Document Decomposition Based on Recursive Radon Transform

Ivo R. Draganov[1] and Roumen K. Kountchev[2]

Abstract – **In this paper we present a robust and efficient algorithm for document decomposition based on recursive Radon transform and morphology. The input image is divided into multiple fields by type – text and graphics not overlapping each other. Then for each text field line detection and word extraction are performed. High accuracy is achieved for the extracted words to be recognized afterwards.**

Keywords – **document decomposition, radon transform, profile projection analysis**

I. Introduction

Document decomposition is an important stage in the document processing systems where different fields such as text with different formatting, graphics, tables, equations etc. should be detected and extracted. In general there are three types of approaches – top-down, bottom-up and hybrid ones [1, 3]. The first group relies on examination of the document image as a whole and separating it to its compounds using a set of rules until some criterion is met [2]. The second type methods use the image pixels as a starting point. Grouping and labelling them into growing regions are then realized for decomposing the document [6]. And the third group is a mixture of both the approaches [4].

According to Mao et al. [3] the following limitations in previous works are met: the absence of formal models for the pages structure which not allows framework incorporation, model parameters estimation and document synthesis; the use of deterministic models and quantitative performance neglecting.

Here we use a simplified structure model of a document to be decomposed based on recursive Radon transform and morphology – a typical top-down approach and decomposition error is given into account. Our main goal is to typify the steps starting from the whole input image and ending with the separate fields. Thus using recursion it is possible to accelerate the whole process instead of using additional analysis for each block extracted.

In part two we give description of our formal document model and of the recursive algorithm. In part three some experimental results are given along with comparison to a previously developed and proven in practice algorithm and in part four a conclusion is made.

II. Document Model and Algorithm Description

The formal model of the physical structure of the documents we are going to process is given in Fig.1. It includes only text fields, graphics and tables not overlapping each other. Each block is represented of upper left and bottom right corner coordinates (A_p, B_p), p = 1÷P. (A_0, B_0) are the boundaries defining corners of the whole document.

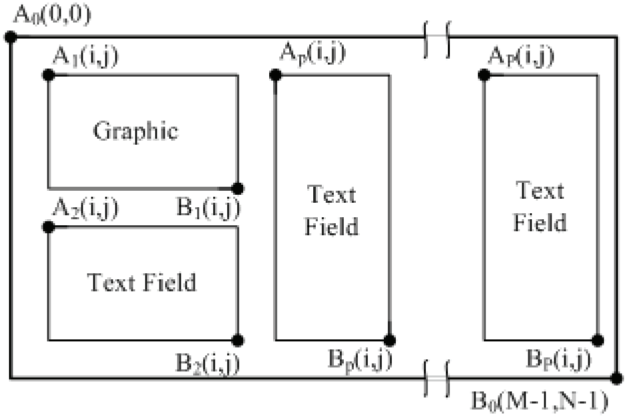


Fig.1. Proposed document physical structure model

We make the following assumptions:

- no two fields overlap with each other;
- all the text fields contain only horizontally printed text which may be of different font, size and style – which covers almost all languages except the east-asian ones which means a horizontally periodicity is observable in each text filed;
- graphics may contain all kinds of patterns, even such with a periodic structure but statistically for a wide range of images it should not be preferably oriented, horizontally or otherwise.

All the limitations mentioned above do not drastically limit the applicability of our algorithm – most of the newspaper and magazine pages correspond to them. Knowing this here follows the steps of the proposed algorithm.

Step 1 – getting the input image, in color for the most general case in RGB color space, [$R(i,j)$, $G(i,j)$, $B(i,j)$], where R,G,B = 0÷255; i = 0÷M-1 and j = 0÷N-1. Conversion is made then to grayscale:

$$I(i,j) = 0.30R(i,j) + 0.59G(i,j) + 0.11B(i,j)\,, \qquad (1)$$

where $I(i,j)$ = 0÷255.

