

# The identification of textured region boundaries

K Krishnasamy and M Petrou

Dept. of Electronic and Electrical Engineering,  
University of Surrey, Guildford GU2 5XH, United Kingdom.

e-mail: ees1mp@ee.surrey.ac.uk

## Abstract

In this paper, we present a novel algorithm which is based on the concept of free angle for the purpose of texture region boundary identification. The input to the algorithm is a grey level image. The output is a binary edge map marking the closed contour boundaries of the textured regions. The algorithm has been tested on several remotely sensed images and gives very good and robust results.

## 1 Introduction

The concept of texture in an image is a concept which cannot be easily defined in a formal way and yet it plays very important role in human perception. The major reason for the difficulty of devising a formal definition is the dependency of the concept on the image resolution. Thus, to some appropriately chosen scale, *all* surfaces are textured. And yet, the humans have no difficulty in distinguishing between textured and non-textured regions in any image. This ability to discriminate surfaces in these two major categories is very important for guiding the attention of the vision system to the potentially interesting regions of the image, according to the recognition task that has to be performed. The usual understanding is that humans identify the textured regions from the fact that there is rapid variation of the intensity function in those regions. The mathematical concept with which computer vision scientists and mathematicians express this variation is through the calculation of the first derivative of the image intensity and in particular through the process of edge detection. Thus, the definition of a textured region becomes synonymous to a region with high density of edgels. A textured region then is nothing more than a cluster of edgels. The problem of isolating the textured regions in an image reduces then to the problem of finding the boundary set of a point cluster. The human vision system has an incredible ability to see patterns of dots and make associations according to some simple rules of Gestalt Psychology (eg [6, 12, 13]). There have been many algorithms proposed in the literature for the analysis of dot patterns (eg [1, 5, 7, 11, 14, 15, 16]) In particular the concept of the boundary of a dot pattern has been investigated within the problem of clustering in multidimensional feature spaces (eg [3, 8, 9, 10]).

In this paper, we shall build upon the ideas of [9] and [10] and apply them to the problem of the identification of texture boundaries in two dimensional im-

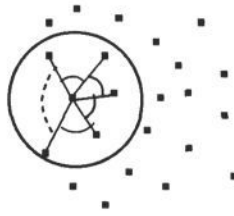


Figure 1: A circular neighbourhood around an edgel and the various viewing angles it has between its neighbours. The largest of them (marked with a dotted curve) is the free angle associated with the edgel.

ages. Our algorithm is based on the concept of the free angle borrowed from Computational Mathematics and on the use of morphological operators. It will be demonstrated with the help of some remotely sensed images. Indeed, in Remote Sensing problems, the case often arises of identifying the texture regions, finding their boundaries and either process them further separately from the other regions, or set them aside as regions of no further interest (eg [2, 4]). For example, towns and forests may pose unwanted interference in the identification of fields and pastures for agricultural applications of computer vision [2], or they may require a different processing mechanism for the identification of road networks etc [4].

In the next section we shall present the concept of the free angle and the algorithm will be presented in detail in section 3. Finally, we shall present the results of applying the algorithm on some remotely sensed images and conclude in section 4.

## 2 The concept of the free angle

As we argued earlier, textured regions are regions with high density of edge pixels. Let us consider the edge map of an image computed by using the Sobel operators, say. If we concentrate on one edgel in this edge map, its “view” of the world around it will be obstructed by the presence of other edgels in its neighbourhood. Thus each edgel is treated like an individual whose field of view is obstructed by the presence of other individuals around it. Therefore, for each edgel, we can compute the angle of the field of view it has between any two of its neighbours. The concept of neighbourhood is defined as a circular region around the edgel and neighbours are all the edgels within that region. The angles we compute have all as vertex the edgel in the middle and are defined by any two other edgels so that no other edgel falls inside the angle. The largest of these angles is defined as the free angle of the edgel at the centre. These ideas are demonstrated in figure 1.

