13th INTERNATIONAL CONFERENCE ON DIGITAL SIGNAL PROCESSING

July 2-4, 1997, Santorini, Greece

DSP97

General Chair
A. G. Constantinides
Imperial College, UK
a.constantinides@ic.ac.uk

Co-Chairs
V. Cappellini
University of Florence, Italy
cappellini@ingfi1.ing.unifi.it

T. Deliyannis
University of Patras, Greece
deliyan@physics.upatras.gr

Technical Program
A. N. Skodras
University of Patras, Greece
skodras@cti.gr

Special Sessions
T. Stathaki
Imperial College, UK
tania@ic.ac.uk

Plenary Sessions
T. Stouraitis
University of Patras, Greece
thanos@ee.upatras.gr

Local Arrangements
I. Haritantis
Univers[illegible] of Patras, Greece

Exhibits
E. Zygouris
University of Patras, Greece

Publications
S. Fotopoulos
University of Patras, Greece

Publicity
V. Anastassopoulos
University of Patras, Greece

Social Program
G. Economou
University of Patras, Greece

Int. Tech. Program Committee
A. Antoniou, Canada
C. [illegible] Burrus, USA
R. [illegible]larke, UK
J. [illegible]elis, Belgium
C. [illegible] Cowan, UK
T. S[illegible]tani, UK
T. Eb[illegible]imi, Switzerland
M. Ghanbari, UK
H. Gharavi, UK
f harris, USA
T. S. Huang, USA
A. K. Katsaggelos, USA
S. Kollias, Greece
S. Y. Kung, USA
B. G. Mertzios, Greece
S. K. Mitra, USA
Y. Neuvo, Finland
A. Paulraj, USA
M. Petrou, UK
I. Pitas, Greece
T. Reed, USA
H. W. Schuessler, Germany
A. Spanias, USA
S. Theodoridis, Greece
L. Torres, Spain
A. Venetsanopoulos, Canada
C. Xydeas, UK

IEEE — University of Patras — EURASIP

Call For Papers

The 13th *International Conference on Digital Signal Processing (DSP97)*, the longest in existence Conference in the area of DSP, organized in cooperation with the IEEE SP society and EURASIP, will be held on the island of Santorini, Greece, July 2-4, 1997. It belongs to a series of events which started from London in 1968 every two years and continued in Florence every three years. The last two successful meetings took place in Cyprus in 1993 and 1995. It is the first time that this prominent scientific conference is held in Greece.

The successful format of the earlier events will be followed again. Namely, a significant number of Special Sessions organised by internationally recognised experts in the areas will be held. The role of the Session Organiser is of vital significance in the sense that she or he will be entirely responsible for the invitation of 8-10 speakers, for the quality of the contributions and the general theme that the session will take. Each of these Special Sessions will be a little like a mini-Symposium and the Session Organiser will be the *Symposiarch*. Besides, the program includes the presentation of new results in lectures and posters as well as plenary sessions by eminent scientists. Lecture and poster sessions will be treated equally in terms of the review process.

Topics of Interest
- Adaptive Signal Processing
- Array Signal Processing
- Biomedical Signal and Image Processing
- Blind Equalisation
- Computer Vision
- Image and Multidimensional Signal Processing
- Motion Detection and Estimation
- Noise Reduction / Cancellation
- Nonlinear Signals and Systems, Chaos and Fractals
- Nonuniform Sampling and Multiresolution Signal Processing
- Signal and Image Processing Algorithms / Architectures / Implementations
- Signal and System Modelling
- Speech Processing
- Theory & Applications of Transforms and Time-Frequency Representation

Special Sessions
- Adaptive Filtering and Analysis
- Digital Filter Theory and Design
- Digital Signal Processing in Mobile Communications
- Higher Order Statistical Signal Processing
- Image and Scene Analysis
- Multimedia Signal Processing
- Neural Networks and Fuzzy Logic in Signal Processing
- Nonlinear and Morphological Signal and Image Processing
- Orthogonal Transforms: Algorithms and Applications
- Speech Coding
- Stereoscopic and 3D Imaging
- Very Low Bit Rate Coding
- Wavelets and Subband Signal Processing and Coding

Plenary Sessions
- Signal Processing in Mobile Communications Systems
- Wavelets and Signal Processing
- Perspectives in Video Coding

Authors are invited to submit one original and two copies of the **camera-ready paper**. The paper must be completed in 4 pages, including figures, tables and references, and should be written in English. Faxed submissions are not acceptable. Papers should be typed on standard A4 paper with 2cm margins on all four sides, in two columns format of 8cm width each, single spaced, in Times or similar type style of 10 points, and should be printed on one side of the paper only. Centred at the top of the first page across both columns should be the complete title, author(s) name(s), affiliation, mailing address and Email. This is followed by a blank space and then the abstract, up to 10 lines, followed by the 2-column text. Authors should indicate one or two of the above categories that best describe the topic of the paper, as well as their preference (if any) regarding lecture or poster sessions. The program committee will make every effort to satisfy these preferences. Submitted papers will be reviewed by two referees and all accepted papers will be published in the Conference Proceedings.

