Algorithms for compressing compound document images with large text/background overlap

B.-F. Wu, C.-C. Chiu and Y.-L. Chen

Abstract: Two algorithms are presented for compressing image documents, with a high compression ratio for both colour and monochromatic compound document images. The proposed algorithms apply a new method of segmentation to separate the text from the image in a compound document in which the text overlaps the background. The segmentation method classifies document images into three planes: the text plane, the background (non-text) plane and the text's colour plane, each of which are processed using different compression techniques. The text plane is compressed using the pattern matching technique, called JB2. Wavelet transform and zerotree coding are used to compress the background plane and the text's colour plane. Assigning bits for different planes yields high-quality compound document images with both a high compression ratio and well presented text. The proposed algorithms greatly outperform two well known image compression methods, JPEG and DjVu, and enable the effective extraction of the text from a complex background, achieving a high compression ratio for compound document images.

1 Introduction

Separating the text from a compound document image is an important step in analysing a document. Some approaches to processing monochromatic document images have already been proposed [1–5]. Queiroz *et al.* proposed a segmentation algorithm based on block-thresholding, in which the thresholds were obtained by rate-distortion analysis [6].

Some other systems based on *a priori* knowledge of some of the statistical properties of the various blocks [7–11] or on textual analyses [12–14] have also been developed. Suen and Wang [15] considered geometric features and colour information to classify segmented blocks into lines of text and picture components. Hasan and Karam [16] proposed a morphological technique to extract text from images. Sobottka *et al.* [17] presented two methods: a top-down and a bottom-up analysis for extracting text. Several other approaches use the features of wavelet coefficients [18–21] to extract text. All these systems focus on processing document images whose texts do not overlap the complex background.

Digipaper [22] and DjVu [23, 24] are two image compression techniques that are particularly geared towards compressing a colour document image. The basic idea behind Digipaper and DjVu is to separate the text from the background and to use different techniques to compress those components. These methods powerfully extract text characters from a simple or slowly varying background. However, they are insufficiently effective when the background includes sharply varying contours or when it overlaps with text.

IEE Proceedings online no. 20040805

doi: 10.1049/ip-vis:20040805

Paper first received 27th January and in revised form 3rd April 2003. Originally published online 5th November 2004

The authors are with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, Republic of China

This work proposes a new algorithm for segmenting text from a complex compound document. Two compression algorithms with a high compression ratio are also proposed. The segmentation algorithm can separate text from complex document images. Therefore, it has many applications, such as to colour facsimiles and document compression. Moreover, the segmentation algorithm can be used to find characters in complex documents with a large text/background overlap.

2 Text segmentation algorithm

We propose a new segmentation algorithm that can separate text from a complex, colour or monochromatic compound document. One of the test images is shown as Fig. 1. Two phases are used to accomplish the desired purpose. In the first phase, which involves colour transformation and clustering analysis, the monochromatic document image is partitioned into three planes: the dark plane, the medium plane and the bright plane. In the second phase, an adaptive threshold is determined to refine the text by adaptive binarisation and block extraction. The two phases in which the algorithm extracts the text from the background are described below.

2.1 Colour transformation

The colour transformation technique is used to transfer a colour document image to the YUV plane and the Y-plane (greyscale image) is used to segment the text from the complicated background image.

2.2 Clustering analysis

The clustering algorithm is used to split the greyscale images. The greyscale images after clustering analysis are classified into three planes: dark, medium and bright. The text is embedded in one of the planes.

The clustering algorithm is described below:

Step 1. Partition the $M \times N$ greyscale image $A(i, j)$ into p sub-block images $x_n(i,j)$. Each sub-block $x_n(i, j)$ is of $K \times L$, where $n = 1, 2, \ldots p$

