as the union of all Voronoi regions $V(P) = \bigcup_{i=1}^{n} R(p_i)$. The points p_i are called Voronoi generators, the edge common to two Voronoi regions is called a Voronoi edge and the vertices where three or more Voronoi edges meet are called Voronoi vertices. We say that a Voronoi generator p_i is adjacent to p_j when their Voronoi regions share a common edge. According to this definition, the Voronoi diagram is such that any point on the edge of two neighboring regions is equidistant from the corresponding Voronoi generators.

2.2 Generalized Voronoi diagram

The planar OVD can be extended to objects like straight lines and arcs segments or polygons and circles. The diagram for these objects is called the generalized Voronoi diagram (GVD).

2.2.1 GVD for lines and arcs

Let $L = \{L_1, L_2, L_3, \ldots, L_n\} \subseteq R^2$ be a set, where L_i can be a straight line or arc segment, such that $L_i \cap L_j \neq 0$, for $i \neq j$. Let us define the distance from a point p to L_i by the shortest distance between p and a point p_i on L_i:

$$d_s(p, L_i) = \min_{x_i} \| \mathrm{x} - \mathrm{x}_i \|; \mathrm{x}_i \in L_i \quad (2)$$

where x and x_i are the position vectors of p and p_i, respectively. The Voronoi region $R(L_i)$ is given by

$$R(L_i) = \{p; d_s(p, L_i) \leq d_s(p, L_j), j \neq i, j \in I_n\} \quad (3)$$

The union of all Voronoi regions $V(L) = \bigcup_{i=1}^{n} R(L_i)$ generates the line Voronoi diagram for the set L.

2.2.2 GVD for polygons and circles

An extension of the line Voronoi diagram for the case of polygonal (simple or generalized[1]) objects can be done considering the generation of the Voronoi diagram for areas. Let $A = \{A_1, A_2, \ldots A_n\}$ in R^2 be a set of areas. Assuming that the areas are connected closed sets with unity Euler number and that they do not intersect each other, we define the distance from a point p to A_i as the shortest distance from p to p_i on A_i as follows:

$$d_s(p, A_i) = \min_{\mathrm{x}_i} \|\mathrm{x} - \mathrm{x}_i\|; \mathrm{x}_i \in A_i, \quad (4)$$

where x and x_i are the position vectors of p and p_i, respectively. According to this distance, we may define the Voronoi regions $R(A_i)$ associated to each area as

$$R(A_i) = \{p; d_s(p, A_i) \leq d_s(p, A_j), j \neq i, j \in I_n\}. \quad (5)$$

The area Voronoi diagram is the set $V(A) = \bigcup_{i=1}^{n} R(A_i)$. The area Voronoi diagram can be seen as the diagram for generalized polygons, where the generalized polygons can be represented by their area contours. Note that the area Voronoi diagram subsumes the line and the ordinary Voronoi diagrams.

2.3 Computational generation of the GVD

Several algorithms have been proposed to generate the planar Voronoi diagram for a set of objects [Aur91]. An interesting algorithm was proposed by Sugihara and Iri [Sug92], based on an incremental construction, to generate the OVD with average running time complexity of $O(n)$, where n is the number of Voronoi generators, that is also stable to numerical errors.

The construction of the OVD based in [Sug92] starts with a trivial diagram for three generators and adds up a new generator one by one at a time. To generate the new Voronoi region $R(p_l)$ for p_l one need first to identify the generator p_i, in whose region $R(p_i)$ the new generator p_l is contained in. Then one draw the perpendicular bisector between p_l and p_i until it intersects the edges of $R(p_i)$. The bisector intersects the edges of $R(p_i)$ in two points. Let call q be one of them. Now take the perpendicular bisector of p_l and p_j, starting from q till it intersects another edge of $R(p_j)$. This procedure should be followed for all Voronoi generators adjacent to p_l until we get back to the region $R(p_i)$. Now, removing the edges enclosed by the closed sequence of bisectors the Voronoi region for the new generator p_l is found. Figure 1 shows the OVD for a set of 14 Voronoi generators. Figure 2 illustrates the procedure to

[1] Generalized polygons are polygons whose edges are straight lines or arc segments.

generate the new Voronoi region $R(p_{15})$, when the generator P15 is inserted.