Address for paper submissions and all non technical correspondence:

	DSP97 Conference Secretariat	Tel.:	+30 61 997 463
	Electronics Laboratory	Fax:	+30 61 997 456 / 991 980
	University of Patras, GR-26110 Patras	Email:	dsp97@cti.gr
	Greece	URL:	http://ellab.physics.upatras.gr/r_d/dsp97/
Authors Deadlines:	Submission of camera-ready papers to be received by		**March 1, 1997.**
	Notification of acceptance will be mailed out by		**April 1, 1997.**

In addition to the technical program, a great social program will be offered to the participants and their companions. It will be a unique opportunity to meet friends, while enjoying the beautiful summer weather of the Aegean.

A Fast Skeleton Algorithm on Block Represented Binary Images

Iraklis M. Spiliotis and Basil G. Mertzios
Automatic Control Systems Laboratory
Department of Electrical and Computer Engineering
Democritus University of Thrace
Xanthi 67100
HELLAS

Fax: +30-541-26473
e-mail: spiliot@demokritos.cc.duth.gr
mertzios@demokritos.cc.duth.gr

Abstract

This paper describes a binary image representation scheme, which is called Image Block Representation and presents a new skeletonization algorithm, which is fast implemented on block represented binary images. The main purpose of the Image Block Representation is to provide an efficient binary image representation that permits the execution of operations on image areas instead of image points. The skeletonization algorithm operates in four subiterations: each subiteration deletes the north, the south, the west and the east boundary points, respectively. Due to the substitution of the boundary points by the block's boundary points the relevant operations are performed fast, while preserving the end points and the object connectivity.

Keywords: Image Block Representation, Skeletonization, Thinning.

I. INTRODUCTION

The most common image representation format is the two-dimensional (2-D) array. However, many research efforts for deriving alternative image representations have been motivated by the need of fast processing of huge amount of data. Such image representation approaches aim to provide machine perception of images in pieces larger than a pixel and are separated in two categories: boundary based methods and region based methods and include quadtree representations [1], chain code representations [2], contour control point models [3], autoregressive models [4], run length encoding [5],[6] and interval coding representation [7]. A region based method, which is called image block representation has been presented [8]-[12].

This paper presents a skeletonization algorithm, which is characterized by low computational cost and it is suitable for fast processing rates, due to the substitution of image pixels from blocks. The algorithm operates in four subiterations: each subiteration deletes the north, the south, the west and the east boundary points, respectively. Due to the substitution of the boundary points by the block's boundary points the relevant operations are performed fast.

II. IMAGE BLOCK REPRESENTATION

A bilevel digital image is represented by a binary 2-D array. Without loss of generality, we suppose that the object pixels are assigned to level 1 and the background pixels to level 0. Due to this kind of representation, there are rectangular areas of object value 1, in each image. These rectangular areas, which are called *blocks*, have their edges parallel to the image axes and contain an integer number of image pixels. At the extreme case, one pixel is the minimum rectangular area of the image.

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 a set of all the nonoverlapping blocks with object level. We call this representation of the binary image, *Image Block Representation (IBR)*.

The block representation concept leads to a simple and fast algorithm, which requires just one pass of the image and simple bookkeeping process. In fact, considering a $N_1 x N_2$ 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.

A block represented binary image $f(x,y)$ is comprised of a set of nonoverlapping blocks that completely cover the image areas with object level and it is denoted as:

$$f(x,y) = \{b_i : i = 0,1,\ldots,k-1\} \quad (1)$$

where k is the number of the blocks. Each block is described by the coordinates of two corner points, i.e.:

$$b_i = (x_{1,b_i}, x_{2,b_i}, y_{1,b_i}, y_{2,b_i}) \quad (2)$$

where for simplicity it is assumed that: $x_{1,b_i} \leq x_{2,b_i}$ and $y_{1,b_i} \leq y_{2,b_i}$. In Fig. 1, the blocks that represent an image of the character d are illustrated.

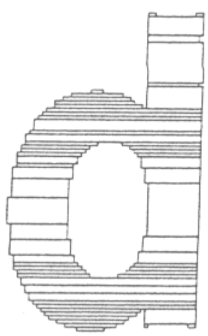

Figure 1. Image of the character d and the blocks.

In many operations it is important to have information concerning not only the location of the blocks but also information concerning the neighbor and connected blocks. The following definition provides a template for block connectivity:

Definition

Two blocks are defined as connected, if their projections on both the x or y axis are overlapped or the are neighbors.