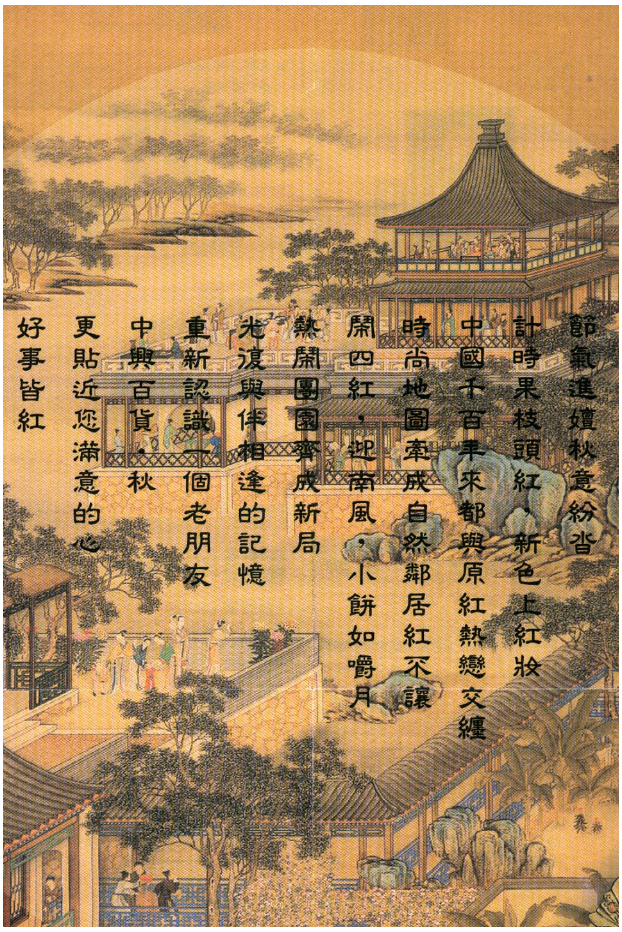

Fig. 1 *Original full image (200 dpi, size = 1024 × 1536)*

Step 2. Calculate the mean of the grey value m and the standard derivation σ of each $K \times L$ sub-block image. For the nth sub-block image $x_n(i,j)$, the mean and standard deviation are

$$m_n = \frac{\sum_{i,j} x_n(i,j)}{K \times L} \tag{1}$$

$$\sigma_n = \sqrt{\frac{\sum_{i,j} [x_n(i,j) - m_n]^2}{K \times L}} \tag{2}$$

Step 3. Split $x_n(i,j)$ according to the mean and standard derivation. Define two centres

$$C'_{n1} \text{ and } C'_{n2}, \text{ by } C'_{n1} = m_n + 0.5 \times \sigma_n \text{ and } C'_{n2} = m_n - 0.5 \times \sigma_n \tag{3}$$

Step 4. Calculate the Euclidean distance from each pixel $x_n(i,j)$ to C'_{n1} and C'_{n2} using

$$D'_{ij,1} = |x_n(i,j) - C'_{n1}| \text{ and } D'_{ij,2} = |x_n(i,j) - C'_{n2}| \tag{4}$$

Then, partition $x_n(i,j)$ into two clusters $\eta_k(k = 1, 2)$

$$\eta_1: \{x_n(i,j) | D'_{ij,1} \le D'_{ij,2}\} \text{ and } \eta_2: \{x_n(i,j) | D'_{ij,1} > D'_{ij,2}\} \tag{5}$$

Step 5. Calculate the mean m_{nk} and the standard derivation σ_{nk} of the two clusters $\eta_k(k = 1, 2)$ using (1) and (2), respectively.

If $\sigma_{n1} > \sigma_{n2}$, then centre $C_{n3} = m_n - 0.5 \times \sigma_n$, and the two new centres C_{n1} and C_{n2} are

$$C_{n1} = m_{n1} + 0.5 \times \sigma_{n1} \text{ and } C_{n2} = m_{n1} - 0.5 \times \sigma_{n1} \tag{6}$$

Else, if $\sigma_{n1} < \sigma_{n2}$, then centre $C_{n3} = m_n + 0.5 \times \sigma_n$ and the two new centres C_{n1} and C_{n2} are

$$C_{n1} = m_{n2} + 0.5 \times \sigma_{n2} \text{ and } C_{n2} = m_{n2} - 0.5 \times \sigma_{n2} \tag{7}$$

Step 6. After step 5, three clustering centres $C_{nk}(k = 1, 2, 3)$ are obtained. Partition $x_n(i,j)$ into three clusters $\psi_k \ (k = 1, 2, 3)$

$$\begin{aligned} &\psi_1: \{x_n(i,j) | D_{ij,1} < D_{ij,2} \text{ and } D_{ij,1} < D_{ij,3}\} \\ &\psi_2: \{x_n(i,j) | D_{ij,2} < D_{ij,1} \text{ and } D_{ij,2} < D_{ij,3}\} \\ &\psi_3: \{x_n(i,j) | D_{ij,3} < D_{ij,1} \text{ and } D_{ij,3} < D_{ij,2}\} \end{aligned} \tag{8}$$

where

$$\begin{aligned} D_{ij,1} &= |x_n(i,j) - C_{n1}| \\ D_{ij,2} &= |x_n(i,j) - C_{n2}| \text{ and} \\ D_{ij,3} &= |x_n(i,j) - C_{n3}| \end{aligned} \tag{9}$$

Repeat steps 2–6 until all of the sub-block images $x_n(i,j) \quad (n = 1, 2, \ldots p)$ have been processed.