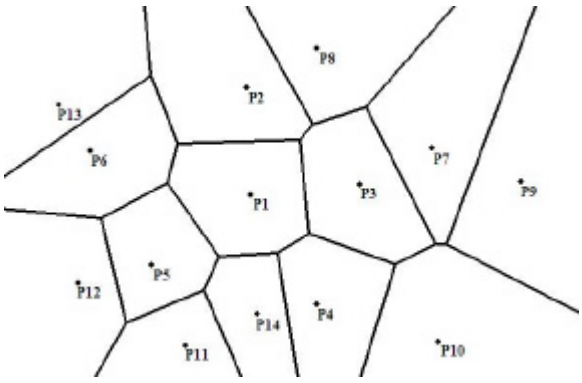

Figure 1: OVD for 14 Voronoi generators.

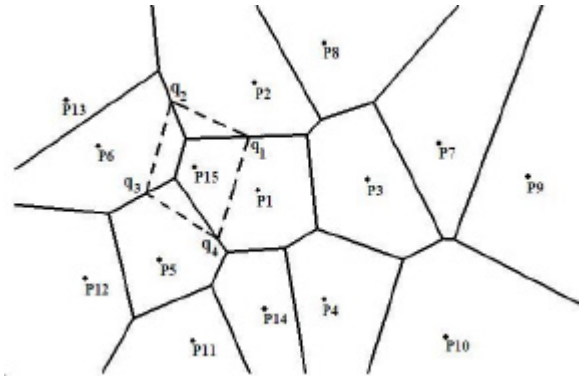

Figure 2: Construction of Voronoi region $R(p_{15})$ for the new generator P15.

2.3.1 Constructing approximate GVD

An extension of the incremental type algorithm described above can be done to construct an approximate generalized Voronoi diagram (AGVD) [Sug93] for straight lines, arc segments, generalized polygons and circles. For that, let us consider a line or arc segment approximated by a sequence of n points with a constant small displacement δ between them. In other words, we are doing a polygonal approximation of the segments with a variable number of vertices. The OVD algorithm can be applied to generate the AGVD for these objects if the Voronoi edges that cross the line or arc edges be invalidated and omitted, remaining only the Voronoi edges of adjacent objects. The same approach can be applied to generalized polygons and circles.

Figure 3 shows the AGVD for a set of generalized polygonal objects in a bounded square region. In this picture all Voronoi edges can be seen and the resolution displacement parameter δ was fixed, but it can be arbitrarily adjusted at the expense of computational cost. Figure 4 shows the corresponding AGVD omitting the invalid Voronoi edges.

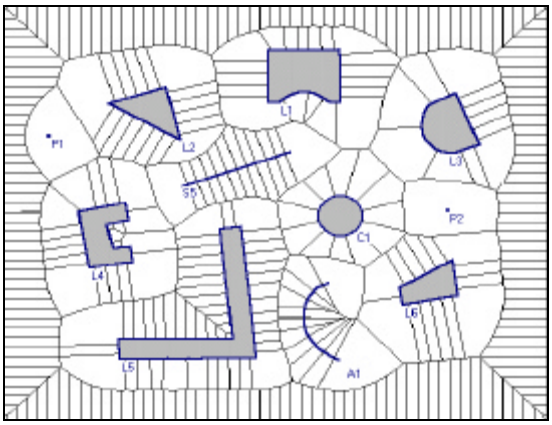

Figure 3: AGVD for a set of objects, including the invalid Voronoi edges in a bounded region.

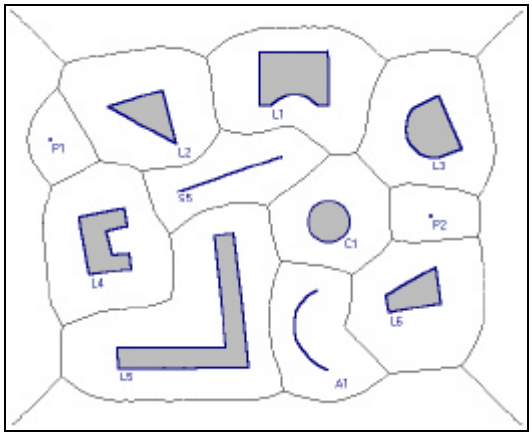

Figure 4: AGVD for objects omitting the invalid Voronoi edges.

