Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

An Approach to Text Segmentation from Color Document Images with Graphic Background

Hong-Ming Suen and Jhing-Fa Wang

Institute of Information Engineering
National Cheng Kung University, Tainan, Taiwan, R.O.C.

Abstract

Many researches focusing on segmenting text from various documents have been made in the past years. Most of them were devoted to dealing with monochrome documents. On the other hand, how to process color materials is still an open research field. Compared with monochrome documents, the processing of color documents will be much more complicated due to its wide variation in the color of text strings and background. In this paper, we present a procedure for separating text from color document images with graphic background. Our approach adopts the bottom-up scheme in processing. First, a binary edge image is created for finding out the connected components. Next, we select those components which look like characters or part of characters and then classify them according to their colors. Finally, we further group those character-like components which are in the same class and aligned horizontally, and then check if they are really characters. Experimental results have demonstrated the feasibility of the proposed approach.

1. Introduction

Segmenting text from paper-based documents is a critical problem in many automated data entry systems. For example, in an automated bank check reading system, the desired character strings must be first extracted and then recognized. The same procedure is also needed in building a system which can automatically read newspapers, journal articles or other kinds of printed materials. In the past years, many researches focusing on separating text from various documents have been made [1-12]. Most of them devoted to dealing with monochrome documents [1-9]. In monochrome materials, the foreground objects are usually black and the background is usually white. This knowledge has been widely used in devising approaches for processing monochrome documents. However, in the case of color-printed documents, foreground objects can be any color and so can the background. Thus due to having no *a priori* knowledge about the color of text strings, we need a new method to locate them in this case.

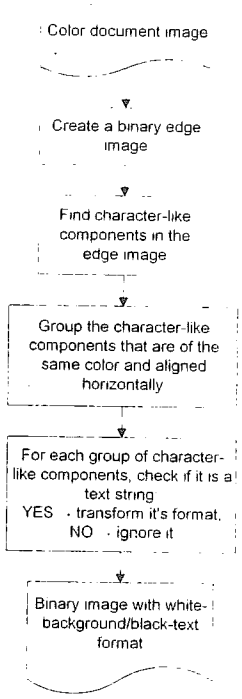


Figure 1. Flow chart of the proposed procedure.

In the approach proposed by Lin *et al.* [10] for color document image segmentation, a color document image is first viewed as a combination of 2^{24} binary images, each of which corresponds to a color plane. Then, to speed up processing, a mean-cut color quantization algorithm is used to decrease the number of the binary images by merging some color planes. Finally, the segmentation and classification techniques used in processing monochrome document images are applied to each binary image independently. In this approach, the color quantization process is a critical step which will affect the following component extraction and identification strongly. For instance, suppose that, after scanning a color document, the R color component of the

text is distributed over the range: 50 <R<80. (This is referred to as the color-diffusion problem.in [12].) Then if the cut point of the color quantization algorithm falls unfortunately within this range, the text will be broken into two parts. As a result, the text will not be successfully recognized any more. The system presented by Chuang *et al.* [11] is an advanced version from Lin *et al.*. In this system, to solve the problem mentioned above, the color planes with means being very close are further merged after color quantization.

In [12], we proposed another approach to text extraction from color document images with uniform-colored background. Good experimental results had demonstrated the feasibility of the proposed approach. In this paper, we further present a procedure devoted to segmenting text from color document images with graphic background. Fig. 1 is the flow chart of the proposed procedure. Our approach adopts the bottom-up scheme in processing. We first find the edge points in the color document image by gradient operation and then create a binary edge image. Next we search the character-like connected components in the edge image and record their locations and colors. In the following processing, the character-like components are first classified according to their colors, and then those components which are in the same class and aligned horizontally are further grouped together. Finally, each group of character-like components is examined to see if it is a text string. If it is, then it will be further transformed into the white-background/black-text binary format; otherwise, it is ignored. The following sections describe and discuss the foregoing steps in more detail.

2. Creating a binary edge image by gradient operation

Gradient operation is a common tool for detecting edge points in gray-level images. The gradient in a gray-level image $f(x, y)$ is the following vector:

$$\nabla f(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

Its magnitude can be seen as the edge strength of a pixel [13]. However, in color images of the RGB model, every pixel is characterized by its R, G, and B color values. In this case, edges are where the R, G, and B color values have abrupt transitions. Thus, we can define the edge strength of each pixel as

$$\text{the edge strength} = |\nabla f_R| + |\nabla f_G| + |\nabla f_B|$$

where $|\nabla f_R|$, $|\nabla f_G|$, and $|\nabla f_B|$ are the magnitudes of the gradients of the R, G, and B color values at that point. Then, with an appropriate threshold for the edge strength, the edge points can be located. In our implementation, if $|\nabla f_R| + |\nabla f_G| + |\nabla f_B| > 320$, then the pixel is considered an edge point; otherwise, it is considered a non-edge point. During processing, we write the result to a new binary image: if the current pixel is an edge point, then the corresponding pixel in the binary image is set to 1 (as white); otherwise, the corresponding pixel in the binary image is set to 0 (as black).

