MTNS 93

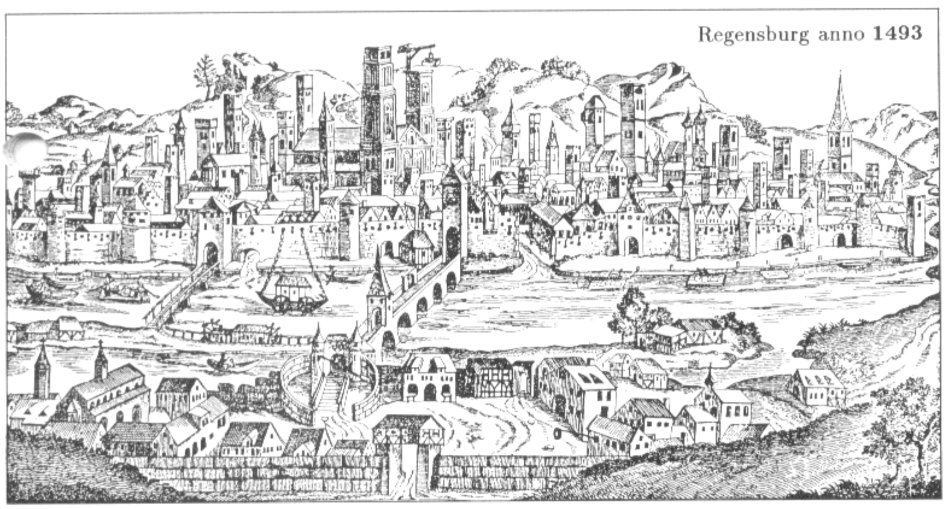
International Symposium on the

Mathematical **T**heory of

Networks and **S**ystems

Regensburg, August 2–6, 1993

Preliminary Program



Regensburg anno **1493**

A SKELETON-BASED HIERARCHICAL SYSTEM FOR LEARNING AND RECOGNITION

Iraklis M. Spiliotis, Dimitris A. Mitzias and Basil G. Mertzios
Department of Electrical Engineering
Democritus University of Thrace, HELLAS

1. INTRODUCTION

Binary images are usually used for the learning and recognition of objects, since only the information of the shape is important for these tasks. The most common binary representation format is a two-dimensional (2-D) array, each element of which has the brightness value of the corresponding pixel. Other approaches are also available aiming to provide machine perception for image pieces greater than a pixel, such as the quadtree representation [1], the chain code representation [2] and the interval coding [3], [8]. In this paper a recently proposed advantageous binary image representation scheme, which represents the image as a set of rectangular areas with object level (blocks) and is called *block representation* [4],[5], is used. Thinning constitutes a basic and significant step in the recognition process in image analysis and computer vision systems. Moreover, the skeleton of an object provides useful information about the shape of the object. In addition, this information may be considered to be independent of the noise and of the width of the object. A thinned pattern is a line drawing representation of a usually elongated pattern. Many approaches for thinning have been proposed in the past. The Medial Axis Transform (MAT) introduced by Blum [6] is a classical way to extract skeleton. Thinning methodologies are described in [7]-[10].

Using a block represented binary image, the fast extraction of the critical points of the object is achieved. The proposed method results to a set of points and all the necessary information concerning the links among them; therefore, a thinned pattern of the given object may be obtained by connecting the critical points. These critical points of the object are necessary for the structural description of the object, since a set of subpatterns is formed from these points.

The extracted subpatterns are classified, using simple geometrical and statistical features. The possible classes are the straight line segment class, the half of a circle class and the circle class. The use of simple features and the small amount of information required for the subpatterns representation, leads to the fast and accurate classification of the subpatterns. Care has been taken to handle subpatterns that are not classified in one of the above classes. In such a case, a separate procedure forms two new subpatterns, from subpattern, which has not been classified.

The information concerning the shape and the connections of the subpatterns, are used for the final classification of the object in a final level. The structural description of the object is obtained by the classifier and is used for object learning and/or recognition. The system has an hierarchical structure and consists of three levels: (i) thinning, critical points extraction and subpattterns formulation, (ii) classification of the subpatterns and (iii) final classification. Section 2 presents the block representation concept and the associated algorithm. Section 3 presents the skeletonization process. In Section 4 the subpattern classifier is described and in Section 5 the final classifier is presented.