3. GEOMETRICAL STRUCTURE FROM DIGITAL IMAGES

The construction of the GVD for a set of objects in a digital picture is actually an approximate diagram due to the fact that the objects are constituted of pixels, which correspond to a discrete structure. Remind that, in a 8-neighborhood scheme, the distance from the central pixel to its neighbors is either 1 or $\sqrt{2}$.

Several algorithms have been proposed to compute the Voronoi diagram from a set of digital points [Par93, Mel92] or for extended digital shapes [Arc86, Bor86, Sud99]. The main approach used by them to compute the Voronoi diagram was based on the principle of labeling the objects and applying the morphological operation of shape dilation to grow the objects. The Voronoi edge is formed between two adjacent objects when two different labels are met.

According to the AGVD algorithm construction discussed in section 2.3, an alternative approach to construct the diagram for a set of digital objects is to find out the pixels that form the border of the objects. Provided with the sequence of pixels of the border of an object the AGVD can be computed. Nevertheless, as was pointed in the previous section, the computational cost becomes high due to the large number of single points. One alternative to diminish the computational cost is to get rid of a number of pixels of the border, but this procedure has a drawback and requires caution because while doing that we may loose the actual geometrical structure of the object. For instance, if some vertices are dropped out. Therefore, in order to preserve the this structure it is necessary to identify the minimal geometrical structure that characterize each one of the objects.

3.1 Vertex Detection

Let Q be a simple polygon and $V = \{v_0, v_1 \ldots, v_{n-1}; \mathrm{mod}(n)\}$ be its ordered sequence of vertices. A simple polygon can be minimally characterized by the set V. The edges of a simple polygon are straight line segments. Generalized polygons can also be characterized by its ordered sequence of vertices, however now the arc edges connecting two vertices must be specified.

To identify the minimum geometric structure of a generalized polygon P, from its digital image, one need to detect the pixels that form its ordered sequence of vertices. Let $S = \{I_1, I_2, \ldots, I_n\}$ be a digital image containing a set of generalized polygons. To get the pixels that form the border of a generalized polygon from its digital image I_k, first the digital image is segmented by threshold. In our case, we binarize the image as we are assuming that the objects are black on a white background. Once binarized, we apply the edge detection algorithm based on the standard second-order Laplacian operator, with mask $M = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$, to find the list L containing the border pixels of all objects $I_k, k = 1, \ldots, n$.

To obtain the ordered sequence of border pixels for each individual object a modified version of the contour following algorithm given in [Cos01], based on the 8-connected neighbor approach, is applied on the list L. The modified algorithm takes care of the chain-code direction in a clockwise manner to properly find the next contour pixel along the edge. After completion of this algorithm, each object has been labeled and their ordered sequence of pixels have been obtained.

To be able to get the ordered sequence of vertex pixels of an object, a corner detection algorithm shall be applied on the set of pixels that form each object. There are several algorithms in the literature for corner detection (see [Cos01] and references therein). Recently, Tsai et al. [Tsa99] proposed an algorithm, with an average running time complexity of $O(n)$, for corner detection that is simple to implement, robust to noise and sensible to identify both convex and nonconvex vertices. The algorithm relies in the analysis of the covariance matrix eigenvalues of a digital curve segment. It takes the sequence of border pixels of an object $P = \{(x_i, y_i), i = 1,2,\ldots,n\}$, where the pixel p_{i+1} is the neighbor of $p_i, \mathrm{mod}(n)$ and (x_i, y_i) is the Cartesian coordinate of the pixel. A region of support around the pixel p_i is defined as $S_k(p_i) = \{p_i; j = i-k, i-k+1, \ldots, i+k-1\}$, where k is an integer number that defines the length of the support region. The covariance matrix is given by