Step 2 – binarization using Otsu algorithm according to:

[1] Ivo R. Draganov is with the Faculty of Telecommunications, Technical University, Kliment Ohridski 8, 1000 Sofia, Bulgaria, E-mail: idraganov@abv.bg

[2] Roumen K. Kountchev is with the Faculty of Telecommunications, Technical University, Kliment Ohridski 8, 1000 Sofia, Bulgaria, E-mail: rkountch@tu-sofia.bg

$$I_b(i,j)=\begin{cases}1, \text{ if } & I(i,j)\ge t_0\\ 0, \text{ if } & I(i,j)<t_0\end{cases}, \qquad (2)$$

where t_0 is the optimal brightness threshold. Working with binary image is faster for the next steps and since we are looking only for the coordinates of (A_p,B_p) the loss of information is of no importance.

Step 3 – dilate the binary image with structural element $S(k,l)$ = 1, for k = 0, l = [-7,+7], i.e. a straight line element with length of 15 pixels:

$$I_{bd}(i,j)=\underset{k,l\in S}{\mathrm{Y}} I_b(i+k,j+l). \qquad (3)$$

The aim of using this step is to fuse all the characters from each text line in each text field and get their bounds in the next step. Homogenous parts from graphics obtained after the binarization with small disconnections are now filled entirely.

Step 4 – edge detection using Sobel operator:

$$\begin{aligned}I_{be}(i,j)=\Big\{&\big[G_x\otimes I_{bd}(i+m,j+n)\big]^2+\\ &+\big[G_y\otimes I_b(i+m,j+n)\big]^2\Big\}^{1/2}\end{aligned}, \qquad (4)$$

$$G_x=\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix},\ G_y=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}. \qquad (5)$$

The meaning of this operation is that now we get only the boundaries of the text lines – all collinear at their longer sides to each other for all text fields and the boundaries of the objects inside the graphics – in general arbitrary curves.

Step 5 – Radon transform over $I_{be}(i,j)$:

$$R_\theta(j')=\sum_{i'=-N/2}^{N/2-1}\sum_{j'=-M/2}^{M/2-1} I_{be}(i',j'), \qquad (6)$$

$$\begin{bmatrix}i'\\ j'\end{bmatrix}=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}i\\ j\end{bmatrix}. \qquad (7)$$

If there is a skew in the initial document image of which magnitude we are not aware θ = [0,179]° should be used with a larger step. Then when we know roughly the skew θ_s we should use $\theta = [\theta_s\text{-}1,\ \theta_s\text{+}1]$° with a smaller step of about 0.1° which is enough precision according to [5].

Step 6 – find the vertical profile projection from the Radon transform with minimal entropy, according to:

$$H_{min}(\theta_{se})=-\sum_{j'=-M/2}^{M/2-1} R_{\theta_{se}}(j')\,log_2\,R_{\theta_{se}}(j'), \qquad (8)$$

where θ_{se} is the exact skew with precision of 0.1°.

Step 7 – binarize $R_{\theta_{se}}(j')$ using again Otsu algorithm:

$$R_{\theta_{se}b}(j')=\begin{cases}1, \text{ if } & R_{\theta_{se}}(j')\ge r_0\\ 0, \text{ if } & R_{\theta_{se}}(j')<r_0\end{cases}, \qquad (9)$$

where r_0 is the optimal threshold.

Step 8 – analyze the binarized profile $R_{\theta_{se}b}(j')$. There are two basic cases – the document has a single text column (Fig.2.a) or a single graphic (Fig.2.b). The single dots correspond to accumulations obtained from text lines boundaries – peaks in the projection profile preserved after the binarization in step 7.

