Applying Fast Segmentation Techniques at a Binary Image Represented by a Set of Non-Overlapping Blocks

B. Gatos [1] and N. Papamarkos [2]

[1] Department of Digital Technologies, Lambrakis Press Archives,
8 Heyden Str,104 34 Athens, Greece
bgatos@lpa.gr

[2] Department of Electrical and Computer Engineering,
Democritus University of Thrace,
67 100 Xanthi, Greece
papamark@ee.duth.gr

Abstract

Run Length Smoothing Algorithm (RLSA) and projection profiles are among the fundamental algorithms in binary image processing, mainly used for segmentation of monochrome images. In this paper, fast RLSA and projection profiles are applied to binary images represented by a set of non-overlapping rectangular blocks. The representation of binary images using rectangular blocks as primitives has been used with great success for several image processing tasks, such as image compression, Hough transform fast implementation and skeletonization. We show that this representation can be applied with great success for fast RLSA application and fast projection profiles evaluation. The experimental results demonstrate that starting from a block represented binary image we can apply RLSA and evaluate projection profiles in significant less CPU time. The average time gain is recorded at 60% and 88%, respectively.

1 Introduction

Digital image representation by a set of simple geometrical shapes has been proved useful for effective image segmentation [1] and fast implementation of many image processing algorithms [2-8]. For example, in order to speed up moment calculation, the images can be described in terms of polygonal shapes [2] or, alternatively, can be represented by recursive morphological operations [3]. Also, for the acceleration of Hough transform applications or for applying fast morphological operations, the binary images can be decomposed by using rectangular blocks of foreground pixels as primitives [4,5]. Furthermore, this representation has been applied with great success for segmentation [6], image compression [7] and skeletonization [8].

RLSA and projection profiles are among the fundamental algorithms in binary image processing, mainly used at the stage of segmentation of monochrome images. RLSA is based on examining the white runs existing in the horizontal and vertical directions. For each direction, white runs whose lengths are smaller than a threshold smoothing value are eliminated. The final image is extracted after applying an AND operation at the horizontal and vertical smoothed images. RLSA is usually followed by connected component analysis and used for image segmentation [9], especially for document page layout analysis [14]. Another important application of RLSA is feature extraction for object recognition [10].

Projection profiles are based on image profiling in various directions [11]. By calculating the local minima of horizontal and vertical projections we can define several segments into which the image can be divided [12]. Projection profiles are also used for document skew detection [13]. In this approach, the skew angle of a document corresponds to a rotation for which the mean square deviation of a projection histogram is maximized.

In this paper, we describe how binary image block representation can be useful for speeding up the implementation of the RLSA as well as the horizontal and vertical projection profiles. Since both processes are mainly used for image segmentation, we introduce a fast segmentation for block represented binary images. The horizontal and vertical successive background pixels that construct white runs of short length border on the rectangular blocks that compose the image. Since the

0-7695-1263-1/01/$10.00

starting points of all horizontal and vertical white stripes are known in advance, we achieve a significant acceleration of RLSA implementation. The decomposition of images into non-overlapping blocks offers fast evaluation of the image projection profiles by simply adding the areas of all rectangular blocks projected horizontally or vertically. Experimental results prove the effectiveness of the proposed techniques.

2 Binary Image Block Decomposition

In order to decompose a binary image to a set of non-overlapping rectangular blocks, we proceed to the following algorithm which involves one top-down scan of the images. According to this approach, an image is scanned in a top-down direction until the first foreground pixel (x,y) is found. Then, a procedure is applied that search and constructs the "best fitting block" at (x,y), which is the block with the largest area that has the (x,y) pixel as its upper left corner. All pixels of the "best fitting block" are transformed to background pixels and the procedure is repeated until the entire image is scanned. An example of binary image block decomposition is demonstrated in Fig. 1.

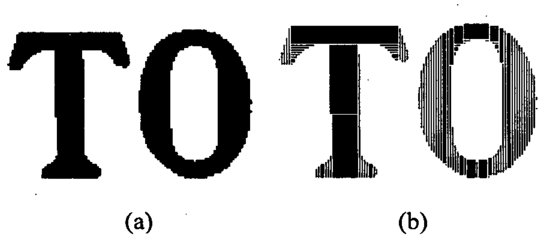


Fig. 1. Block decomposition of a binary image

3 Block Run Length Smoothing Algorithm

The RLSA for a point represented binary image is implemented by applying a left-right (for horizontal smoothing) or top-down (for vertical smoothing) scan of the image and then by selecting all white runs (successive white pixels) with length less than a given threshold. These pixels are converted into foreground pixels (see Fig 2). As we can observe in Fig. 2c, we must pass from all possible white and black runs of the image before selecting the white runs of short length. After applying horizontal and vertical smoothing, the final image is derived by applying an AND operation of those two resulting smoothed images (see Fig. 3).