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}, \tag{6}$$

where

$$c_{11} = \left(\frac{1}{2k+1} \sum_{j=i-k}^{i+k} x_j^2 \right) - c_x^2, \tag{7}$$

$$c_{22} = \left(\frac{1}{2k+1} \sum_{j=i-k}^{i+k} y_j^2 \right) - c_y^2, \tag{8}$$

$$c_{12} = c_{21} = \left(\frac{1}{2k+1} \sum_{j=i-k}^{i+k} x_j y_j \right) - c_x c_y, \tag{9}$$

$$c_x = \left(\frac{1}{2k+1} \sum_{j=i-k}^{i+k} x_j \right), \tag{10}$$

$$c_y = \left(\frac{1}{2k+1} \sum_{j=i-k}^{i+k} y_j \right). \tag{11}$$

The covariance matrix is Hermitian with real eigenvalues, which are given by:

$$l_L = \frac{1}{2}\left(c_{11} + c_{22} + \sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2} \right), \tag{12}$$

$$l_S = \frac{1}{2}\left(c_{11} + c_{22} - \sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2} \right). \tag{13}$$

The analysis of the eigenvalue l_S shows that a corner is detected when its value is greater than a predefined threshold value. Each corner is separated

by at least k pixels. It has been experimentally observed that pixels on a straight line have their l_S value very close to zero and l_S much greater correspond to a corner. After these observations we have chosen the region of support parameter with $k = 7$ and the threshold value for detecting a corner (vertex) as 10^{-1}. According to Tsai's paper, circular shapes would have $\frac{l_S}{l_L} \approx 1$. However, unfortunately, our experiments have shown that the algorithm is not reliable to identify circles as was claimed.

To characterize digital arc edges in generalized polygons, the eigenvalue l_S is used to identify a minimum set of vertices in order to approximate the arc segment by a polygonal curve. Let $A = \{e_1 = v_1, e_2, \ldots, e_m = v_m\}$ be the set of vertices that characterize an arc edge of a generalized polygon P, then its geometric structure will be given by the ordered sequence of vertices $\{v_0, \ldots, v_k = e_k, e_{k+1}, \ldots, e_{k+m} = v_{k+m}, v_{k+m+1}, \ldots, v_{n-1}\}$.

3.2 Identifying Digital Circles

The geometric structure of a circle can be minimally characterized by its center $C = (x_0, y_0)$ and radius r. To recognize digital circles an algorithm was proposed by Sauer [Sau93] with linear time complexity $O(n)$, where n is the number of pixels of the digital function. In short, the algorithm is as follows: Let $C = (r, x_m, y_m)$ be a digital closed curve, where (x_m, y_m) is the centroid coordinate and r its radius mean value, then to the digital curve be a digital circle the following relation has to be fulfilled:

$$p_y - 0.5 - y_m \le \pm\sqrt{r^2 - (p_x - x_m)^2} \le p_y + 0.5 - y_m \tag{14}$$

where (p_x, p_y) are the pixel coordinates. Essentially, the identification of a circle translates to the computation of the centroid and the radius under a threshold value. The implementation of this algorithm was done to decide if the sequence of pixels of the border of an object form a digital circle.

The identification of the ordered sequence of vertices V of a generalized polygon, and of the center and radius of a circle, provide the minimal characterization of their geometrical structure and from that it allows its full reconstruction.

3.3 The Minimum Geometric Data Structure

Based on the geometric data structure extracted from the digital image of the objects, the AGVD can be constructed according to the algorithm provided in section 2.3. The geometrical data structure of the objects are given in the following format:

P (x,y)

S (x0,y0) (x1,y1)

C (xc,yc) r

A N (x1,y1)...(xn,yn)

L N (x1,y1)...(xn,yn),

where P specifies the Cartesian coordinate (x, y) of a single pixel (point); S corresponds to a straight line segment from pixel $p_0 = (x_0, y_0)$ to pixel $p_1 = (x_1, y_1)$; C gives the pixel coordinates $C = (x_c, y_c)$ of the center of a circle of radius r; A gives the sequence of n vertices from $p_1 = (x_1, y_1)$ to $p_n = (x_n, y_n)$ of the polygonal line that forms an arc segment; finally, L gives the sequence of n vertices that forms a generalized polygon.