It is clear now, that edgels in textured regions will be surrounded by so many other edgels that they will have small free angles associated with them, while edgels near the perimeter of the point clusters will have large free angles as demonstrated in figure 2. It is also clear that the concept of the neighbourhood plays important role in the determination of the value of the free angle associated with each pixel. This is demonstrated in figure 3. Thus, the choice of the neighbourhood size plays crucial role to the performance of the algorithm presented in the next section.

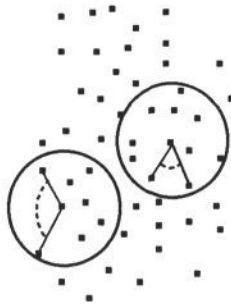


Figure 2: The circular neighbourhoods and the corresponding free angles associated with two edgels: The edgel near the boundary of the point cluster has a larger free angle than the internal edgel.

### 3 The algorithm

From the definition of the free angle given in the previous section, it becomes obvious that in order to identify the boundary pixels of a point cluster, all we have to do is to calculate the free angle associated with each pixel and threshold the free angle values appropriately: Edgels with values larger than the threshold are the boundary edgels, while edgels with the free angle less than the threshold are interior to the textured region and can be omitted. It happens, however, that large free angles are associated with edgels which form the boundaries between non-textured regions as well. To discriminate between them we perform the morphological operations of erosion and dilation of the edge map. That is, around each boundary edgel we switch off all edgels within a certain radius. Then, around the edgels that are left over we switch on the edgels again. This way, we eliminate small clusters of edgels as well as linear arrangements of edgels that correspond to hedges, roads etc.

Finally, the boundary edgels have to be connected into a continuous string to form the closed textured regions. The connecting process can be aided by the information of the free angle associated with each pixel. In essence, the free angle gives us the information as to which “side” of the pixel is the outside and which is the inside. Thus, any two nearby pixels can be connected provided the connecting segment does not cross through the “field of view” of another boundary pixel.

The basic steps of the algorithm are the following:

1. Create the edge map of the image using simple differencing operators.
2. Compute the free angle for each edgel and threshold the edge map keeping only those edgels which have free angle larger than a certain threshold, thus creating a preliminary boundary map.
3. Erode the edge map of the image by considering a circular area around each boundary pixel and switching off all the edgels within the area.
4. Dilate the eroded edge map of the image by considering a circular area around each pixel and switching on all the edgels within the area.