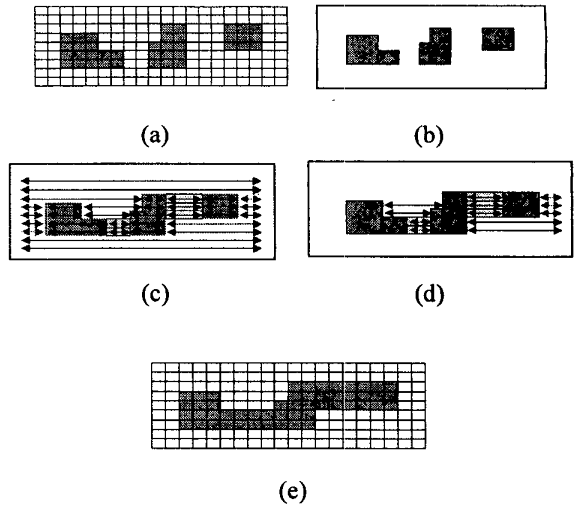


Fig. 2. Horizontal RLSA description. (a) original image, (b) block decomposition, (c) Point RLSA, (d) Block RLSA, (e) final image.

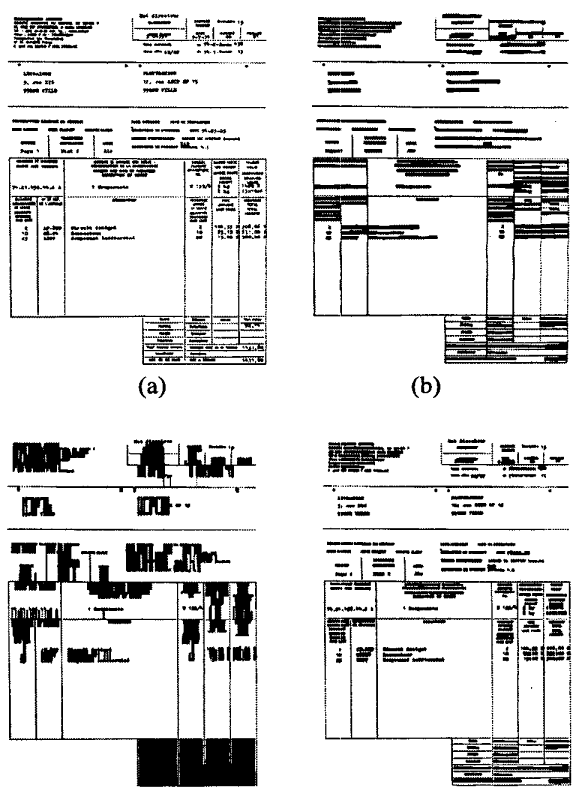


Fig. 3. RLSA application. (a) original image, (b) Horizontal RLSA, (c) Vertical RLSA, (d) final smoothed image.

In the case that the image is represented by a set of non-overlapping rectangular blocks, we do not need to scan the entire image and pass from all possible white and black runs. We just examine the white runs that start from the right side of all possible image blocks in the case of horizontal smoothing (see Fig. 2b), or from the bottom side of the blocks for vertical smoothing. From all white runs, we select those with short length and turn them to black runs. In this way, a significant acceleration of RLSA is achieved. Since all possible white runs border on image blocks, the smoothed image produced from block represented image is the same with the smoothed image produced from a point represented image.

Let IMAGE(x,y) the binary image X_{MAX} x Y_{MAX} that is decomposed into N rectangular blocks with coordinates (X1[i], Y1[i]) – (X2[i], Y2[i]). The algorithm for a horizontal smoothing with smoothing factor X_f, is as follows:

```
For i = 0 to N Do
        For y = Y1[i] to Y2[i] Do
                x = X2[i] + 1
                X = x
                While IMAGE(x,y) = false And
                x-X < Xf And x < XMAX Do
                        x = x + 1
                Loop
                If x-X < Xf And x < XMAX Then
                        For j = X to x Do
                                IMAGE(j,y) = true
                        Loop
                EndIF
        Loop
Loop
```

4 Block Projection Profiles

The projection profiles of a binary image are evaluated by projecting all pixels to the vertical and horizontal directions (see Fig.4). In the case of a block represented image, the lengths of all black runs of all blocks are known in advance. Taking advantage of this information, the projection profiles are evaluated by adding all black runs of all image blocks at the horizontal and vertical directions. The horizontal profile evaluation of a block represented binary image is demonstrated in the example of Fig. 5.