Figure 5 exhibits, on the left side, the actual digital image of a set of generalized polygonal objects, captured by a CCD camera disposed on the top center of an arena, and on the right side the corresponding AGVD. There are 4 simple polygons, 2 generalized polygons and 1 circular object in the picture. The total number of pixels that form the border of all objects is $n = 1661$. The application of the algorithm to identify the minimal geometrical structure reduces this number of pixels (or points) to $\mathrm{m} = 60$. The number of Voronoi generators to construct the AGVD, according to the current value of the resolution displacement parameter d, is given by $\mathrm{p} = 141$. We can see that by this approach only 8.4% of the total number of pixels were sufficient to construct the AGVD.

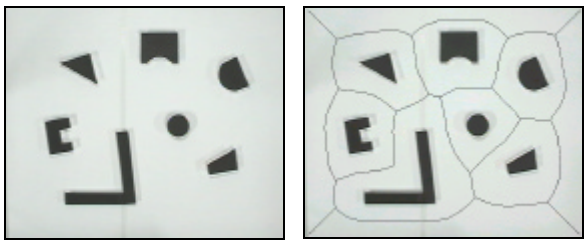

Figure 5: AGVD for a digital image with generalized polygons and a circular object.

Considering only the application of the algorithm for the AGVD construction base in the number of pixels, we can see that there would be a gain of 91.6% in computational time. Nevertheless, the overall computation cost to construct the AGVD from the digital image changes only after the computation of the total number of border pixels of the objects. In the full pixel approach the AGVD could be generated straightway from the set of all pixels that form the borders. On the other hand, in the reduced pixel approach, the AGVD construction is based on the identification of the minimal geometrical structure, which needs the application of the vertex detection algorithm of section 3.1, to reduce the number of pixels.

Full Pixel		Reduced Pixel	
# Pixels	Time(ms)	# Pixels	Time(ms)
260	2690	24	110
393	5600	33	270
559	10660	47	380
691	15600	55	490
763	18790	66	660

Table 1: The table illustrates the gain obtained by the reduced pixel approach over the full pixel approach.

Therefore, despite of both algorithms have a linear average running time complexity, the overall computational cost depends on the hidden constant factors, which for the AGVD using the minimal geometrical structure still provides a smaller average running time than just computing it considering the full pixel approach. This fact has been observed comparing the CPU time for the AGVD construction based on both approaches. Table 1 gives the time versus pixel number relation for both approaches. The linear fit of each approach shows that the AGVD construction based on the reduced pixel proposed here is better than the full pixel approach by an order of 38%. Figure 6 shows the graph of the linear fit curves taking into account only the slopes.

4. ROBOT APPLICATION

The generalized Voronoi diagram technique has been applied in many different areas. One of these applications can be seen in the field of mobile robotics, where the problem of collision-free path planning plays a central role for the robot safe navigation.

If we consider a global vision system composed of a single CCD camera posed on the top center of an arena, where the arena is seen as the workspace and the objects correspond to the obstacles, the AGVD provides a roadmap with maximal clearance from the obstacles. In addition, in a closed 2D workspace the AGVD is fully connected and any configuration (position and orientation) in the free configuration space C_{free} can be accessed by a robot navigating on the roadmap and then departing to reach the specified configuration. The Voronoi roadmap can be seen as a graph, therefore the shortest path between two configurations in C_{free} can be computed.

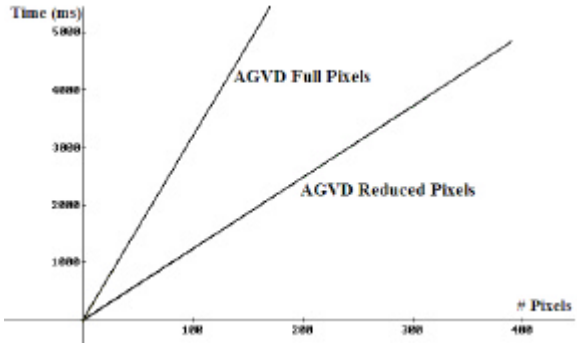

Figure 6: Linear fit curves for the AGVD construction based on the full pixels and reduced pixels approaches.

Considering a disk-like robot with radius r and taking into account that the Voronoi edges are the maximal clearance paths for the robot collision-free navigation, this will provide a natural threshold for the resolution displacement parameter as at most $d < 2r$. The Voronoi edges that are invalidated and omitted in the AGVD would, otherwise, lead to the robot collision with the obstacles.