In digital images, the gradient components G_x and G_y must be evaluated by a digitized way. In our procedure, ∇f_R, ∇f_G, and ∇f_B of each pixel are computed

Figure 2. Color document image with graphic background.

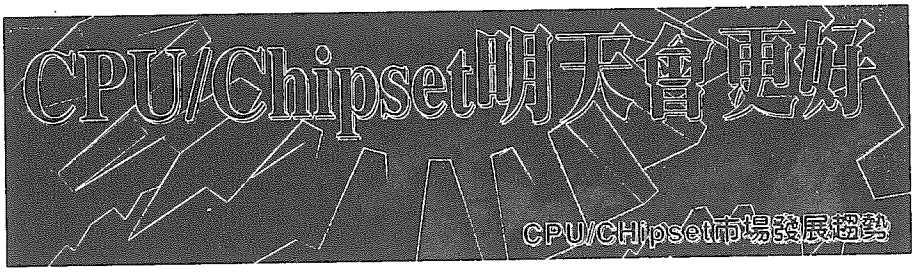


Figure 3. Binary edge image of Fig. 2.

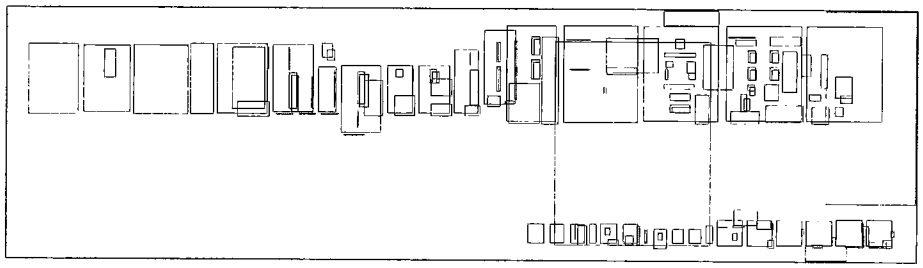


Figure 4. The character-like components found in Fig. 3

by applying the Sobel operators [13] to the R, G, and B color values, respectively. Fig. 2 shows a color document image with size 3360×924 pixels. Fig. 3 is its binary edge image obtained by using the method described above.

3. Finding connected components together with their colors and locations

Connected component labeling [13] is a technique that can help to locate the desired objects in an image. This technique is frequently used in the bottom-up strategy for processing monochrome document images [1,4,7,8]. In our approach, we use the technique to find the connected components of value 0 (i.e., the black region) in the binary edge image. As can be seen in Fig. 3, the connected components of value 0 include the image background, graphic objects, and character components. However, the character components are just what we want, and their sizes are commonly within some range, unlike the image background and graphic objects whose sizes are completely unpredictable. In our system, we use the following rules to previously filter out the image background component and some graphic objects.

1. If the width or height of the connected component is equal to the width or height of the image, then it is a background component.
2. If the width or height of the connected component is greater than 800 pixels, then it is a graphic object.
3. If both the width and height of the connected component are less than 25 pixels, then it is a noise component.

All the connected components identified by the foregoing rules are ignored. The remaining components are then called character-like components and will be further processed by the following steps. Fig. 4 shows all the character-like components found in Fig. 3. In the procedure of connected component labeling, we also record the location and color of each connected component. The location of a connected component is represented by a bounding rectangle that surrounds it; the color of a connected component is represented by the means of R, G, and B color values of the pixels in it.

4. Classifying character-like components by color and alignment

The characters in a text string are commonly of the same color. Hence, if we classify the character-like components according to their colors, those components in the same text string will be clustered together. Besides, since color document images may have wide variety in the color of characters, the classification scheme must be adaptive for them. In our system, we use a two-stage scheme for the classification task. In the first stage, we use the maximin-distance algorithm [14] to examine the color distribution of the character-like components, and then determine an appropriate number of clusters to be gotten. Moreover, the initial cluster centers needed by the classification algorithm employed in the next stage are also selected properly in this stage. In the second stage, we use the K-means algorithm to really perform the classification task. This two-stage classification strategy has been proven by experiments to be suitable for such cases. Table 1 is the classification result of the character-like components shown in Fig. 4. Note that although there are only two kinds of color characters in the original image (one is yellow; the other is white, as shown in Fig. 2), there exist 11 clusters in the classification result. This is because the character-like components include some graphic objects. However, these undesired graphic components will be checked out in the following processing.