5 Experimental results

The proposed techniques have been implemented and tested with a variety of images. Experiments were conducted with binary images represented by a set of non-overlapping rectangular blocks. We produced smoothed images using block RLSA and calculated horizontal and vertical projection profiles by adding the black runs of all blocks. The resulting smoothed images were found to be identical to the corresponding images that result from point represented images. We also checked that horizontal and vertical projections are identical for point and block representation. Table 1 shows the times consumed, when processing the 8 CCITT b/w images, for RLSA application and projection profiles evaluation starting from point or block representations as well as the CPU time gained when using block representation. The average time gain is recorded at 60% for RLSA application and 88% for projections profile evaluation.

6 Conclusions

This paper proposes new techniques for fast implementation of two of the most well-known segmentation techniques in binary images, i.e. the RLSA and projection profiles. The proposed approaches are based on decomposition of the binary images in a set of non-overlapping rectangular blocks. Since the starting points of all horizontal and vertical white stripes are known in advance, we achieve a significant acceleration of the RLSA application. Also, projection profiles are implemented by simply adding the areas of all rectangular blocks projected horizontally or vertically. The experimental results obtained are very promising.

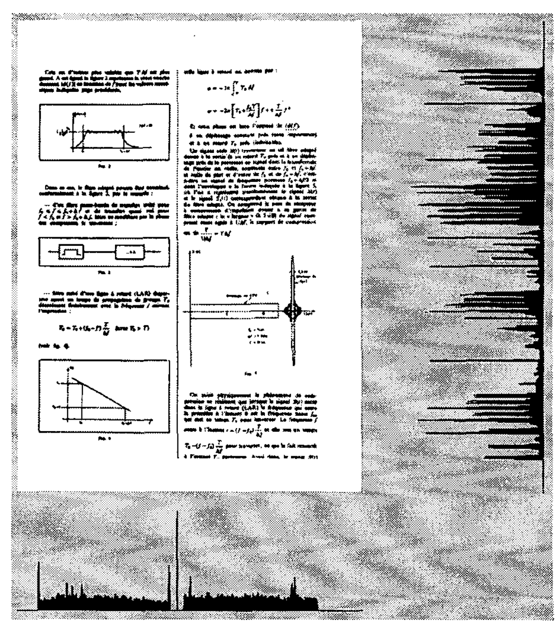
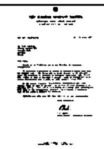
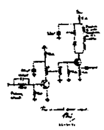
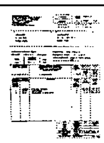
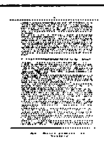
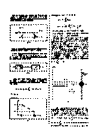
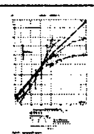
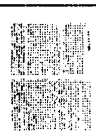


Fig. 4. Projection profiles example.

Table 1. Processing times for RLSA and Projection Profiles

		RLSA	Projection Profiles
	Point representation	0,285 sec	0,092 sec
	Block representation	0,055 sec	0,005 sec
	CPU gain	**80,70 %**	**94,56 %**
	Point representation	0,285 sec	0,093 sec
	Block representation	0,050 sec	0,004 sec
	CPU gain	**82,46 %**	**95,70 %**
	Point representation	0,300 sec	0,096 sec
	Block representation	0,105 sec	0,010 sec
	CPU gain	**65,00 %**	**89,58 %**
	Point representation	0,340 sec	0,105 sec
	Block representation	0,175 sec	0,025 sec
	CPU gain	**48,53 %**	**76,19 %**
	Point representation	0,295 sec	0,098 sec
	Block representation	0,100 sec	0,011 sec
	CPU gain	**66,10 %**	**88,77 %**
	Point representation	0,285 sec	0,095 sec
	Block representation	0,075 sec	0,006 sec
	CPU gain	**73,68 %**	**93,68 %**
	Point representation	0,330 sec	0,101 sec
	Block representation	0,160 sec	0,021 sec
	CPU gain	**51,51 %**	**79,21 %**
	Point representation	0,280 sec	0,121 sec
	Block representation	0,235 sec	0,015 sec
	CPU gain	**16,07 %**	**87,60 %**
Average	Point representation	0,300 sec	0,100 sec
	Block representation	0,119 sec	0,012 sec
	CPU gain	**60,33 %**	**88,00 %**