Based on the approach described in the previous sections, a Global Vision module has been developed to provide a roadmap. In addition to this module, a Trajectory Planning module and a Navigation Control module have been developed and integrated in a mobile robot path planning system (see [Roq02] for further details).

5. CONCLUSION

In this paper we have shown that based on the geometrical structure that are extracted from the digital images of generalized polygons and circular objects, the construction of the approximate generalized Voronoi Diagram can be done with a running time complexity of $O(n)$, where n is the number of pixels defining the objects. The approach proposed here reduces the number of pixels improving the gain in computational time of the order of 38%, as pointed out in the previous section. In addition, this algorithm is also robust to numerical errors [Sug92]. The reduction in the number of

pixels depends on the number of vertices representing the objects and on the resolution displacement parameter d , whose value can be arbitrarily adjusted at the expense additional of computational cost.

An application of the reduced pixel approach has been done for robot path planning, where a global vision module was developed to capture the robot workspace image, identify the geometrical structure of the obstacles and the robot configuration, and finally generate the AGVD roadmap [Roq02].

6. ACKNOWLEDGMENTS

The authors thank Prof. Rudnei Dias da Costa for some comments on the paper, Bruno Alves for his help and the Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul – FAPERGS, for partial financial support.

7. REFERENCES

[Arc86] Arcelli, C. and Sanniti di Baja, G., Computing Voronoi Diagrams in Digital Pictures, Pattern Recognition Letters, 4, pp. 383-389, 1986.

[Aur91] Aurenhammer, F., Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure. ACM Computing Survey, 23, pp.345-405, 1991.

[Bor86] Borgefors, G., Distance Transformation in Digital Images, Computer Vision Graphics and Image Processing, 34, pp. 344-371, 1986.

[Cos01] Costa, L. F. and Cesar Jr., R. M., Shape Analysis and Classification: Theory and Practice, CRC Press, Boca Raton, USA, 2001.

[Mel92] Melkemi, M. and Chassery, J. M., Edge-Region Segmentation Process based on Generalized Voronoi Diagram Representation. IEEE International Conference on Pattern Recognition, pp. 323-326, 1992.

[Mel94] Melkemi, M. and Vandorpe, D., Fast Algorithm for Computing the Shape of a Set of Digital Points, IEEE International Conference in Image Processing, 1, pp. 705-709, 1994.

[Oka92] Okabe, A., Boots, B. and Sugihara, K., Spatial Tesselation Concepts and Applications of Voronoi Diagrams. John Wiley & Sons, New York, 1992.

[Par93] Parui, S. K., Sarkar, S. and Chaudhuri, B. B., Computing the Shape of a Point Set in Digital Images, Pattern Recognition Letters, 14, pp. 89-94, 1993.

[Roq02] Roque, W. L. and Doering, D., Multiple Visiting Stations Trajectory (Re)Planning for Mobile Lab Robots Based on Global Vision and Voronoi Roadmaps, Proceedings of the 33rd International Symposium on Robotics, Stockholm, Sweden, 2002.

[Sau93] Sauer, P., On the Recognition of Digital Circles in Linear Time. Computational Geometry: Theory and Applications, 2, pp. 287-302, 1993.

[Sud99] Sudha, N., Nandi, S. and Sridharan, K., A Parallel Algorithm to Construct Voronoi Diagram and Its VLSI Architecture, Proceedings of IEEE International Conference on Robotics and Automation, pp. 1683-1688, Detroit, US, 1999.

[Sug92] Sugihara, K. and Iri, M., Construction of the Voronoi Diagram for One Million Generators in Single-Precision Arithmetic. Proceedings of the IEEE, 80, pp. 1471-1484, 1992.

[Sug93] Sugihara, K., Approximation of Generalized Voronoi Diagrams by Ordinary Voronoi Diagrams. CVGIP: Graphical Models and Image Processing, 55, 522-531, 1993.

[Tsa99] Tsai, D.-M., Hou, H.-T. and Su, H.-J., Boundary-Based Corner Detection Using Eigenvalues of Covariance Matrices. Pattern Recognition Letters, 20, pp. 31-40, 1999.