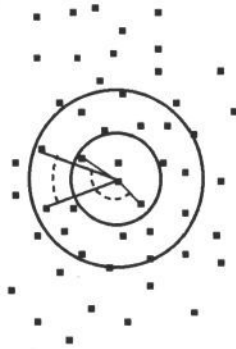


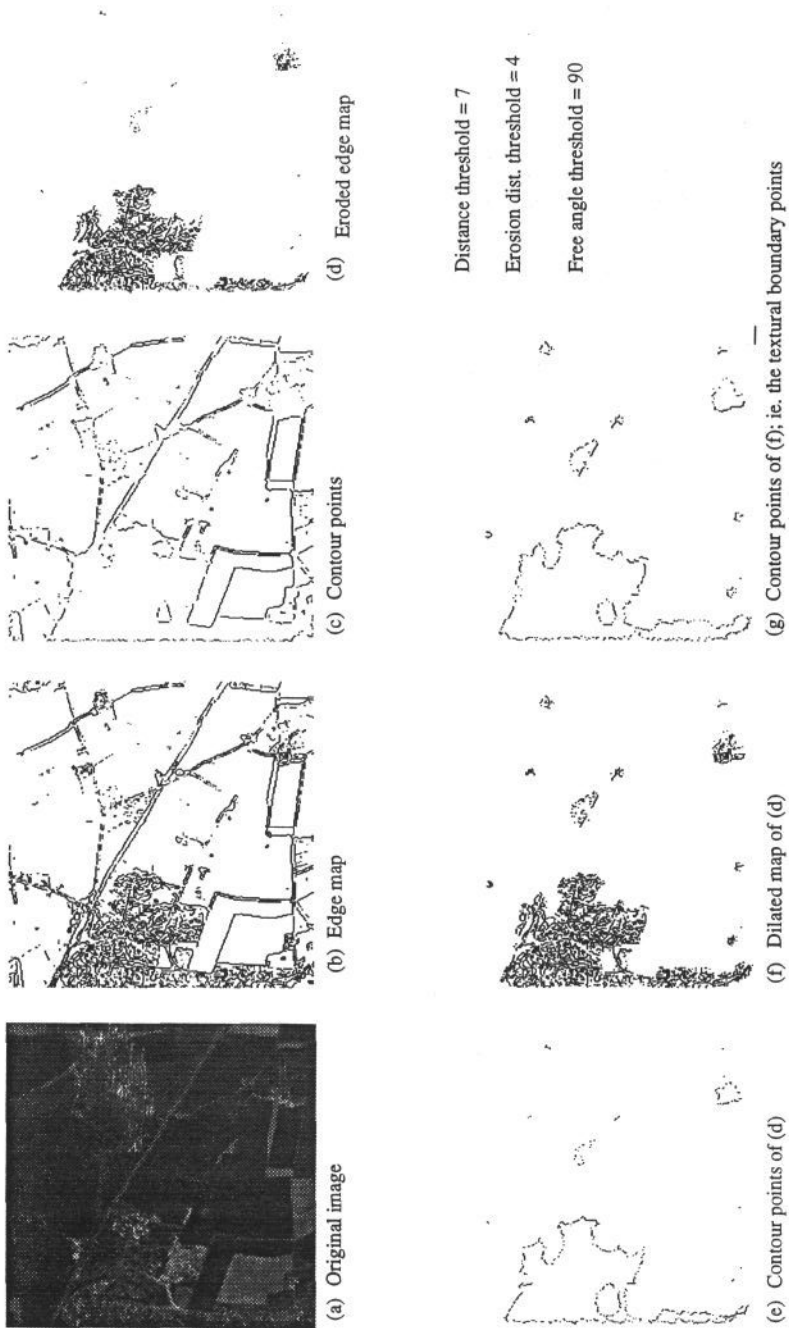
Figure 3: Different free angles may be associated with the same edge for different neighbourhood sizes.

5. Compute the free angle of the eroded and dilated edge map, threshold and thus create the final boundary map.
6. Connect the boundary pixels by filling the gap between two nearby edgels provided no other boundary edgel “sees” the world through the gap between them.