Text strings are composed of characters that are oriented along a straight line. In effect, in most cases text strings are aligned horizontally. Thus, in this study, we just search the character-like components aligned in the horizontal direction. However, if needed, the components aligned in other directions can also be detected by the same technique. In our system, we use the Hough transform to detect the horizontally collinear character-like components in each cluster. The Hough

Cluster	Character-like component	Color (R, G, B)	Location (x, y)
1	#1	(183, 100, 55)	(892, 3030)
	#2	(196, 80, 32)	(849, 3059)
	#3	(138, 73, 32)	(836, 2419)
	#4	(193, 73, 37)	(764, 2733)
2	#1	(229, 222, 223)	(821, 2785)
	#2	(233, 223, 222)	(821, 2894)
	#3	(224, 225, 215)	(819, 3223)
	#4	(233, 225, 218)	(821, 3004)
	#5	(228, 225, 223)	(826, 2168)
	. . .	. . .	. . .
	#17	(235, 233, 224)	(825, 2308)
. . .	. . .	. . .	. . .
8	#1	(69, 146, 110)	(366, 2493)
	#2	(88, 188, 147)	(300, 1319)
	#3	(80, 194, 130)	(313, 1063)
	#4	(96, 194, 147)	(258, 1819)
	#5	(101, 194, 142)	(249, 1583)
	#6	(87, 192, 142)	(222, 1955)
	#7	(96, 195, 143)	(154, 1818)
	#8	(92, 192, 138)	(136, 1956)

Table 1. Classification result of the character-like components shown in Fig. 4.

transform is a point-to-line transformation specified by the following equation [13]

$$x\cos\theta + y\sin\theta = \rho.$$

A point (x, y) in the Cartesian space will be transformed into a curve in the $\rho-\theta$ plane. In effect, if (x_1, y_1) and (x_2, y_2) are both on the line specified by the equation

$$x\cos\theta_m + y\sin\theta_m = \rho_n$$

in the Cartesian space, then their corresponding curve in the $\rho-\theta$ plane will intersect at (ρ_n, θ_m). Thus by searching the $\rho-\theta$ plane, we can identify the points that lie on a specific line. In our procedure, if we find a group of character-like components that are aligned horizontally, we go on to calculate their distribution region and then compute the saturation of character-like components in this region using the following formula

$$\text{Saturation of character-like components in the region} = \frac{\text{total area of the character-like components in the region}}{\text{area of the region}}$$

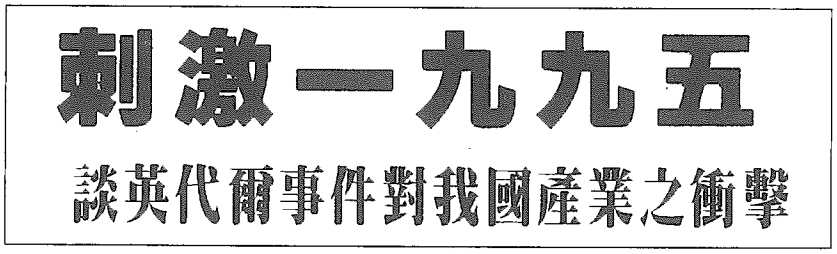
If this value is greater than 0.8, then this region is determined to contain characters and then all the pixels in the region are reset to black or white according to the following rule: if the color of the pixel is close to the color of the character-like components in the region, reset it to black; otherwise, reset it to white. After this conversion, this region is transformed into the white-background/black-text binary format.

5. Experimental results

The proposed approach was implemented on a Pentium/133 PC under the Windows environment. The color document images were acquired by a HP ScanJet IIcx color scanner. Fig. 5 is the processing result of the image shown in Fig. 2. As illustrated, the two text lines originally settled on the graphic background were successfully located and transformed into the white-background/black-text format. The quality of the extracted characters was also good. The processing time of this image was 51 seconds. Figs. 6 and 7 illustrate two more examples: (a) is the original color document image; (b) is the processing result. The text strings were also successfully extracted in these two cases. The processing times of the two images were 45 and 52 seconds, respectively.