2. BLOCK REPRESENTATION

In a digital binary image there are rectangular areas of object value 1 with edges parallel to the image axes. At the extreme case one pixel is the minimum rectangular area of the image. These rectangulars are called *blocks* . Consider a set that contains as members all the nonoverlapping blocks of a specific binary image, in such a way that no other block can be extracted from the image (or equivalently each pixel with object level belongs to only one block). This set represents the image without loss of information. It is always feasible to represent a binary image with such a set, of all the nonoverlapping blocks with object level. We call this representation of the binary image, *block representation*. According to the above discussion, two useful definitions concerning block representation are formulated [4],[5]:

Definition 1: *Block* is called a rectangular area of the image, with edges parallel to the image axes, that contains pixels of the same value.

Definition 2: A binary image is called *block represented,* if it is represented as a set of blocks with object level, and each pixel of the image with object value belongs to one and only one block.

The block representation concept leads to a simple and fast algorithm, which requires just one pass of the image and simple bookkeeping process. Consider a binary image $f(x,y)$, $x=0,1,..,N_1-1$, $y=0,1,..,N_2-1$. The block extraction process requires a pass from each line y of the image. In this pass all object level intervals are extracted and compared with the previous extracted blocks.

In the following, a uniquely determined block representation algorithm, which is based on the row by row processing, is given.

Block representation algorithm

Step 1: For each line y of the image f,

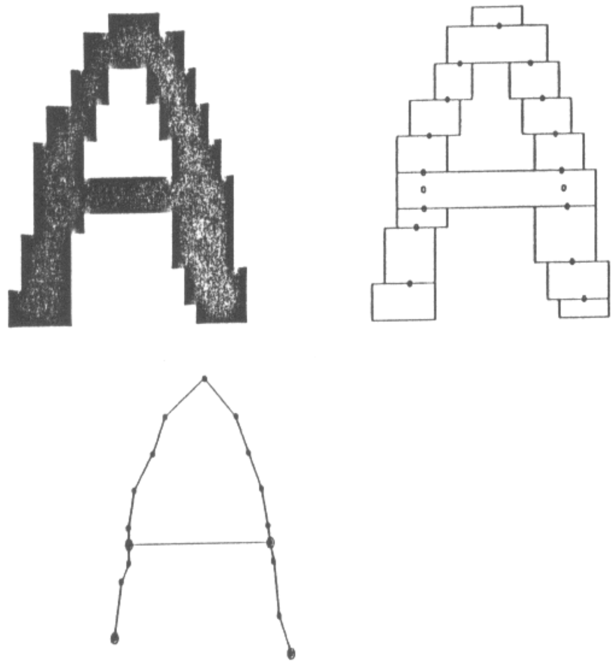
Step 2: Find the object level intervals in line y

Step 3: Compare intervals and blocks that have pixels in line y-1,

Step 4: If an interval does not matches with any block, this is the beginning of a new block.

Step 5: If a block matches with an interval, the end of the block is in the line y.

As a result of the application of the above algorithm, we obtain a set of all the rectangular areas with level 1, that form the object. The block extraction process is implemented easily with low computational complexity, since it is a pixel checking process without numerical operations. The block representation may be seen as a physical way for the representation of binary images. Each block is represented with four integers, the coordinates of the upper left and down right corner in vertical and horizontal axes.



(a) (b) (c)

Figure 1. (a). The image of the character A, (b). The extracted blocks and the critical points and (c). The thinned object and the subpatterns.

3. THINNING AND CRITICAL POINTS

A block represented binary image is stored as a set of the blocks that constitute the image. In such an image it is easy to find the connections among the blocks. Two blocks are said to be connected if their projections in x and y axes are covered or are neighbors.

For each pair of connected blocks one *junction* point (the central point of the common line segment of two connected blocks) is extracted. The junction point belongs to both the connected blocks. For each block we check if the distance among its junction points and the extremities (the central points of the edges of the small dimension) of the block exceeds a threshold value. In such a case, in the corresponding extremity of the block, an *endpoint* of the skeleton is extracted. This endpoint belongs to the block. The threshold value usually is defined as a percentage of the corresponding edge length of the image object.