The optimal partition of the images depends on the distribution of the intensity of the background images and the lengths, sizes and layouts of the text strings. However, the analysis of these parameters is very complex. Therefore, images are partitioned into equal sub-blocks for simplicity. This study used $K = 256$ and $L = 128$ and a value of p that depended on the image size.

2.3 Adaptive binarisation

After the first phase, the distributions of grey values are simpler than those of the original document image. Then, the text is extracted from the background image using the thresholding algorithm. Thresholding techniques can be categorised into two classes: global and local. This work utilises a local thresholding algorithm. Set the threshold value TH_n of the dark and bright planes of the sub-block images $x_n(i,j)$ $(n = 1, 2, \ldots p)$ to

$$TH_n = m\_f_n \pm \left(\frac{m\_f_n}{m\_b_n - m\_f_n} \right) \times \sigma\_f_n \tag{10}$$

where m_f_n and σ_f_n are the mean value and standard deviation of the dark or bright plane of $x_n(i,j)$ and m_b_n is the mean value of the other planes of $x_n(i,j)$.

Restated, m_f_n is determined by the foreground pixels of the plane that is being processed, while m_b_n relates to the background pixels of the other two planes. From (10), the value of TH_n can be adapted to the grey value of $x_n(i,j)$. The foreground pixels in the dark plane are darker than the background pixels. The adaptive thresholding value can be calculated as

$$TH_n = m\_f_n - \left| \frac{m\_f_n}{m\_b_n - m\_f_n} \right| \times \sigma\_f_n$$

The threshold value TH_n is biased to the left of the mean value m_f_n. The foreground pixels in the bright plane are brighter than the background pixels. The adaptive thresholding value can be calculated as

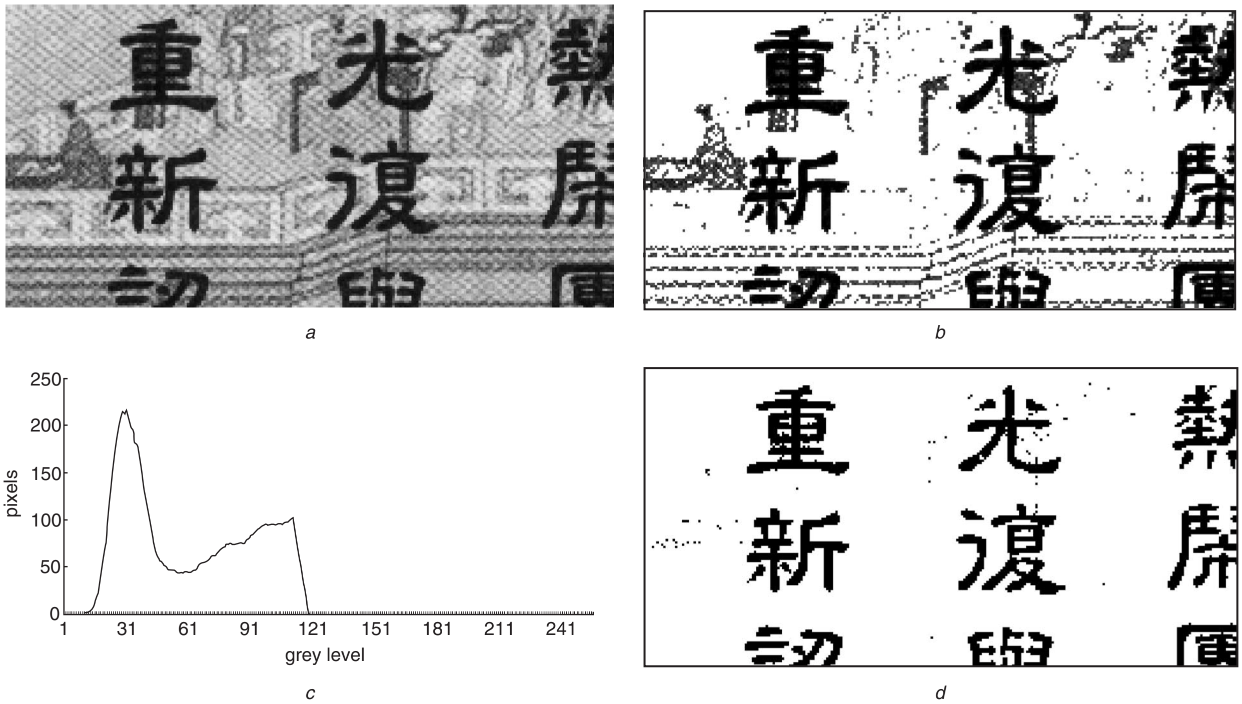

Fig. 2 *Example of text extraction algorithm (size = 256 × 128)*
a Original sub-block image
b Dark plane of the sub-block
c Histogram of image *b*
d Bilevel image after adaptive thresholding

$$TH_n = m\_f_n + \left| \frac{m\_f_n}{m\_b_n - m\_f_n} \right| \times \sigma\_f_n$$

