IEEE India Council Conference, INDICON 2009; Ahmedabad; 18 December 2009 through 20 December 2009; Category number CFP09598; Code 79708; Article number 5409409; DOI- http://dx.doi.org/10.1109/INDCON.2009.5409409

Text Extraction from Document Images using Edge Information

Sachin Grover
National Institute of Technology
Rourkela, Orissa 769008
Email: groversachin.211@gmail.com

Kushal Arora
National Institute of Technology
Allahabad, Uttar Pradesh 211004
Email: kushal18@gmail.com

Suman K. Mitra
Dhirubhai Ambani Institute of
Information and Communication Technology
Gandhinagar, Gujarat 382007
Email:sumanmitra@daiict.ac.in

***Abstract*—Detection of text from documents in which text is embedded in complex colored document images is a very challenging problem. There are a lot of potential uses of text extraction in image searching, archiving documents etc. In this paper, we propose a simple edge based feature to perform this task. It aims at detecting textual regions from the document and separating it from the graphics portion. The algorithm is based on the sharp edges of the characters which are missing in images. We find these edges and use them to classify text from images. This edge information can also be used for other image interpretation tasks.**

I. Introduction

Documents in which text is embedded in complex colored backgrounds are increasingly common today, for example, in magazines, advertisements and web pages. Robust detection of text from these documents is a challenging problem. Text extraction has a vast number of applications :

- Text searches in Images - Currently, Image searches deliver inaccurate results as they do not search the image content. Text extraction would enable better searching by extracting the content of an image.
- Content based Indexing - For the purpose of archiving and indexing documents, the content of the document is required in the digital format. Knowledge about the text content of documents can help in the building of an intelligent system which archives and indexes the printed documents.
- Reading foreign language text - One of the common problems faced by a person in foreign land is that of communication, understanding road signs, signboards etc. The proposed method, aims to alleviate such problems by reading the text information from the image scenes which are captured by a camera.
- Archiving documents - Archives of paper documents in offices or other printed material like magazines and newspapers can be electronically converted for more efficient storage and instant delivery to home or office computers.

In this paper, we will demonstrate that simple texture measures based on edge information provide very useful information for text detection from complex document images.

II. Challenges and Related Works

Text extraction from images is a very challenging problem. To find a completely robust and generalized method for text segmentation, still remains an area of further research. Here a method has been proposed which is independent of the orientation of the text, background of the image as well as font size of the text.

A lot of work has been done in document segmentation techniques. The approaches developed for ordinary documents, such as binarization by adaptive thresholding, are not generally applicable, because it seems impossible to find an optimal threshold. Traditionally, page segmentation methods are divided in three groups: top-down, bottom-up and hybrid approaches [1].

Bottom up techniques [2], [3] typically use merging and grouping from the thinnest elements characters and merging them recursively to words and then to text lines to paragraphs. Most popular bottom-up techniques are mathematical morphology, run length smoothing algorithm, and region growing-based methods.

Top down techniques [4] start by detecting the large scale features of the image and proceed by successive splitting until they reach the the smallest scale features (individual characters or text lines). These techniques are often fast, but the efficiency depends on priori knowledge about the class of documents to be processed. The most well known are projection methods, histogram analysis, rule based systems, or space transforms (Fourier transform, Hough transform, etc.).

There are others that do not fit into these categories and are therefore called hybrid. Among these texture based [5] and background analysis based are important. Examples of these are those based on Gabor filtering and mask convolution, fractal signature and wavelet analysis.

Texture-based methods have been proposed as a solution to this problem, where text is assumed to represent a different texture than non-text. With this approach, text is often considered as a periodic texture, because the characters on a text line form a more or less periodic structure in the horizontal direction, while the text lines, in turn, form a periodic structure in the vertical direction. But going by this, the orientation of text comes under consideration and thus we have to work on the horizontal, vertical and angular text differently.

978-1-4244-4859-3/09/$25.00 ©2009

There are two different approaches to texture based text segmentation: pixel-based and block-based. According to the first approach, pixels are segmented into text or non-text on the basis of texture information computed in the neighborhood of each pixel [6], [7]. Finding the thresholds for the classification is a big problem in this. With the second approach, the image is divided into non-overlapping or partly overlapping blocks and then each block is labeled based on texture information in it [8], [9], [10].