In the case where a block has more than two junction points and endpoints, one central junction point is selected as *treepoint*. If such a central junction point does not exist, the centroid of the block is selected as the treepoint of the block. The endpoints, treepoints and junction points constitute the *critical* points of the block.

The graphical link of the critical points belonging to the same block, forms the image of the skeleton of the object, since each junction point belongs to two connected blocks. In the case where a treepoint exists in a block, each other critical point of the block is linked only with the treepoint. This method is fast and is in accordance with human perception about skeleton. Figure 1 demonstrates an image of the character A, the extracted blocks, the critical points, the subpatterns and the thinned character.

4. CLASSIFICATION OF THE SUBPATTERNS

A subpattern is a sequence of points in R^2. The second level task is the classification of the subpatterns in a specified number of the classes. These classes are defined as:

a). L, for a straight line segment
b). S, for a semicircle
c). C, for a circle

We use only these three classes in order to simplify the problem. Each subpattern is classified as one or more of the three basic classes. The classification of each subpattern is based on simple geometrical features. At first, the equation and the length (L_s) of the *Basic Hypothetical Line* (BHL), which passes from the first P_0 and the last P_{N-1} point of the subpattern is formed. Now the following hold:

a) If the deviation (according to the length L_s of the BHL) of the intermediate points P_i, $i = 1,2,...,N-2$ is small, then the subpattern is classified as class-L.
b) If the deviation of the points from the hypothetical semicircle, which is defined as half of the circle having as centre the centroid of the BHL and its diameter equals to the length L_s is small, then the subpattern is classified as class-S.
c) If the first and last points of the BHL are the same ($L_s=0$), then the centroid defined by the points P_i, $i = 0,1,...,N-2$ is calculated. If the distances from the points P_i to the centroid are in the same range, then the subpattern is classified as class-C.
d) In every other case, where a subpattern is not classified to any of the above three classes, then the algorithm searches for a suitable point which divides the subpattern to two new subpatterns. The separation may result to subpatterns which belong to any of the desired classes (L,S,C).

All the above classification procedure is fast implemented by the use of a look-up table. In every step just one feature is examined. In the case where a subpattern is classified as class-L, then the other features are not calculated. Therefore, the feature extraction process is characterized by low computational complexity, since is controlled by the classifier and only the necessary features are calculated.

5. THE FINAL CLASSIFIER

The information extracted from an object concerns the type and the number of the subpatterns (NT[Sb]) and the number (N[TEP]) and the position (P[TEP]) of the tree-end-points. The final classifier is based on the successive reduction of the complexity of the problem. The schematic diagram of the final classifier is shown in Fig. 2.

In a first step (C-1), the input object is compared only with the object classes having the same number and type of subpatterns. This leads to the rejection of a great number of classes. Following this approach, only a small part (S^i) of the initial classes is allowed to be competitive for the classification of the input object.

In a second step (C-2), the classifier rejects from the remained classes (S^i) those that have different number of treepoints or endpoints. This leads to a part (S^i_j) of the input classes (S^i).

In the final step (C-3) a matching-like process is implemented. The treepoints and the endpoints are positioned on a specified grid of dimension (gxq) over the image of the object. The choice of the magnitude of the grid (length g and q) is dependent on the considered application. Specifically, grid must satisfy the requirement of local generalisation, in order to avoid the small variations due to the noise and to reduce the complexity of the system. On the other hand, we should avoid very large grids, which result to low discrimination efficiency.

If an examined object is not classified to any of the object classes, then this object may be considered as a new class, which can be stored in the classifier. Since the final classifier is very flexible, the new class can be stored in any of the above three steps of the final classification procedure as it is seen in Fig. 2.