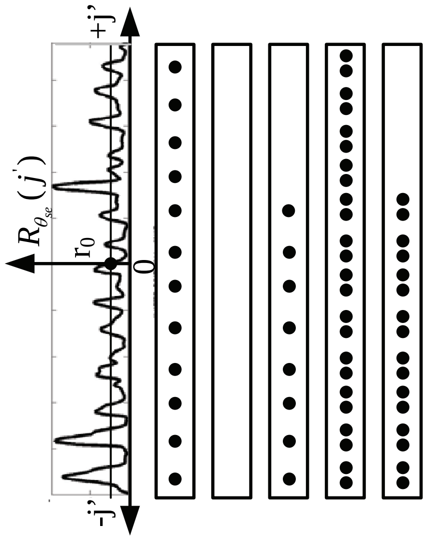


Fig.2. Different binarized projection profiles $R_{\theta_{se}b}(j')$

Fig.2.c is for a document with a graphic and a single text column beneath it, Fig.2.d of a two column document and Fig.2.e – of a graphic and two text columns underneath. When we have adjacent text columns multiple periodicities will emerge – for every text line in a column two by two dots will repeat with their distance between them and only if the separate columns are exactly collinear by their lines different dots from different columns will coincide with each other. Because of that even if we get a profile like those from Fig.2.a we need to make a horizontal projection profile to be sure we have only one column instead of two adjacent ones.

So in this step we introduce a cut point along *j'* each time a gap in the periodicity of dots is observed. Afterwards step 7 is repeated for each two parts – upper and lower of the nonbinarized profile $R_{\theta_{se}}(j')$. Here new dots may appear if there is a text column next to a graphic shorter than these above and below it.

Step 9 – horizontal projection profile is made for each graphic and text block found no matter if there is a single or multiple periodicities observed in the latter:

$$Q(i') = \sum_{i'=-N'/2}^{N'/2-1} \sum_{j'=-M'/2}^{M'/2-1} I(i', j'), \qquad (10)$$

where *M'* and *N'* define the size of the currently processed block in pixels. If *Q(i')* ≠ 0 for each *i'* then the block is homogenous one and can be extracted as such by its opposite vertexes $A_p(i,j)$ and $B_p(i,j)$. For every *Q(i')* = 0 a horizontal cut point is introduced and for the new two parts steps 7-9 are repeated until no zeros in the horizontal projection profile are observed.

The reiteration of step 7 after step 8 and steps 7-9 after step 9 brings two recursions – along horizontal and vertical directions. More general case would be if we repeat step 5 also before vertical and horizontal cut is made but only if we expect different text and graphic fields to have their own skew again not overlapping each other – thus we get recursive Radon transform.

After the whole document decomposition to fields' level is made for each text field from the last vertical projection profile each two dots no matter starting from the bottom or from the top corresponds to the upper and the lower boundary of each text line. Then for each text line we find the horizontal projection profile and from all the profiles obtained get a histogram of the lengths of the zero valued gaps in them. It is normally to assume and appears exactly that this histogram is two-mod one – a mod situated in the range of the smaller length values for the gaps between the characters and another – for the bigger gaps between words. Again Otsu algorithm can be used to find an optimal threshold separating the two modi – every time a gap is observed in text line with length above it then a cut point should be introduced separating neighbouring words.

III. Experimental Results

As experimental dataset we use one color image (RGB, 24 bpp) of a PAMI transactions page with dimensions 2550x3300 pixels scanned at 300dpi used as well by Jain and Yu [4] so we can make a direct comparison to their algorithm. We use IBM compatible PC with Pentium 4 CPU at 3,2 GHz, 1 GB RAM, MS Windows XP SP2, Matlab R2007A.

In Fig.5 are given images of the page obtained at different stages from the proposed algorithm and some of the parameters used as well: a) the grayscale image of the page, b) image with edges detected, c) Radon transformed image for θ = [*89*, *91*]° with step 0.1°, d) vertical projection profile for θ_{se} = 90° and e) binarized vertical projection profile.

In Fig.3 is given the histogram of intercharacter and interword distances for text fields of 36 text lines with the same formatting. The length threshold for interword distance is detected to be 5 pixels.

In Table I a comparison is made between the algorithm proposed by Jain and Yu [4] for the correctly segmented symbols from a text field and the total CPU time for the single page decomposition. The segmentation accuracy of words by means of nondisintegrated characters and time consumption are almost the same with a small advantage for our algorithm.