■

The information concerning block connectivity requires a suitable data structure for storage. Therefore, each block b_i is represented as the ordering:

$$b_i = (x_{1,b_i}, x_{2,b_i}, y_{1,b_i}, y_{2,b_i}, nc_i, \mathrm{c}_i) \quad (3)$$

where x_{1,b_i}, x_{2,b_i} are the coordinates of the i-th block according to the horizontal axis, y_{1,b_i}, y_{2,b_i} are the coordinates of the block according to the vertical axis, nc_i is the number of the connected blocks and c_i is a list with the indexes of these connected blocks.

The image block representation is reduced to the run length encoding [5],[6] of binary images at the extreme cases, where each block is comprised of pixels belonging in only one row of the image. Such a case is that of a chessboard image, where the transitions from white to black have 1 pixel length and the number of the blocks is $N^2/2$, i.e. it is exactly equal to the object level run lengths. However, in most practical situations, the image block representation is superior to the run length encoding, since the number of the blocks is significantly smaller than the number of the run lengths. In Fig. 2 four test images are illustrated, while in Table 1 the number of the pixels with object level, the number of the rows with object pixels, the number of the blocks extracted from these images (using the Algorithm 1) and the required storage space for both the 2-D rerpesented and the block represented images are shown. It can be seen that the number of the blocks generated by the Algorithm 1, is significantly less than the number of the rows with black pixels. In the worst case of the island Mikonos image (of Fig. 2 (b)), where the number of the rows with object pixels is 249 and the number of the blocks is 232, it should be noted that the number of the gulfs and peninsulas of the island is significantly large and therefore the number of the blocks is respectively large.

III. FAST SKELETONIZATION ALGORITHM

The algorithm uses the criteria specified by the well-known skeletonisation algorithm of Zhang and Suen [13]. The proposed algorithm is implemented iteratively; each iteration is divided into four subiterations. In the first, second, third and fourth subiteration, the north, west, south and east pixels of the object, are removed respectively. The main advantage of the proposed algorithm is that operates in blocks and therefore permits the deletion of areas of pixels, instead of a single pixel.

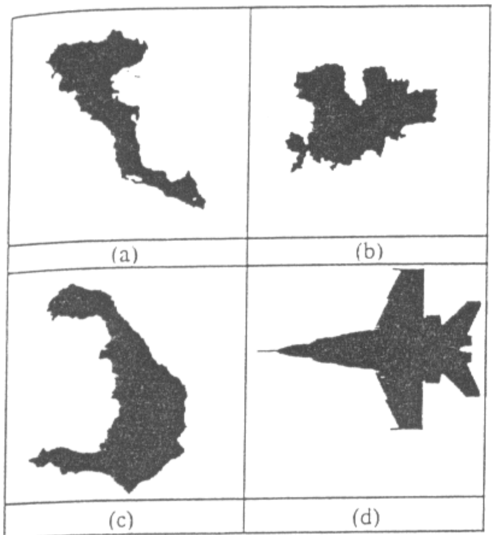

Figure 2. A set of test images. (a). Image of the island Corfu of 512x512 pixels. (b) Image of the island Mikonos of 512x512 pixels. (c) Image of the island Santorini of 512x512 pixels. (d) Aircraft image of 512x697 pixels.

Table 1. The number of the pixels with object level, the number of the rows with object pixels, the number of the blocks, the required storage for the 2-D images and the required storage for the block represented images for the set of the test images of Figure 2.

Image	Pixels with object level	Rows with object pixels	Number of blocks	Storage for the 2-D image	Storage for blocks
Corfu	41605	411	250	32768 bytes	2000 bytes
Mikonos	47368	249	232	32768 bytes	1856 bytes
Santorini	63203	474	257	32768 bytes	2056 bytes
Aircraft	118831	494	397	44608 bytes	3176 bytes

A. North subiteration

At first the neighboring blocks of the considered block are determined, sorted and placed in two lists; one for the upper neighbors and one for the lower neighbors. Different procedures are applied for those blocks that have unity width than those with greater width.

A.1. Blocks with unity width

For the blocks with unity width, the blocks that remain after the subiteration are described as:

$$R = N \bigcup \bar{S} \bigcup A \quad (4)$$

where N are the areas that have a north neighbor, S are those areas that have a south neighbor and A are those areas that have transitions from 01 patterns different from 1. If b is the considered block and a_i, $i=1,\ldots,k$ are the north neighbor blocks of b, then

$$N=(\max(b_{x1},a_{i,x1}),\ \min(b_{x2},a_{i,x2})),\ i=1,\ldots,k. \quad (5)$$

S is computed in a similar manner, and $\bar{S}$ is computed using logic operations in blocks, as described in [11], [12].