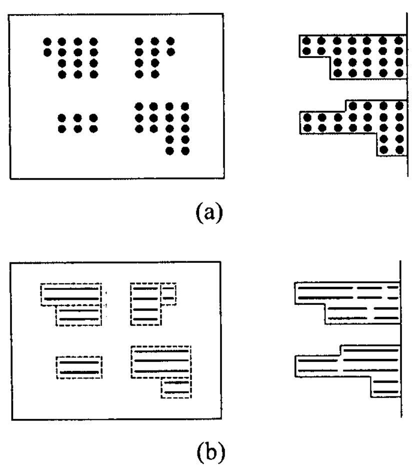


Fig. 5. Horizontal Projection Profiles. (a) Point representation, (b) Block representation.

References

[1] M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis and Machine Vision*, Chapman & Hall, London, 1993.

[2] M. H. Singer, "A general approach to moment calculation for polygons and line segments" , *Pattern Recognition*, vol. 26, pp. 1019-1028, 1993.

[3] D. Wang, V. Haese-Coat and J. Ronsin, "Shape decomposition and representation using a recursive morphological operation", *Pattern Recognition*, vol. 28, pp. 1783-1792, 1995.

[4] S. J. Perantonis, B. Gatos and N. Papamarkos, "Block decomposition and segmentation for fast Hough transform evaluation", *Pattern Recognition*, vol. 32(5), pp. 811-824, 1999.

[5] B. Gatos, N. Papamarkos, I. Andreadis and S. J. Perantonis, "Applying Morphological Transformations on a Block Represented binary Image", Proc. of the *Fourth International Workshop on Document Analysis Systems (DAS2000)*, pp. 487-495, Rio de Janeiro, Brazil, December 2000.

[6] S. Perantonis, B. Gatos and N. Papamarkos, "Image segmentation and linear feature identification using rectangular block decomposition", Proc. of the *Third IEEE Int. Conf. On Electronics, Circuits and Systems, ICECS 96*, pp. 183-186, Rodos, Greece, 1996.

[7] A. Quddus, and M. M. Fahmy, "Binary text image compression using overlapping rectangular partitioning", *Pattern Recognition Letters*, vol. 20, pp. 81-88, 1999.

[8] K. C. Fan, D. Chen and M. Wen, "Skeletonization of binary images with nonuniform width via block decomposition and contour vector matching", *Pattern Recognition*, vol. 31, pp. 823-828, 1998.

[9] R. Kasturi, S. Bow, W. El-Masri, J. Shah, J. Gattiker and U. Mokate, "A System for Interpretation of Line Drawings", *IEEE Trans. On Patt. Anal. And Mach. Intell.*, vol. 12, pp. 978–991, 1990.

[10] J. L. Fisher, S. C. Hinds and D. P. D'Amato, "A rule-based system for document image segmentation", Proc. of the *10th Int. Conf. On Pattern Recognition*, pp. 567-572, 1990.

[11] A. Verikas, M. Bachauskene, S. Vilunas and D. Skaisgiris, "Adaptive Character Recognition System", *Pattern Recognition Letters*, vol. 13, pp. 207–212, 1992.

[12] B. Gatos, N. Gouraros, S. Mantzaris, S. Perantonis, A. Tsigris, P. Tzavelis and N. Vassilas, "A new Method for Segmenting Newspaper Articles", Proc. of the *Second European Conference On Research and Advanced Technology for Digital Libraries (ECDL'98)*, pp. 695-696, Heraklion, Crete, Greece, September 1998.

[13] G. Ciardiello, G. Scafur, M. T. Degrandi, M. R. Spada and M. P. Roccoteli, "An experimental system for office document handling and text recognition", Proc. of the *9th Int. Conf. On Pattern Recognition*, pp. 739-743, 1998.

[14] C. Strouthopoulos, N. Papamarkos and C. Chamzas, "PLA using RLSA and a neural network", *Engineering Applications of Artificial Intelligence*, vol. 12, pp. 119-138, 1999.