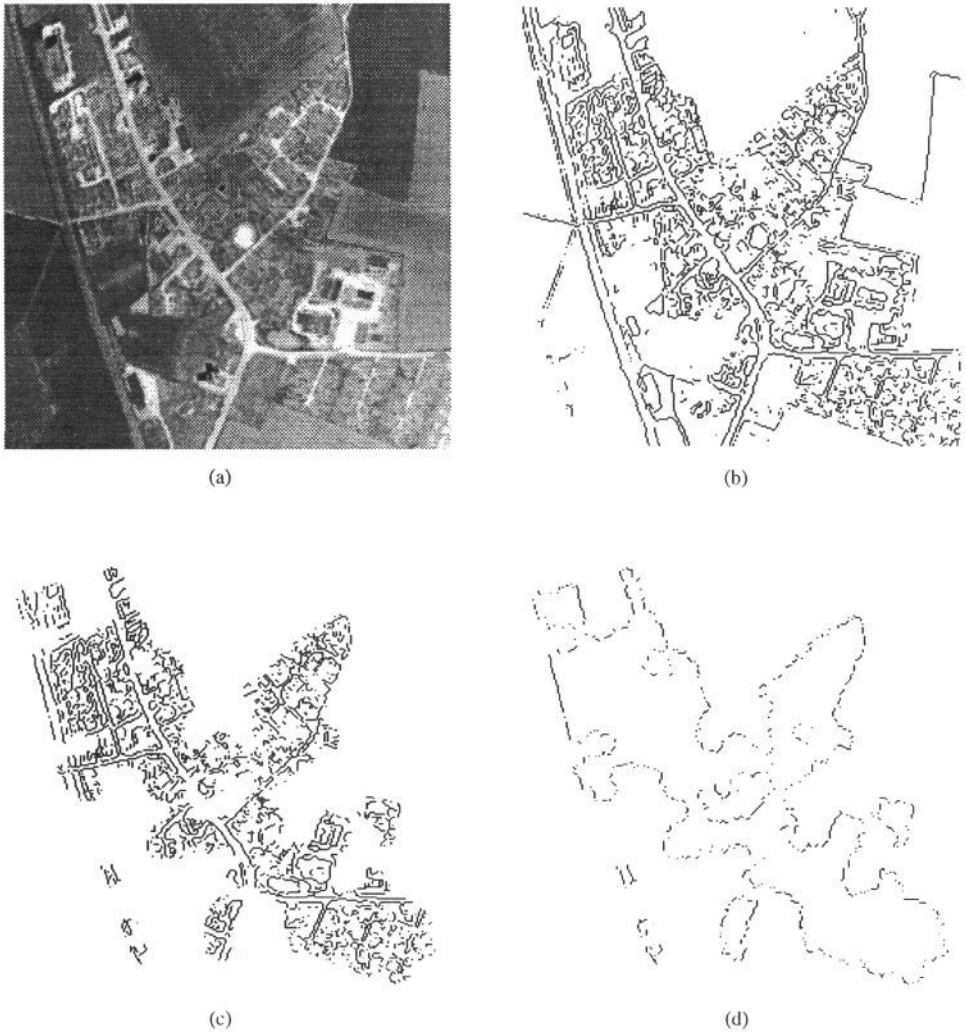
## 4 Results and discussion

The algorithm described in the previous sections was applied to several remotely sensed images. Three examples of such images are presented in figures 4, 5 and 6. Figure 4 shows all the intermediate results of the algorithm. Panel (a) is the original image and panel (b) the output of a Sobel edge detector. Panel (c) shows the boundary edgels after the free angle has been assigned to each edgel in panel (b) and thresholded using as threshold value  $90^\circ$ . The edgels in panel (c) are used for the morphological smoothing of the edge map. Notice that roads and field boundaries are among them. Panel (d) shows the eroded edge map and panel (e) shows the boundaries of this map. If no dilation was performed, the textured regions would be shrunk as can be seen from this output. The eroded and dilated edge map is shown in panel (f) and the final output in panel (g). In figures 5 and 6 the top left panel shows the original input image, while the top right panel is the output of the Sobel edge detector. The bottom left panel is the edge map after the morphological smoothing and the bottom right panel shows the identified boundary edgels.