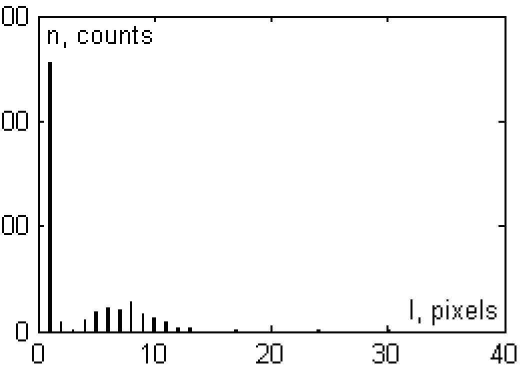


Fig.3. Intercharacter and interword distances histogram

TABLE I
TEXT SEGMENTATION ACCURACY AND TIME CONSUMPTION COMPARISON

Parameters Method	Correctly Segmented Symbols	Wrongly Segmented Symbols	Correct Rate, %	Total CPU Time, sec
Jain and Yu, [4]	1556	9	99,4	1,3000
Our	1558	7	99,6	1,2030

In Fig.4 we make a closer examination for the reason of the wrongly segmented symbols presence. Obviously there are two types of errors occurring. The first one (Fig.4.a) concerns characters which are rounded at the bottom or at the top like 'a', 'e', 'o' etc. Because of poor quality at some places in the original a fixed structural element used to dilate the image can't cope with the erosion present.

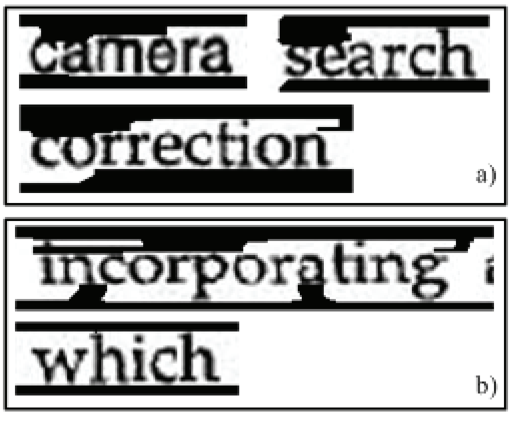


Fig.4. Wrongly segmented symbols

The second error appears in one place (Fig.4.b) but potentially can be observed often if a larger dataset had been used. In longer words with no characters containing ascenders near the character 'i' like 'incorporating' the dot above it can be separated and become a reason for wrong recognition while in words like 'which' i.e. with ascenders present near the 'i' such problem doesn't exist.

IV. Conclusion

The proposed algorithm for document decomposition based on recursive Radon transform is as accurate and fast as other well proven one in the practice. As further work an algorithm for adaptive selection of structure element parameters and more sophisticated document models should be developed.

Acknowledgement

This paper was supported by the National Fund for Scientific Research of the Bulgarian Ministry of Education and Science (Contract - BY-И-305/2007).

References

[1] L. Likforman-Sulem, A. Zahour and B. Taconet, “Text Line Segmentation of Historical Documents: A Survey”, *International Journal on Document Analysis and Recognition*, Vol. 9, No. 2, pp. 123-138, April 2007.