The threshold value TH_n is biased to the right of the mean value m_f_n. Figure 2 gives an example of the algorithm. Figure 2*a* is the sub-block image $x_n(i,j)$, Fig. 2*b* is the dark plane of the sub-block and Fig. 2*c* shows the histogram of the sub-block image $x_n(i,j)$. The threshold value TH_n is biased to the left of the mean value m_f_n to clarify the text. Figure 2*d* shows the bi-level image.

2.4 Spreading and growing the regions of foreground pixels

The pixels of the foreground and noise are obtained simultaneously using the thresholding method. Accordingly, the isolated pixels are deleted and the constrained run length algorithm (CRLA) [1] is applied to remove the noise pixels. The CRLA is applied in horizontal and vertical directions, yielding the binary images $Mh(i, j)$ and $Mv(i, j)$ $(1 < i < M, 1 < j < N)$, respectively. Then, the AND operator is applied to $Mh(i, j)$ and $Mv(i, j)$ pixel-by-pixel, and a binary image $Mhv(i, j)$, which merges the neighbouring pixels in both directions, is obtained.

The CRLA and the logic operation together constitute the spreading process, after which the binary spreading image $Mhv(i, j)$ is processed using the region-growing method to arrange the foreground pixels into rectangular blocks. The region-growing method is described below.
Step 1. Collect the foreground pixels of the image $Mhv(i, j)$ row by row.
Step 2. Compare the foreground pixels collected in step 1 with those in the current blocks. If the foreground pixels and blocks overlap, merge the pixels into the block. Otherwise, establish a new block for the foreground pixels.
Step 3. After the region is grown, check every block. If a block has not grown, regard it as an isolated block.
Step 4. Check whether the blocks overlap. Merge overlapping blocks into a single block.
Step 5. Once $Mhv(i, j)$ reaches the last row, then go to step 6; otherwise return to step 1.
Step 6. Change all existing blocks into isolated blocks. Merge any overlapping blocks into a single block.
Step 7. Delete small noise blocks and stop.
The processes of the spreading and growing of regions yield the positions of the foreground blocks.

2.5 Distinguishing text from foreground blocks

The blocks that contain foreground pixels are extracted following the spreading and growing processes. The blocks that contain text strings must now be identified. In this study, three parameters: transition pixel ratio, foreground pixel ratio and block size: are used to identify these blocks.

The transition pixel ratio is defined as

$$T = \frac{\textit{total number of transition pixels in block}}{\textit{area of block}} \tag{11}$$

The transition pixel is located at the boundary of the foreground pixels. For example, in the following bilevel image:

$$0\,0\,0\,\underline{1}\,\underline{1}\,0\,\underline{1}\,0\,0\,\underline{1}\,1\,\underline{1}\,0\,0\,0$$

the pixels marked '_' are the transition pixels, of which five are present. The foreground pixel ratio is defined as

$$B = \frac{\textit{total number of foreground pixels in block}}{\textit{area of block}} \tag{12}$$

The ratio of the number of transition pixels to the area of the block indicates the complexity of the blocks, and the foreground pixel ratio reflects the density of the foreground pixels. The block size is defined as block width × block length. The T parameter will arise when the text-like features increase. When the block is filled with many speckles, the T parameter is bigger than the T parameter of the text block. We can set an upper bound of the T parameter to delete the blocks filled with many speckles. However, the T parameter is very small and close when a block is filled with a line or a big black/white object. We cannot set the T parameter to delete the unwanted blocks. Therefore, we use the B parameter to remove the blocks filled with a big black/white object. The parameter of block size can be used to remove the noise blocks which are too small or too large. $T \leq 0.3, B \leq 0.5$ and $300 \leq \text{block size} \leq 30000$ in the mask plane are set.

3 Document image compression algorithm

The foreground/background representation was proposed in the ITU MRC/T.44 recommendation [25]. This prototype document image representation is used in Xerox's XIFF image format, which presently uses CCITT-G4 to code the mask (text) layer, and JPEG is used to code the foreground (text colour) and background layers. However, the compression ratio of MRC/T.44 is insufficient for document images. Thus, this foreground/background representation is used and two compression algorithms are proposed for compound document images.

Several gaps appear in the background image when pixels of text are extracted from it. The gaps are replaced by pixels with the average grey value of the neighbouring pixels to improve the efficiency of compression. The colour of the text can be obtained from the original image, according to the position of the text. The pixels of the text are called 'used pixels' and the others are called 'unused pixels'. The unused pixels can be replaced by pixels of an appropriate colour to enhance the compression. The colour-filling algorithm is as follows.