The textures occurring in document images are quite different from the ordinary textures, because the edges of the characters are sharp and the background intensity is sometimes gradually changing. Because of this, it may not be reasonable to apply texture operators designed for gray scale images, but to use operators that measure the most relevant information from document textures instead. These operators should be rotation invariant, because the edges of the characters can have all orientations and the text lines can also be often in different directions. The methods to be chosen should also be invariant to gray level or color variations in the background.

Ojala et al. [11] have shown that key information for texture discrimination is provided by two orthogonal features measuring local spatial pattern and contrast information, respectively. Their results also suggest that the densities of these local features computed over a region should be used for texture description. In contrast, the spatial frequency information often used in mainstream research does not appear to be so important.

Following these lines, the edge-based texture measures appear to have many of the desired properties. The gradient magnitudes usually have high values in the edges of the characters, even when the text is embedded in pictures. The edges are also invariant with respect to the background variations and to image rotation. An additional advantage of the edge-based approach is that the same detected edges can also be used for other image interpretation tasks.

III. Proposed Methodology

Our method for text extraction consists of following steps:

1) Color to gray scale conversion of the image, if necessary.
2) Edge detection by 3X3 Sobel operator, non maximum suppression, and thresholding.
3) Edge image partitioning into small non-overlapping blocks and computing an edge-based feature for each block.
4) Block satisfaction either as text or as non-text based on the value edge-based feature.
5) Post processing of the result obtained for the improvement of text detection.

Since we are interested in edges, it is natural to detect them in a gray-scale image.One of the simplest ways for converting an image to gray scale is by forming a weighted sum of the R, G, and B components. Then edge detection is then performed on the gray-scale image by convolving the image with Sobel masks, separately for horizontal and vertical edges.

The Sobel masks are used because it is a second degree gradient operator and is much more aggressive in enhancing sharp changes. Thus it is best suited to find the sharp edges of the text.

Convolution is followed by elimination of non-maxima and thresholding of weak edges. A threshold th_1 for eliminating weak edges is computed using the formula:

$$th_1 = \sqrt{\frac{\sum_{i=1}^{h-1}\sum_{j=1}^{w-1} G_x^2(i,j) + G_y^2(i,j)}{(h-1)(w-1)}} \tag{1}$$

where h and w are the height and width of the image and $G_x(i,j)$ and $G_y(i,j)$ are the x- and y-components of the gradient magnitude $G(i,j) = \sqrt{G_x^2(i,j) + G_y^2(i,j)}$ for a pixel at (i,j).

After this, the edge image is divided into small non overlapping blocks of $m * m$ pixels, where m depends on the image resolution. Typical values of m can be $[0.05R, 0.3R]$ (for all kind of texts)where R is the resolution of the image in dpi. Dependency of m on R is apparent by the fact that same text will have more pixels in the same frame. And thus the value of m would increase as the value of R is increases. The constant of proportionality was found to vary between 0.05 to 0.3.For each block feature F is calculated using the formula :

$$F = \frac{\sum_{i=1}^{h-1}\sum_{j=1}^{w-1} G_(i,j) H(G(i,j) - th_1) H(E(i,j) - 1)}{m^2} \tag{2}$$

where $E(i,j)$ is the edge image (1-edge and 0-non-edge) and H is the step function.

$$H(x-a) = \begin{cases} 1 & \text{if x}\geq\text{a,} \\ 0 & \text{if x}<\text{a} \end{cases}$$

In equation 4, F is the average gradient magnitude per edge pixel and the average gradient magnitude per pixel. F defined above reflects the fact that *the number of edge pixels and their gradient magnitudes are usually higher for text than for non-text blocks*.

Using this property we perform Block classification using pre-defined threshold th_2 which will distinguish the text from the image (blocks with $F > th_2$ are assigned to text and rest to non text). Unfortunately, it is difficult to determine its value automatically. F_{min} and F_{max} were found first. For our experiment the value of th_2 was required to be closer to lower bound so that no kind of text is missed. The value of th_2 was determined using $th_2 = k * (F_{max} - F_{min})$ where k varies between $[0.1, 0.3]$. We assume that the large values of F are associated with text, whereas the small values may correspond either to text or to non-text.