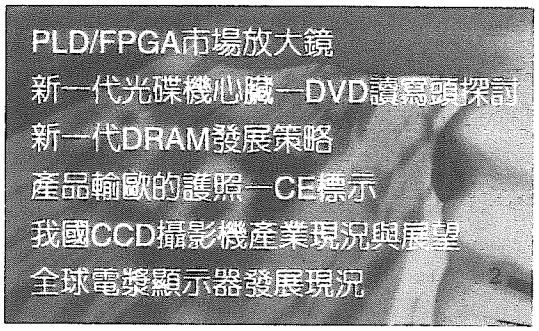
Figure 5. Processing result of the document image shown in Fig. 2.

(a)

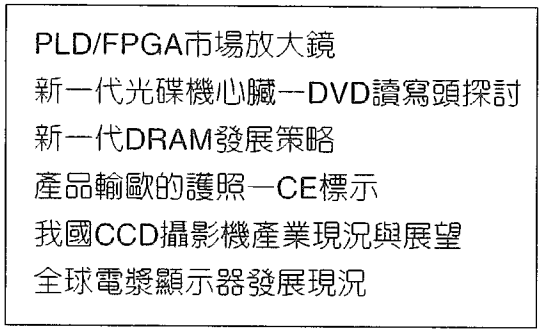


(b)

Figure 6. (a) Color document image with size 2795 X 802 pixels ; (b) processing result.



(a)



(b)

Figure 7. (a) Color document image with size 2187 X 1302 pixels; (b) processing result.

6. Summary and conclusion

In this paper, we present a procedure for separating text from color document images with graphic background. Since in such images, the text string can be any color and moreover the background may be complicated in color and layout, locating text could be difficult. To reduce the complexity for previous processing, our first step is to create a binary edge image for finding out the connected components. Next, we select those components which look like characters or part of characters and then classify them according to their colors. Finally, we group those character-like components which are in the same class and aligned horizontally, and then check if they are really characters. For each identified text string, we further transform it into the white-background/black-text format. Experimental results have shown that this bottom-up processing scheme can effectively extract text from color document images with graphic background.

References

1. Lloyd Alan Fletcher and Rangachar Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, pp. 910-918, 1988.
2. Anil K. Jain and Sushil Bhattacharjee, "Text segmentation using Gabor filters for automatic document processing," *Machine Vision and Application*, 5, pp. 169-184, 1992.
3. Su Liang, M. Ahmadi, and M. Shridhar, "A morphological approach to text string extraction from regular periodic overlapping text/background images," *CVGIP: Graphical Models and Image Processing*, Vol. 56, No. 5, pp. 402-413, 1994.
4. Jun Ohya, Akio Shio, and Shigeru Akamatsu, "A relaxational extracting method for character recognition in scene images," *IEEE Proc. Computer Vision and Pattern Recognition (CVPR'88)*, pp. 424-429, 1988.
5. J. S. Payne, T. J. Stonham, and D. Patel, "Document segmentation using texture analysis," *IEEE Proc. 12th International Conference on Pattern Recognition*, pp. 380-382, 1994.
6. Friedrich M. Wahl, Kwan Y. Wong, and Richard G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing*, Vol. 20, pp. 375-390, 1982.
7. A. A. Zlatopolsky, "Automated document segmentation," *Pattern Recognition Letters*, Vol. 15, No. 7, pp. 699-704, 1994.
8. K. Kubota, O. Iwaki, and H. Arakawa, "Document understanding system," *IEEE Proc. 7th International Conference on Pattern Recognition*, pp. 612-614, 1984.
9. Theo Pavlidis and Jiangying Zhou, "Page segmentation and classification," *CVGIP: Graphical Models and Image Processing*, Vol. 54, No. 6, pp. 484-496, 1992.
10. Yi-Sheng Lin and Wen-Hsiang Tsai, "Image segmentation for color document analysis," *IPPR Conf. on Computer Vision, Graphics, and Image Processing*, Nan-Tou, Taiwan, R.O.C., pp. 135-142, 1994.
11. Yen-Huei Chuang and Wen-Hsiang Tsai, "Segmentation of texts, graphics, and special components for color document image analysis," *IPPR Conf. on Computer Vision, Graphics, and Image Processing*, Tao-Yuan, Taiwan, R.O.C., pp. 471-478, 1995.
12. Hong-Ming Suen and Jhing-Fa Wang, "Text string extraction from images of color printed documents," *IPPR Conf. on Computer Vision, Graphics, and Image Processing*, Tao-Yuan, Taiwan, R.O.C., pp. 534-541, 1995.
13. Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.
14. Julius T. Tou and Rafael C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, 1974.