Step 1. Mark the pixels in the mask as used pixels; the other pixels are unused pixels.
Step 2. Fill the gap with the colour of the pixel that adjoins the used pixel, row by row. Mark the filled pixels as used pixels.
Step 3. Fill the gap with the colour of the pixel next to the used pixel, column by column. Mark the filled pixels as used pixels.
Step 4. Repeat the processes in steps 2 and 3 until no unused pixels remain. The foreground plane is thus obtained.

This work proposes two compression algorithms, algorithms I and II, to compress compound document images. Algorithm I uses two compression methods for different planes. Each component of the encoder is described below.

(i) Method for compressing the mask
The mask image compression method uses JB2 [26]. JB2 is an algorithm proposed by AT&T as the JBIG2 standard [27] for compressing fax and bilevel images. JB2 provides better compression than JBIG1 [28], which is often used for faxes, and in both the lossless and the lossy compression of arbitrarily scanned images have scanning resolutions from 100–800 dpi.
(ii) Method for compressing foreground/background
The combination of discrete wavelet transform [29, 30] and zerotree coding [31–33] has been proposed to compress a

a *b*

Fig. 3 *Segmentation images obtained using proposed algorithm I*
a Mask image
b Background image

pure image with a high compression ratio. Such coding algorithms provide good image quality. This study uses the embedded zerotree wavelet (EZW) coding algorithm [31] to compress foreground/background images.

These compression methods are applied to mask, foreground and background images. The document image compression algorithm mentioned earlier is called compression algorithm I.

Compression algorithm I can extract the text plane from an overlapping background image and compress it using JB2. The compressed data thus obtained are approximately 50% of all the compressed data. Therefore, compression algorithm II is proposed to improve the compression ratio.

Compression algorithm II uses a downsampling method to reduce the number of data of the original document images. The downsampling method replaces each 2×2 pixel block by its mean value. The size of the image is thus diminished to a quarter of that of the original. The text segmentation algorithm extracts the text plane. The processing time and the size of the text plane are reduced because the size of the image is a quarter of that of the original. After the segmentation algorithm is applied, the full-size background and quarter-size foreground are compressed using the wavelet-based compression algorithm, and a quarter-size mask that uses JB2.

In the decompression phase, the quarter-size mask is enlarged by upsampling. The upsampling method expands each pixel in the mask into a 2×2 pixel. After upsampling, text looks thinner than in the original image, so the characters are expanded by one pixel around their boundaries.

4 Experimental results

The proposed algorithms were simulated in Windows 2000 (Pentium III 700, 128 MB RAM) using programs written in C++ language. A 24-bit true colour image and 200 dpi processing were used. R, G and B values characterise each pixel in 24-bit true colour, and every value is represented by 8 bits. Two compression algorithms, JPEG and DjVu, are selected for comparison with the proposed algorithms. Figure 3 shows the mask image and the background image obtained using proposed method I.

The test image is processed using algorithm I and JPEG, as shown in Fig. 4. The colour of the image compressed by JPEG is very seriously lost; the block effect is very obvious and the text is blurred. The visual quality obtained using method I is better than that obtained using the JPEG algorithm.

Figure 5 displays the image processed by DjVu. Method I and DjVu use the MRC format to compress, so the test image is divided into three; the mask image, the background image and the reconstruction image, for comparison.

Mask image: From Figs. 3*a* and 5*b*, text extracted by method I is clearer than that extracted by DjVu. The text can be extracted from the complex background using segmentation algorithm I, so the mask image to which algorithm I is applied can be postprocessed, such as by OCR, more precisely than can the mask of DjVu. Accordingly, the segmentation algorithm can also be applied to an OCR system to recognise text on a complex background.

a *b*

Fig. 4 *Comparison of proposed algorithm I and JPEG*

a Proposed algorithm I (mask: 8647 bytes; foreground: 2359 bytes; background: 15728 bytes; compression ratio: 176.7)
b JPEG (compression ratio = 122.4)