Post-processing techniques were used to decrease false alarm rate. The need of it was felt seeing the single m sized boxes where the edge of the image had a gradient value very close to that of the text. The blocks which did not have any neighbors were removed from the result as a single m sized block could not hold any text. This was done by simply giving an arbitrary value to every block if it had a neighbor and giving

a different value if it does not. This way the false alarm rate was decreased.

Binary image was made marking the text region in white and non-text region in black. The results were marked in the original image.

A. Pseudo code

The pseudo code of the algorithm is as follows:

initialize the picture
$i := 0.299 * red + 0.587 * green + 0.119 * blue$; ▷ gray scale conversion
$h := [1\ 2\ 1; 0\ 0\ 0; -1\ -2\ -1]$; $v := [1\ 0\ -1; 2\ 0\ -2; 1\ 0\ -1]$;
$i_h := convolution(i, h)$; $i_v := convolution(i, v)$;
for every pixel **do**
    $temp := \sum i_h^2 + i_v^2$; $g(i,j) := \sqrt{i_h^2 + i_v^2}$;
end for
$h(i,j) := 0$; ▷ new white image created
$th := \sqrt{4 * temp/length * breadth}$; ▷ Average value
$th_1 = th/\text{maximum}(g)$;
if $g(i,j)/\text{maximum}(g) \geq th_1 \&\& g(i,j) \geq th$ **then**
    $h(i,j) = 1$; ▷ Edges marked
end if
divide the image into box of size m;
for each box **do**
    $F_1 := step(g(i,j) - th) * step(g(i,j) - 1)/m^2$;
    $F(i,j) := \sum_{i=1}^{h-1} \sum_{j=1}^{w-1} g(i,j) * F_1$;
end for
$th_2 := k * (F_{max} - F_{min})$;
if !neighbor **then** ▷ checking for neighbor
    $arb(i,j) = 2$;
else
    $arb(i,j) = 1$;
end ifres(length,breadth) = zeroes; ▷ new image for the final binary image
if $F(i,j) > th_2 \&\& arb == 2$ **then**
    $res(wholebox) := 1$;
end if

IV. Experimental Results

A huge set of test pictures, scanned from newspapers and magazines, scanned at 150, 300 and 600dpi were selected. There were pictures taken to test for usual text, that is white paper and black text. Then there were pictures taken to test with different colored background as well as vertical text. Even angled text was tested and there were some test pictures for text of different language and different fonts. The results were promising in all the test pictures. The values of k and m used were found through trial and error.

On decreasing the value of k the false alarm rate increases significantly. The value of k depends on how bright the edge has been detected. If the image is good we get good results for k = 0.25 .

We have also worked on finding a suitable value for m. We found that for smaller text, value of m is between $[0.1R, 0.15R]$ and for larger text, value is between $[0.2R, 0.25R]$. But we could not find an appropriate value for both kind of texts which could work equally well simultaneously.

The algorithm works well for the vertical text. Fig 1 shows the experimental result, when the proposed algorithm is used for a newspaper article image scanned at 150dpi. Here the value of m used is 30 ($0.2R$) and k = 0.2. Here the rate of false alarm was 0.05 and sensitivity was 0.990.

The algorithm also works equally well in case the text is on colored background. Fig 2 shows the experimental result. The value of m used is 40 ($0.133R$) and k =0.22. Here the rate of false alarm was a little to 0.04 but sensitivity was 0.991.

The algorithm also worked equally well on circular text as the algorithm does not depend the orientation.

V. Conclusion

The results were good with high sensitivity and low false alarm rate. For large text we can see that as the block marks the exact boundary and leaves the gap in the alphabet too. But there is a disadvantage of using this method when the gradient of intensities of text and image are quite similar. Finding a generalized value which can work on every kind of image also needs some working.

References

[1] L. O. Gorman and R. Kasturi, *Document Image Analysis*. Los Alamitos, California, USA: IEEE Computer Society Press, 1995.

[2] D. Wang and S. N. Srihari, "Classification of newspaper image blocks using texture analysis," *Computer Vision Graphics, and Image Processing*, vol. 47, pp. 327–352, 1989.

[3] T. Pavlidis and J. Zhou, "Page segmentation and classification," *Computer Vision Graphics, and Image Processing*, vol. 56, no. 6, pp. 484–496, 1992.