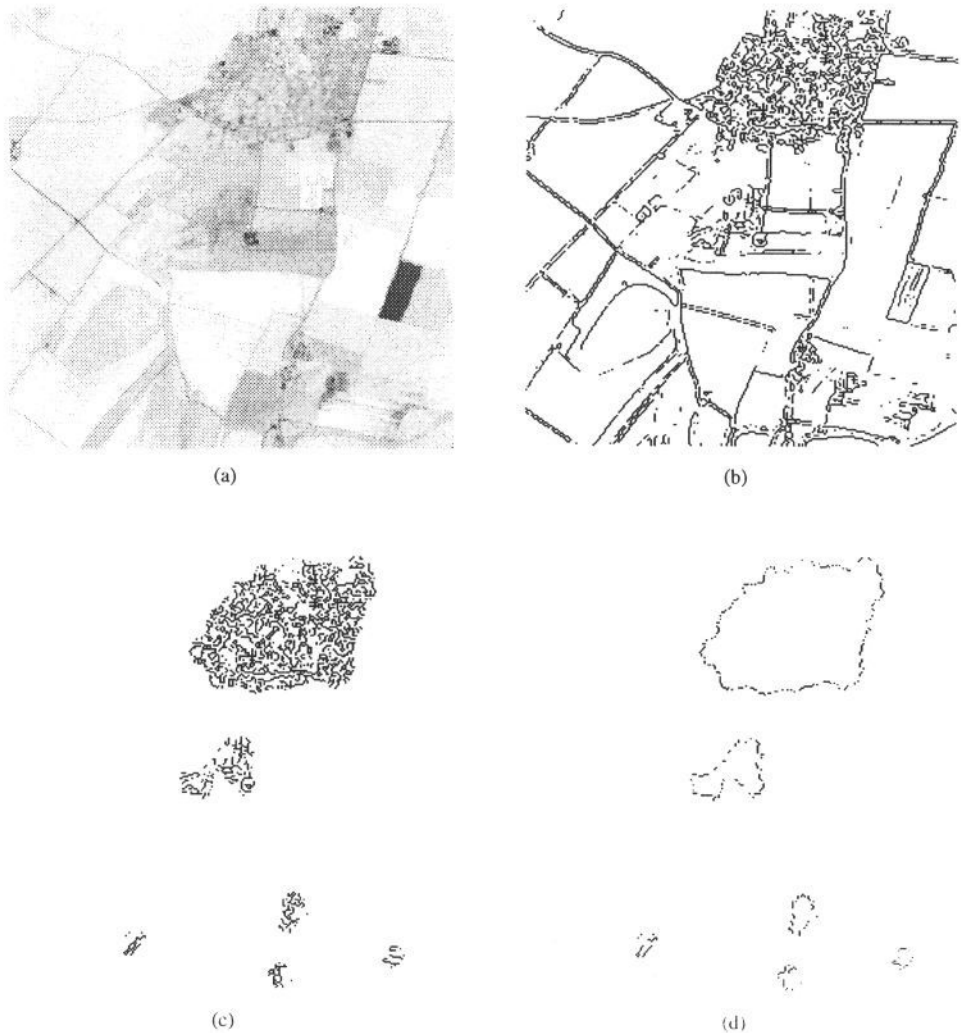
All these results were obtained with the same threshold applied to the free angle, namely  $90^\circ$ . The thresholds for the neighbourhood size were 7, 8 and 9 pixels respectively, and the structuring element used for the morphological smoothing had radius 4, 6 and 7 pixels respectively. The effect of each of these thresholds to the output is demonstrated in figure 7. Symbol  $Dt$  is used to indicate the neighbourhood radius, symbol  $EDt$  is used for the radius of the structuring element, and  $FAt$  is the symbol for the threshold of the free angle. Panels (a) and (b) differ only



**Figure 4:** Input image and outputs of various stages of the algorithm.



**Figure 5:** (a) Original image. (b) The output of a Sobel edge detector. (c) The morphologically smoothed edge map. (d) The boundary edges identified. The thresholds used were: Radius of neighbourhood 8, Size of Structuring element 6, and free angle threshold  $90^\circ$ .



**Figure 6:** (a) Original image. (b) The output of a Sobel edge detector. (c) The morphologically smoothed edge map. (d) The boundary edgels identified. The thresholds used were: Radius of neighbourhood 9, Size of Structuring element 7, and free angle threshold  $90^\circ$ .

in the size of the structuring element which is of size 5 in (a) and size 6.5 in (b). As expected, the smaller structuring element allows smaller textured regions to be identified. Panels (a) and (d) differ only in the threshold of the free angle. We see that as the threshold increases from  $90^\circ$  to  $120^\circ$  more boundary detail is lost. Panel (c) shows that an even smoother boundary and over all smoother result can be obtained if the neighbourhood and the structuring element sizes are increased. Panels (c) and (g) show again the effect of increasing the angle threshold while keeping the other parameters the same. Notice that the effect is not simply to produce smoother boundaries, but to make new smaller regions to appear. This is because the thresholded boundary edgels in the first stage of the algorithm play role in the smoothing process as well. Panel (h) shows the result when the angle threshold has been reduced to  $60^\circ$ . The roughness of the produced output is very evident.