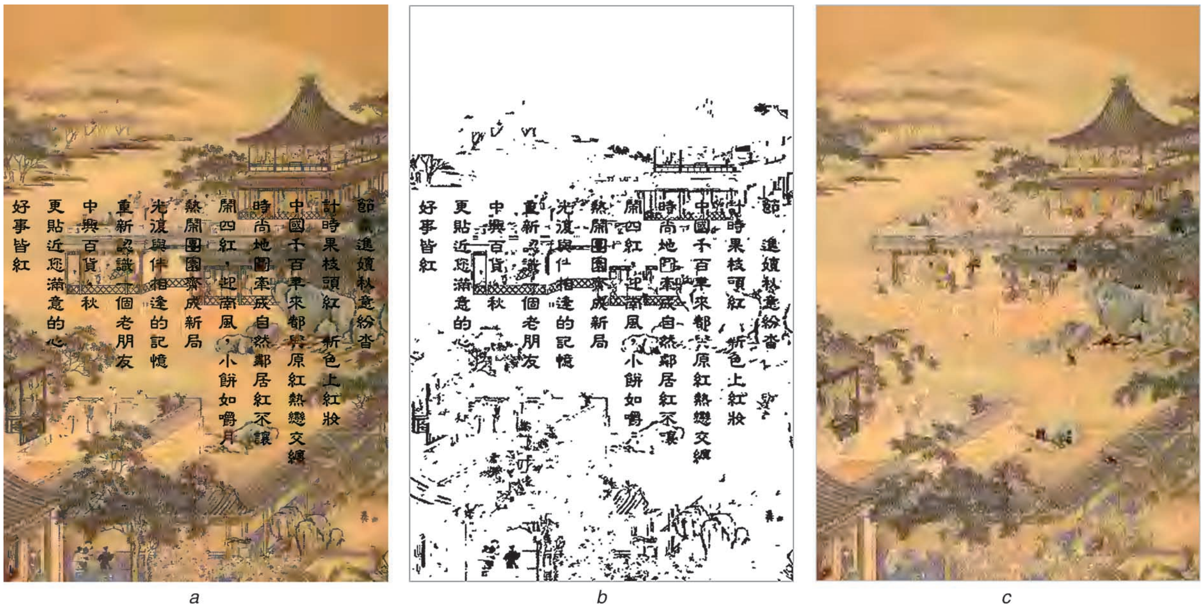

Fig. 5 *Images processed by DjVu*

a Compression ratio = 163
b Mask plane
c Background plane

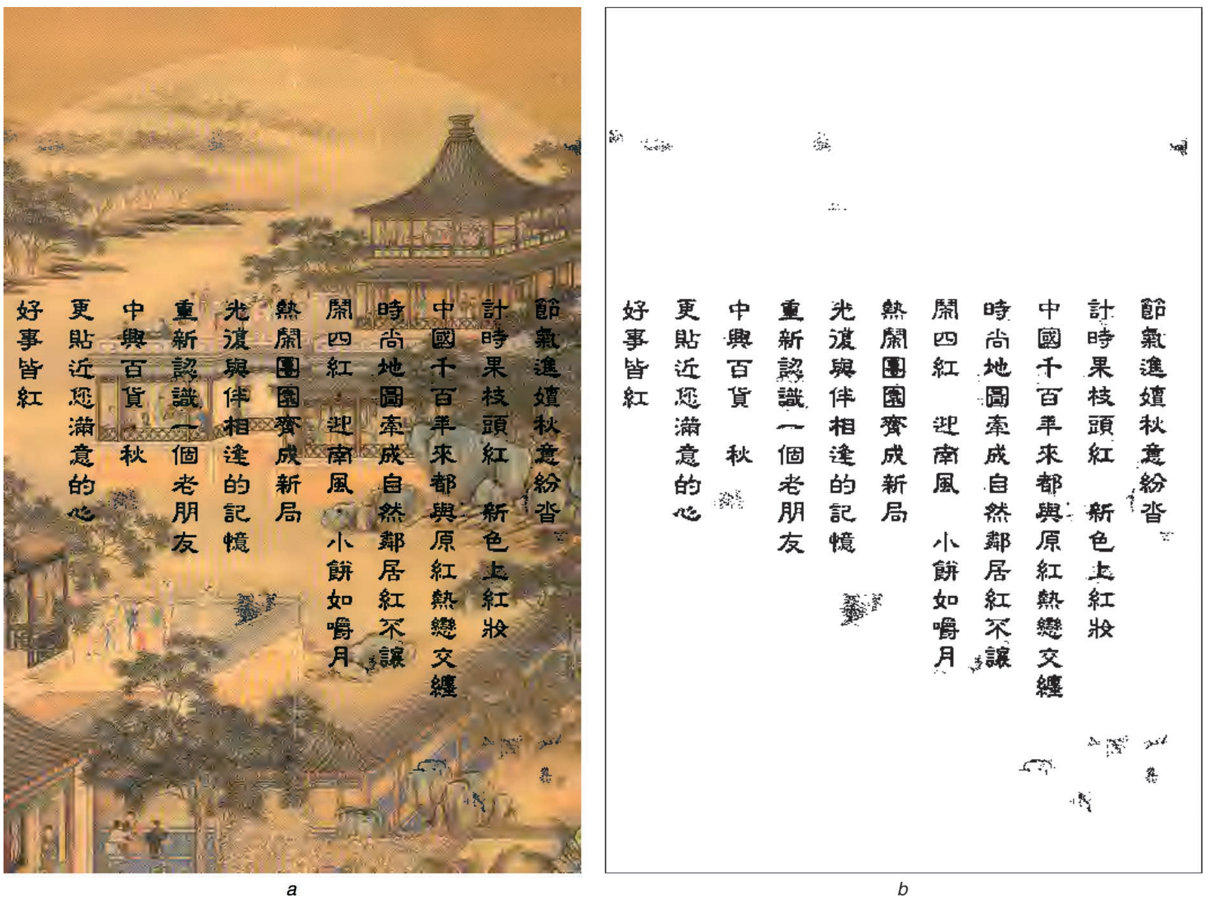

Fig. 6 *Images processed by method II*

a Proposed algorithm II (mask: 4362 bytes; foreground: 590 bytes; background: 15728 bytes; compression ratio: 228.2)
b Mask plane

Background image: Figure 3*b* displays the background image obtained by method I, and Fig. 5*c* shows the background image obtained by DjVu. Although DjVu is especially good for extracting sharp edges, some parts of the text are missing. Clearly, the background image obtained using method I is more precise than that obtained using DjVu.

Reconstruction image: Figure 4*a* displays the reconstruction image obtained by method I, and Fig. 5*a* displays the reconstruction image obtained by DjVu.

Figure 6 shows the reconstruction image and the mask image obtained by method II. The latter is a quarter of the size of that obtained using method I. The mask images

Table 1: Comparison for the proposed methods JPEG and DjVu

	Method I	Method II	JPEG	DjVu
Compression ratio	176.7	228.2	122.4	163
PSNR, dB	18.9	18.8	17.6	18.7
Encoding time, s	12	6	1	3
Decoding time, s	5	4	0.7	1

obtained using algorithm II are not directly downsampled from those obtained using method I, but they are extracted from downsampled document image. Therefore, the processing time and the amount of memory used are reduced.

Table 1 presents the compression ratio, PSNR and coding time of the proposed methods, JPEG and DjVu. Methods I and II yield better quality images than JPEG. Furthermore, the compression ratio and PSNR of methods I and II are higher than those of JPEG. The average PSNR of the proposed methods is close to the average PSNR of DjVu, but the visual quality obtained using the proposed method is better than that obtained using DjVu. The total compression ratio associated with method II is higher than that associated with DjVu.

5 Conclusions