[4] Q. Yuan and C. L. Tan, "Text extraction from gray scale document images using edge information."

[5] A. K. Jain and S. Bhattacharjee, "Text segmentation using gabor filters for automatic document processing."

[6] D. F. Dunn and N. E. Mathew, "Extracting colour halftones from printed documents using texture analysis," *Pattern Recognition*, vol. 33, no. 3, pp. 445–463, 2000.

[7] M. I. C. Murguiu, "Document segmentation using texture variance and low resolution images," in *Proceedings of IEEE Southwest Syniposium on Image Analysis and Interpretation*, Tucson, Arizona, USA, 1998, pp. 164–167.

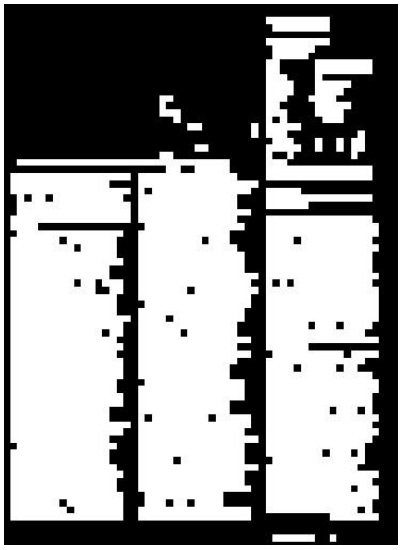
[8] L. Clique, L. Lombardi, and G. Mazini, "A multirestoration approach for page segmentation," *Pattern Recognition Letters*, vol. 19, no. 2, pp. 217–225, 1998.

[9] K. Etemad, D. S. Doermann, and R. Chellappa, "Multiscale segmentation of unstructured document pages using soft decision integration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 92–96, 1998.

[10] A. K. Jain and Y. Zhong, "Page segmentation using texture analysis," *Pattern Recognition*, vol. 23, no. 2, pp. 743–770, 1996.

[11] T. Ojala, T. Menp, and M. Pietikinen, "Gray scale and rotation invariant texture classification with local binary patterns," in *Proceedings of 6th European Conference on Computer Vision*, Dublin, Ireland, 2000, pp. 404–420.

(a) Original Image

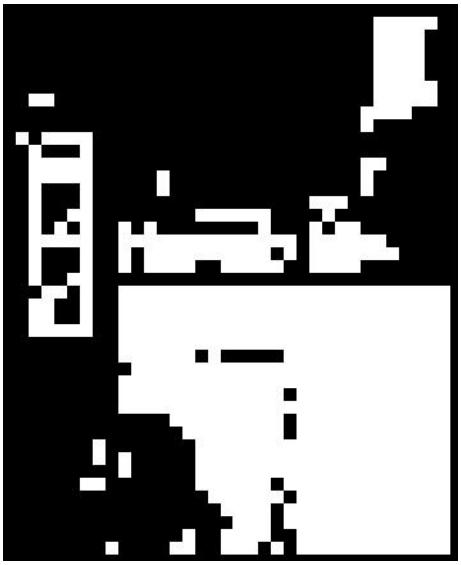


(b) Binary text image (White is the text region)

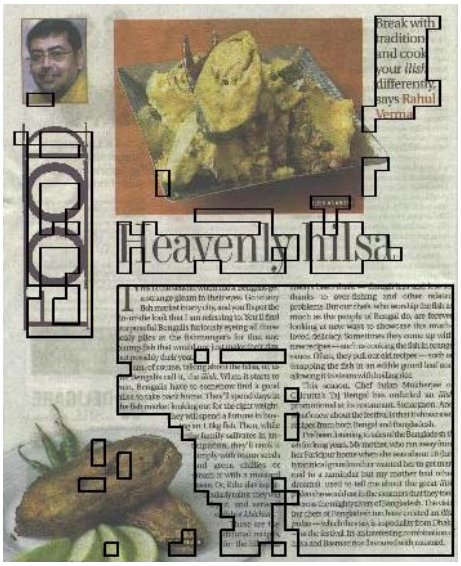
(c) Results marked on Fig 2(a)

Fig. 2. Experimental Results m=40 and k=0.25

(a) Original Image



(b) Binary text image (White is the text region)



(c) Results marked on Fig 1(a)

Fig. 1. Experimental Results m=30 and k=0.2