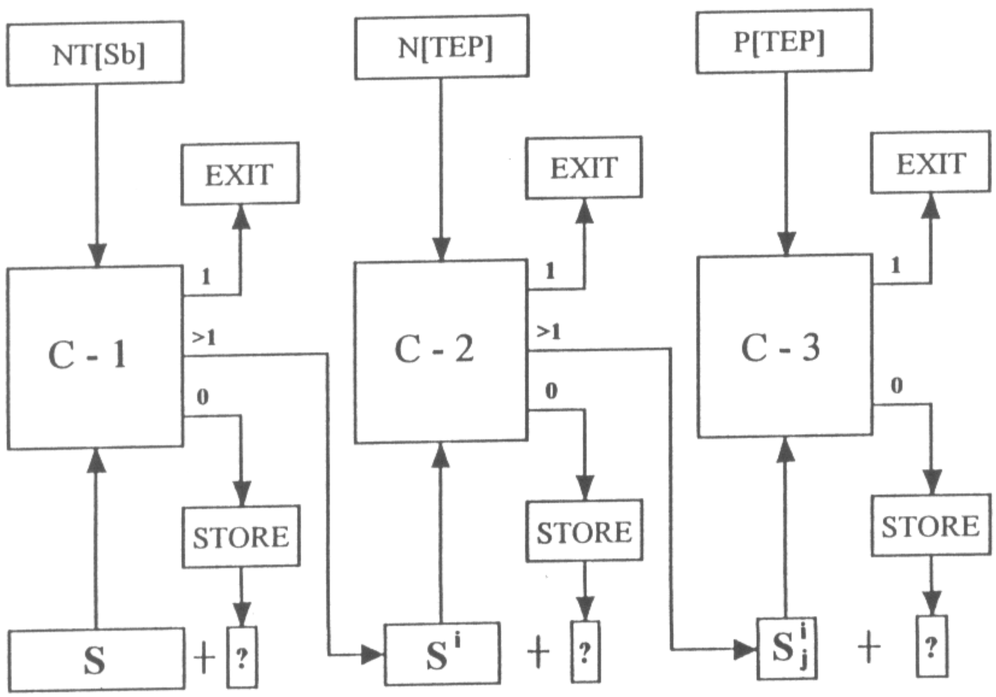


Figure 2. The schematic diagram of the final classifier

REFERENCES

[1] H. Samet, "The quadtree and related hierarchical data structures", *Computing Survey*, vol. 16, No. 2, pp. 187-260, 1984.

[2] H. Freeman, "Computer processing of line drawings", *ACM Computing Surveys*, vol. 6, pp. 57-97, 1974.

[3] J. Piper, "Efficient implementation of skeletonisation using interval coding", *Pattern Recognition Letters*, vol. 3, pp. 389-397, 1985.

[4] I. M. Spiliotis and B.G. Mertzios, "Real-time computation of statistical moments on binary images using block representation for learning systems", *4th International Workshop on Time-Varying Image Processing and Moving Object Recognition*, Florence, Italy, June 10-11, 1993.

[5] I. M. Spiliotis and B.G. Mertzios, "Fast computation of two-dimensional moments on binary images using block representation", submitted to *IEEE Trans. on Pattern Anal., Machine Intel.*

[6] H. Blum, "A transformation for extracting new descriptors on shape", *Symposium on Models for Perception of Speech and Visual Form*, Weiant Whaten-Dunn, Ed., MIT Press, Cambridge, Mass., 1967.

[7] C.J. Hilditch, "Linear skeletons from square cupboards", *Machine Intel.*, vol. 4, pp. 403-420, 1969.

[8] L. Ji and J. Piper, "Fast homotopy - preserving skeletons using mathematical morphology", *IEEE Trans. on Pattern Anal., Machine Intel.*, vol. 14, No. 6, June 1992.

[9] R. M. K. Sinha, "A width independent algorithm for character skeleton estimation", *Computer Vision, Graphics and Image Proc.*, vol. 40, pp. 388-397, 1987.

[10] L. Lam, S. W. Lee and C. Y. Suen, "Thinning methodologies - a comprehensive survey", *IEEE Trans. on Pattern Anal., Machine Intel.*, vol. 14, No. 8, pp. 869-885, 1992.

[11] K. S. Fu, *Syntactic Pattern Recognition and Applications*, Englewood Cliffs, NJ: Prentice Hall, 1982.