Document image segmentation has been studied for over ten years. Directly extracting text from a complex compound document is difficult because the text overlaps the background. This work proposed a new segmentation method for separating text from compound document images with high text/background overlap. Based on the new segmentation method, two methods for compressing compound document images were presented. High-quality compound document images with both a high compression ratio and well presented text were thus obtained. The proposed compression algorithms were compared with JPEG and DjVu. The proposed methods perform much better.

6 References

1 Wahl, F.M., Wong, K.Y., and Casey, R.G.: 'Block segmentation and text extraction in mixed text/image documents', *Comput. Graph. Image Process.*, 1982, **20**, pp. 375–390

2 Nagy, G., Seth, S.C., and Stoddard, S.D.: 'Document analysis with an expert system'. Proc. Pattern Recognition in Practice II, Amsterdam, Netherlands, 1986, pp. 149–159

3 Fletcher, L.A., and Kasturi, R.: 'A robust algorithm for text string separation from mixed text/graphics images', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1988, **10**, (6), pp. 910–918

4 Kamel, M., and Zhao, A.: 'Extraction of binary character/graphics images from grayscale document images', *CVGIP, Graph. Models Image Process.*, 1993, **55**, (3), pp. 203–217

5 Tsai, W-H.: 'Moment-preserving thresholding: A new approach', *Comput. Vis. Graph. Image Process.*, 1985, **29**, pp. 377–393

6 Queiroz, R.L., Fan, Z., and Tran, T.D.: 'Optimizing block-thresholding segmentation for multilayer compression of compound images', *IEEE Trans. Image Process.*, 2000, **9**, (9), pp. 1461–1471

7 Fisher, J.L., Hinds, S.C., and D'Amato, D.P.: 'A rule-based system for document image segmentation'. Proc. 10th IEEE Int. Conf. on Pattern Recognition, Atlantic City, NJ, USA, 1990, pp. 567–572

8 Akiyama, T., and Hagita, N.: 'Automated entry system for printed documents', *Pattern Recognit.*, 1990, **23**, (11), pp. 1141–1154