The algorithm counts the transitions for the two extreme pixels of the considered block. For the middle pixels of the block, the algorithm recognizes the areas A as those that have at least one north and at least one south neighbor pixel.

A.2. Blocks with width greater than 1

For the blocks with width greater than 1, the algorithm determines the remaining areas as those that have a north neighbor block. Care should be taken only for the two extreme pixels, the upper left and the upper right of the block, where all the criteria of Zhang and Suen are applied.

The procedure is quite similar for the south subiteration.

B. East subiteration

At first, the neighboring blocks of the considered block are determined, sorted and placed in two lists; one for the upper neighbors and one for the lower neighbors.

B.1. Blocks with unity width

In the case of a block with unity width, all the criteria of Zhang and Suen [13], are taken into account.

B.2. Blocks with width greater than 1

In the case of a block with width greater than 1, the algorithm deletes all the middle points of the left side of the block. For the two extreme pixels, the upper left and the lower left of the block, the criteria of Zhang and Suen are also applied.

The procedure is quite similar for the west subiteration.

The algorithm is fast implemented, since it operates on image areas instead on single pixels. Table 2 demonstrates the required computational times for the execution of the skeletonization for the images of Fig. 2, using the Zhang and Suen algorithm and the proposed algorithm that operates in blocks.

Table 2. The required computational times in seconds, for the execution of the skeletonization operation using the Zhang and Suen algorithm and the proposed algorithm. The third column is the reduction factor.

Image	Zhang - Suen	Blocks	Reduction
Corfu	12.1	2.9	4.1
Mikonos	14.6	2.8	5.2
Santorini	12.6	3.1	4.1
aircraft	18.4	4.7	3.9

CONCLUSIONS

In the recent years other of fast image processing and analysis algorithms that operate on block represented binary images have been presented in the literature: specifically the real-time computation of the statistical moments (both software [10] and hardware algorithms [14]), the fast implementation of image shift, image scale, image rotation, determination of the minimum and of the maximum distance from a point to an object, perimeter measurement, area measurement, logic operations, connectivity checking, object detection and edge extraction [12].

The proposed skeletonization algorithm may be considered as a parallel skeletonization algorithm that operates in a serial machine. This conclusion arises from the fact that the algorithm decides which are the pixels that should form a new block, or equivalently deletes simultameously a number of boundary pixels for which the conditions of removing are valid.

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] D.W. Paglieroni and A.K. Jain, "Control point transfroms for shape representation and measurement", *Computer Vision, Graphics and Image Processing*, vol. 42, pp. 87-111, 1988.

[4] R.L. Kashyap and R. Chellappa, "Stochastic models for closed boundary analysis: Representation and reconstruction", *IEEE Trans. Information Theory*, vol. IT-27, no. 5, pp. 627-637, 1981.

[5] J. Capon, "A propabilistic model for run-length coding of pictures", *IRE Trans. Information Theory*, vol. IT-5, no. 4, pp. 157-163, 1959.

[6] W.K. Pratt, *Digital Image Processing*, John Wiley & Sons, 2nd Edition, 1991.

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

[8] I.M. Spiliotis and B.G. Mertzios, "Real-time computation of statistical moments on binary images using block representation", *Proceedings of the 4th International Workshop on Time-Varying Image Processing and Moving Object Recognition*, Florence, Italy, June 10-11, pp. 27-34, 1993.

[9] I.M. Spiliotis, D.A. Mitzias and B.G. Mertzios, "A skeleton-based hierarchical system for learning and recognition", *Proceedings of MTNS 93, International Symposium on the Mathematical Theory of Networks and Systems*, Regensburg, Germany, August 2-6, pp. 873-878, 1993.

[10] I.M. Spiliotis and B.G. Mertzios, "Real-time computation of two-dimensional moments on binary images using image block representation", *IEEE Transactions on Image Processing*. Accepted for publication.

[11] I.M. Spiliotis and B.G. Mertzios, "Logic operations on 3-dimensional volumes using block representation", *Proceedings of the International Workshop on Stereoscopic and Three Dimensional Imaging (IWS3DI'95)*, pp. 317-322, 6-8 September 1995, Fera, Santorini, Greece.

[12] I.M. Spiliotis and B.G. Mertzios, "Fast algorithms for basic processing and analysis operations on block representated binary images". *Pattern Recognition Letters*. To appear.

[13] T.Y. Zhang and C.Y. Suen, "A fast parallel algorithm for thinning digital patterns". *Commun. ACM*, vol. 27, No. 3, March 1984, pp. 236-239.

[14] I.M. Spiliotis and B.G. Mertzios, "Real-time Computation of 2-D Moments on Block Represented Binary Images on the Scan Line Array Processor", *Proc. of VIII European Signal Processing Conference (EUSIPCO-96)*, September 10-13, 1996, Trieste, Italy.