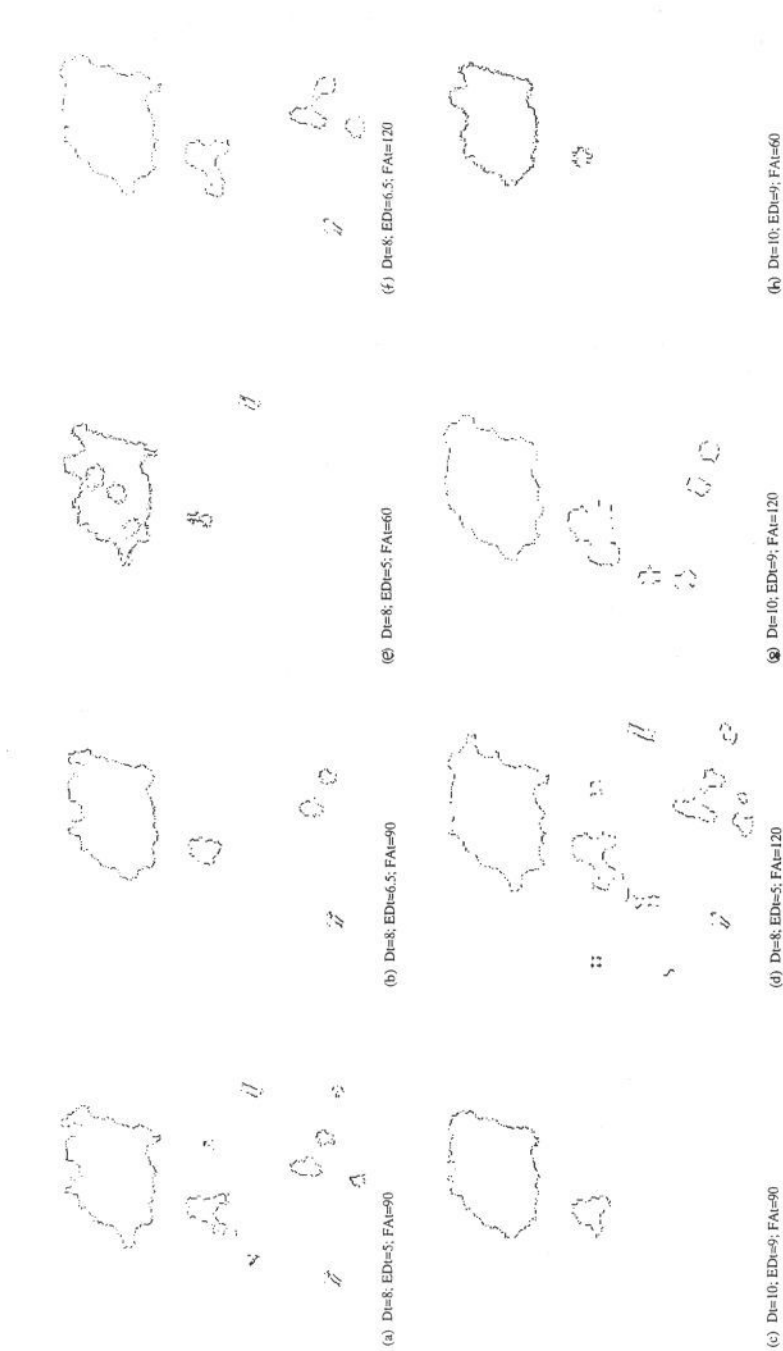
In theory, the tokens which are chosen to create the effect of texture do not need to be just edgels. They could be the local maxima of any filter designed to respond to the repetition of a pattern or any regularity present in the image. The algorithm is designed to identify the boundaries of point sets and these point sets could be texels of any nature derived by any way as long as high density of such tokens implies a textured region. For the images we applied the algorithm to, the use of edgels as tokens of texture is adequate. These are images with 10m resolution like the panchromatic SPOT images. The edgels were obtained with the help of Sobel edge detection which proved satisfactory for these particular resolution. It is important not to use any operators with any higher degree of smoothing because otherwise we shall lose all local intensity variation that creates the perception of texture. The neighbourhood used in the determination of the free angle was circular in the results presented above. However, the use of a city block metric (which will result in a square neighbourhood) will be equally acceptable and will result in a much more efficient algorithm. Finally, if the threshold for the free angle is set to  $180^\circ$  the algorithm can be used to find the convex hull of the point sets.