9 Shih, F.Y., Chen, S.S., Hung, D.C.D., and Ng, P.A.: 'A document segmentation, classification and recognition system'. Proc. IEEE Int. Conf. on System Integration, Morristown, NJ, USA, 1992, pp. 258–267

10 Pavlidis, T., and Zhou, J.: 'Page segmentation and classification', *CVGIP, Graph. Models Image Process.*, 1992, **54**, (6), pp. 484–496

11 Zlatopolsky, A.A.: 'Automated document segmentation', *Pattern Recognit. Lett.*, 1994, **15**, (7), pp. 699–704

12 Wang, D., and Srihari, S.N.: 'Classification of newspaper image blocks using texture analysis', *Comput. Vis. Graph. Image Process.*, 1989, **47**, pp. 327–352

13 Jain, A.K., and Bhattacharjee, S.: 'Text segmentation using Gabor filters for automatic document processing', *Mach. Vis. Appl.*, 1992, **5**, pp. 169–184

14 Lin, Y., and Srihari, S.N.: 'Document image binarization based on texture features', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1997, **19**, (5), pp. 540–544

15 Suen, H.M., and Wang, J.F.: 'Text string extraction from images of colour-printed documents', *IEE Proc., Vis. Image Signal Process.*, 1996, **143**, (4), pp. 210–216

16 Hasan, Y.M.Y., and Karam, L.J.: 'Morphological text extraction from images', *IEEE Trans. Image Process.*, 2000, **9**, (11), pp. 1978–1983

17 Sobottka, K., Kronenberg, H., Perroud, T., and Bunke, H.: 'Text extraction from colored book and journal covers', *Int. J. Doc. Anal. Recognit.*, 2000, **2**, pp. 163–176

18 Li, J., and Gray, R.M.: 'Context-based multiscale classification of document images using wavelet coefficient distributions', *IEEE Trans. Image Process.*, 2000, **9**, (9), pp. 1604–1616

19 Choi, H., and Baraniuk, R.G.: 'Multiscale image segmentation using wavelet-domain hidden Markov models', *IEEE Trans. Image Process.*, 2001, **10**, (9), pp. 1309–1321

20 Cheng, H., and Bouman, C.A.: 'Multiscale Bayesian segmentation using a trainable context model', *IEEE Trans. Image Process.*, 2001, **10**, (4), pp. 511–524

21 Acharyya, M., and Kundu, M.K.: 'Document image segmentation using wavelet scale-space features', *IEEE Trans. Circuits Syst. Video Technol.*, 2002, **12**, (12), pp. 1117–1127

22 Huttenlocher, D., Felzenszwalb, P., and Rucklidge, W.: 'DigiPaper: A versatile color document image representation'. Proc. IEEE Int. Conf. on Image Processing, Kobe, Japan, 1999, pp. 219–223

23 Haffner, P., Bottou, L., Howard, P.G., Simard, P., Bengo, Y., and Cun, Y.L.: 'Browsing through high quality document images with DjVu'. Proc. IEEE Int. Forum on Research and Technology Advances in Digital Libraries, Santa Barbara, CA, USA, 1998, pp. 309–318

24 Bottou, L., Haffner, P., Howard, P.G., Simard, P., Bengio, Y., and Cun, Y.L.: 'High quality document image compression with DjVu', *J. Electron. Imaging*, 1998, **7**, (3), pp. 410–425

25 'MRC. Mixed raster content mode', ITU Recommendation T.44, 1997

26 Howard, P.G.: 'Text image compression using soft pattern matching', *Comput. J.*, 1997, **40**, (2), pp. 146–156

27 'JBIG Committee FDIS Text', ISO/IEC International Standard 14492, 1999

28 'JBIG. Progressive bi-level image compression', ITU recommendation T.82, ISO/IEC Internation Standard 11544, 1993

29 Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I.: 'Image coding using wavelet transform', *IEEE Trans. Image Process.*, 1992, **1**, (2), pp. 205–220

30 Mallet, S.C.: 'A theory for multiresolution signal decomposition: The wavelet representation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1989, **11**, (7), pp. 674–693

31 Shapiro, J.M.: 'Embedded image coding using zerotrees of wavelets coefficients', *IEEE Trans. Signal Process.*, 1993, **41**, pp. 3445–3462

32 Said, A., and Pearlman, W.A.: 'A new, fast and efficient image codec based on set partitioning in hierarchical trees', *IEEE Trans. Circuits Syst. Video Technol.*, 1996, **6**, (3), pp. 243–250

33 Taubaman, D., and Zakhor, A.: 'Multi-rate 3D subband coding of video', *IEEE Trans. Image Process.*, 1994, **3**, (5), pp. 572–588