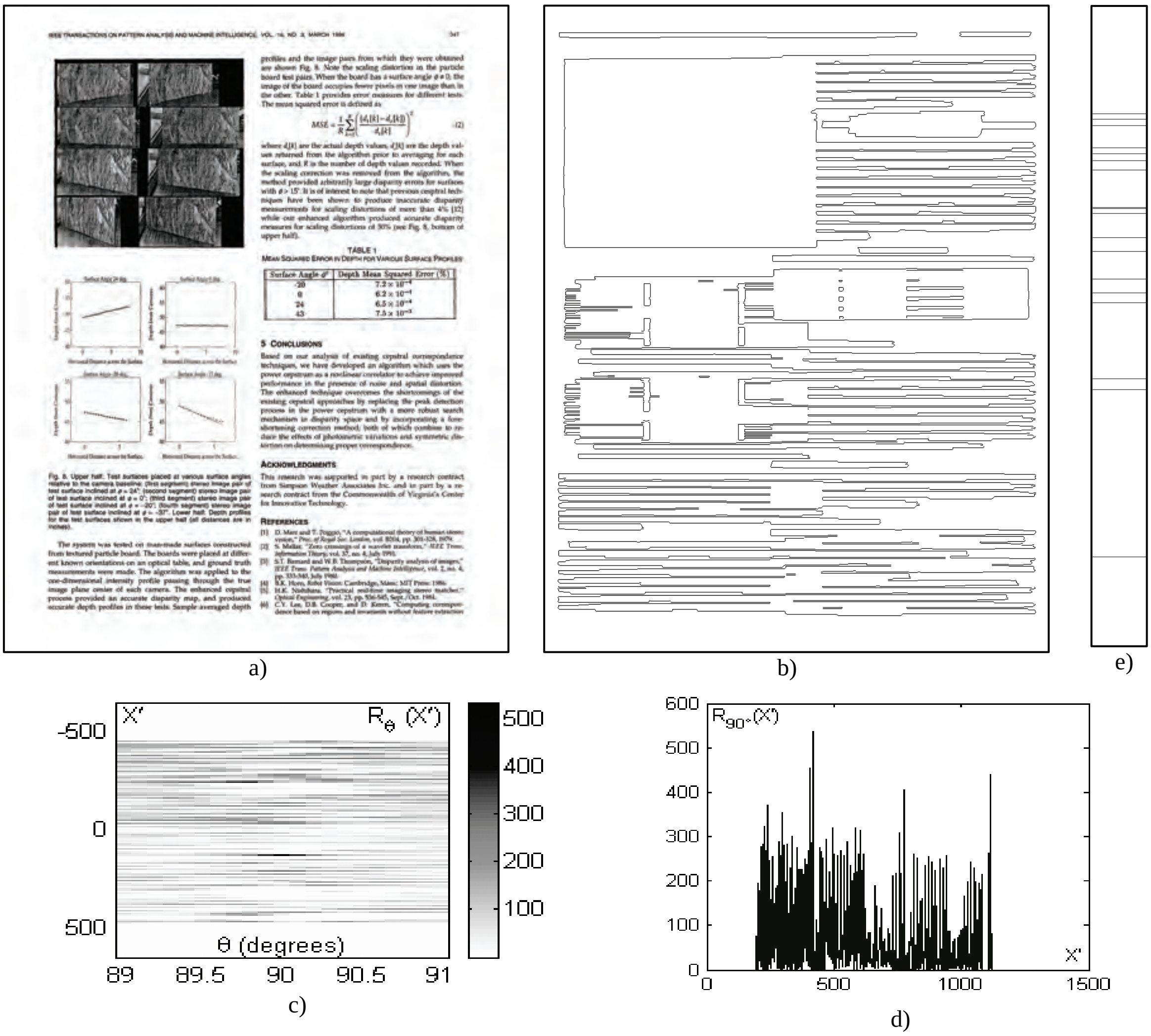
[2] A. Popova, M. Dimitrov and V. Grancharov, “Text Regions Segmentation in Image Printed Documents”, *ICEST’2004 Conference Proceedings*, Vol. 1, pp. 139-142, Bitola, Macedonia, 2004.

[3] S. Mao, A. Rosenfeld and T. Kanungo, “Document Structure Analysis Algorithms: A Literature Survey”, *In Proceedings of SPIE Electronic Imaging*, Vol. 5010, pp. 197-207, January 2003.

[4] A. Jain and B. Yu, “Document Representation and Its Application to Page Decomposition”, *In Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp. 294-308, March 1998.

[5] A. Bagdanov and J. Kanai, “Evaluation of Document Image Skew Estimation Techniques”, *In Proceedings of SPIE*, Vol. 2660, pp. 343-353, March 1996.

[6] J. Ha, R. Haralick and I. Phillips, “Document Page Decomposition by the Bounding-Box Projection Technique”, *ICDAR’95 Conference Proceedings*, Vol. 2, pp. 1119-1122, Montreal, Canada, August 14-15, 1995.



a) b) e)

c) d)

Fig.5. Different stages of processing the test page