**Acknowledgments** This work was done under the CEC project 0025 of the Environment Programme.

## References

- [1] Ahuja N. - Dot Pattern Processing Using Vornoi Neighborhoods, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-4, No. 3, May 1982, pp. 336-343.
- [2] Corr D. G., Tailor A. M., Cross A., Hogg D. C., Lawrence D. H., Mason D. C. and Petrou M., 1989, - Progress in automatic analysis of multi-temporal remotely-sensed data, *Int J of Remote Sensing*, Vol 10, pp 1175-1195.
- [3] Devijver P.A. - Selection of Prototypes for Nearest Neighbor Classification, *Proceedings of the Indian Statistical Institute Golden Jubilee International Conference on Advances in Information Sciences and Technology*, Vol. I: Pattern Recognition and Digital Technique, Calcuta 1982, pp. 84-106.





**Figure 7:** The effect of the various thresholds on the output. Dt stands for neighbourhood radius, EDt stands for size of structuring element and FAt stands for free angle threshold.

- [4] Ducksbury P. G. - Parallel Texture Region Segmentation using a Pearls Bayes Network, *BMVC 1993*, Vol 1, pp 187-196.
- [5] Edelsbruner H., Kirkpatrick D., Seidel R. - On the Shape of a Set of Points in the Plane, *IEEE Transactions on Information Theory*, Vol. IT-29, No. 4, Jul. 1983, pp. 551-559.
- [6] Fairfield J. - Contoured Shape Generation: Forms that people see in dot patterns, *Proc. of the IEEE Conference on Systems, Man, and Cybernetics*, 1979, pp. 60-64.
- [7] Gesu V. Di, Maccarone M.C. - Description of Fuzzy Images by Convex Hull Technique, *Proc. of the 8. IEEE International Conference on Pattern Recognition*, Paris, Oct. 1986, pp. 1276-1278.
- [8] Hart P.E. - The Condensed Nearest Neighbors Rule, *IEEE Transactions on Information Theory*, Vol. IT-14, May 1968, pp. 515-516.
- [9] Janković Dj., Matić S., Živojnović V. - On a Method of Finding Contour prototypes for nonparametric classification, *Proc. of the IEEE International Symposium on Information Theory*, Ann Arbour, Oct. 1986.
- [10] Janković D. and Sklansky J. - Finding Contour Points of an N-Dimensional Point Cloud, Technical Report, Department of Electrical Engineering, University of California, Irvine, 1990.
- [11] Jarvis R.A. - Computing the shape hull of points in the plane, *Proc. of the IEEE Computer Society Conference on Pattern Recognition and Image Processing*, Long Beach, 1977, pp. 231-241.
- [12] O'Callaghan J.F. (1974) Computing the Perceptual Boundaries of Dot Patterns, *Computer Graphics and Image Processing*, No.3, 1974, pp. 141-162.
- [13] O'Callaghan J.F. - Human perception of homogeneous dot patterns, *Perception*, Vol. 3, 1974., pp. 33-45.
- [14] Toussaint G. - Pattern Recognition and Geometric Complexity, *Proc. of the 5. IEEE International Conference on Pattern Recognition*, Miami Beach, Florida, Dec. 1980., pp. 1324-1347.
- [15] Tuceryan M., Ahuja N. - Extracting perceptual structure in dot patterns: an integrated approach, *Technical Report IULU-ENG-87-2206*, University of Illinois at Urbana-Champaign, Jan. 1987.
- [16] Zucker S.W., Hummel R.A. - Toward a Low-Level Description of Dot Clusters: Labeling Edge, Interior, and Noise Points, *Computer Graphics and Image Processing*, Vol. 9, 1979, pp. 213-233.