International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 19, No. 8 (2005) 997–1025

World Scientific
www.worldscientific.com

MULTI-LAYER SEGMENTATION OF COMPLEX DOCUMENT IMAGES

BING-FEI WU* and YEN-LIN CHEN†

Department of Electrical and Control Engineering, National Chiao Tung University
1001 Ta-Hsueh Road, Hsinchu 300, Taiwan
**bwu@cc.nctu.edu.tw*
†*ylchen@cssp.cn.nctu.edu.tw*

CHUNG-CHENG CHIU

Department of Electrical Engineering, Chung Cheng Institute of Technology
No. 190, Sanyuan 1st Street, Dashi Jen, Taoyuan 335, Taiwan
chiu@ccit.edu.tw

Text is commonly printed on a complex background. Segmenting text is an important part in document analysis. In the past some methods have been shown for the segmentation of texts with images. However, previous studies have not sufficiently addressed complex compound documents. This investigation presents an algorithm for the segmentation of text in various document images. The proposed segmentation algorithm applies a new multilayer segmentation method to separate the text from various compound document images, independent from the text and background overlapping or not. This method solves various problems associated with the complexity of background images. Experimental results obtained using various document images scanned from book covers, advertisements, brochures and magazines, reveal that the proposed algorithm can successfully segment Chinese and English text strings from various backgrounds, regardless of whether the texts are over a simple, slowly varying or rapidly varying background texture.

Keywords: Complex compound document; image segmentation; document analysis; text extraction.

1. Introduction

Rapid advances in multimedia technologies have led to the use of overlapping of texts and background images in various documents, advertisements, brochures and magazines. The complexity of the background images is critical for the application of the text segmentation algorithm. Segmenting texts from a complex compound document image is an important area of document analysis. Document image segmentation, which separates text from a monochromatic background, has been investigated for over a decade. Some systems that are based on prior knowledge of some statistical characteristics of various blocks,[4,12] or texture analyses[8] have been

successively presented. A text segmentation algorithm based on block-thresholding, which involves thresholds on rate-distortion, has been proposed.[11] Also, a system that focuses on extracting and classifying bibliographical information from book covers has been developed.[15] Many other approaches apply the features of wavelet coefficients to extract text.[1–3,6]

All such systems concentrate on processing document images whose texts do not overlay a complex background. These approaches are effective in extracting characters from monochromatic backgrounds. However, they are not applicable to backgrounds with sharply varying contours or that overlap with texts. These background images include (i) monochromatic backgrounds with/without texts; (ii) slowly varying backgrounds with/without texts; (iii) highly varying background with/without texts and, (iv) complex varying backgrounds with/without texts of various colors. Extracting texts is particularly difficult when the compound document image includes a combination, or all of these backgrounds.

A text extraction algorithm has been presented for application to WWW images.[17] This algorithm uses the Euclidean minimum-spanning-tree (EMST) method to cluster the R-G-B space of the input image into a number of color classes. For each color class, the bounding boxes of connected components are obtained using the connected-component labeling method; the shape and features of the bounding boxes are considered to classify the connected components as text-like or nontext-like. However, this global algorithm is not sufficient to extract the texts from various document, advertisement, brochure and magazine images, because texts in these A4-size images are widely distributed. When scanners are used to capture these images, the pixel values of the texts are spread out because of the optical property of the scanners, since the complex varying backgrounds overlap with variously colored text. As a result, the global algorithm fragments the text into each color class.

A global segmentation method is used for color documents[13] by applying R-G-B color spatial information to select the line segments as initial clusters. Then, the number of line segments that are close to each other is reduced and a predefined threshold applied to group the neighbor pixels of the remaining line segments. This method assumes that for the documents under consideration, the background color for each frame is uniform over the whole frame under consideration. Therefore, this method does not apply when images include rich and colorful backgrounds or small texts, because the line segments cannot be correctly selected. Meanwhile, the global algorithm will fragment the texts in each cluster.

Some edge-based methods have been presented for detecting the texts from complex document images.[10,16] These methods use the Sobel or the Canny operator to detect the edge features from the text, and then calculate an edge-based feature to detect the text. The edge-based methods can detect the edge feature and apply it to extract the texts from document images. However, the edge feature of backgrounds will be detected at the same time. Therefore the edge-based method is not applicable when backgrounds have sharply varying contours and overlap with the texts.

The text detection method[14] uses three second-order Gaussian derivative filters to obtain the edge-feature vector from three differently sized images, and applies the K-means algorithm (with $K = 3$) to cluster the pixels based on those edge-feature vectors. One of the three clusters is labeled as the text plane. The text plane contains many complex backgrounds and texture patterns. The refinement phase determines the strokes and edge information about the text plane, and groups the strokes which have similar heights and are aligned horizontally. Additionally, nontext edges interfere with the text plane clustered according to the edge-feature vectors when the texts are connected with the complex texture background. Therefore, the edge-based method is not applicable when backgrounds have sharply varying contours and overlap with the texts.

Some text detection and tracking methods that focus on digital video have been developed recently.[5,7] The text detection approaches use the wavelet transfer or the gradient image of the R-G-B space to obtain the edge information about the images. The text extraction methods are edge-based and multiscale methods. They are sufficient for detecting and tracking texts from video frames. In document images, numerous texts are small and thin. The edge features of small texts and thin strokes diminutive after the down-sampling process. Hence, these methods are inappropriate for detecting the texts from document images. Furthermore, the nontext edges interfere with the edge-based text extraction method when the texts are connected to complex texture backgrounds that have sharply varying contours.

The multilayer segmentation method (MLSM) presented in this paper uses a block-based unsupervised clustering algorithm to cluster the pixels with similar values. The disadvantage of the local method is that much of the structural information is lost. Therefore, a jigsaw-puzzle layer construction algorithm is proposed to reconstruct the structural information based on various objects in the processed images. Hence, the MLSM can be used to solve many of the problems associated with the various document images scanned from book covers, advertisements, brochures and magazines, regardless of whether the texts are over a simple, slowly varying or rapidly varying background texture. The MLSM focuses on the images in the document, and so can overcome the disadvantages of edge-based and down-sampling methods.

This research presents a good method for extracting texts from different compound document images. The compound document image is made up of many objects, including differently colored texts, figures, scenes and complex backgrounds. Any of these objects may overlap each other. They have different features, according to which the document image can be partitioned into many object layers. The MLSM can separate texts from 8-bit grayscale or 24-bit true-color document images, regardless of whether the texts overlay a simple, slowly varying or rapidly varying background.

The rest of this paper is organized as follows. Section 2 introduces the multilayer segmentation method. Section 3 describes a text extraction algorithm. Section 4

presents the experimental results of this study and some discussions. Finally, in Sec. 5 conclusions are drawn.

2. Multilayer Segmentation Method

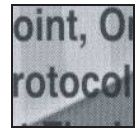
The multilayer segmentation method (MLSM) segments objects from various compound document images in two stages. The different objects in a document image are segmented into various object layers. The first stage of the method is the block-based clustering algorithm, which clusters distinct objects embedded in sub-block images into different "layered-sub-blocks"(LSBs). The second stage applies the jigsaw-puzzle layer construction algorithm to assemble the adjacent LSBs of the same objects into an object layer.

The block-based clustering algorithm partitions the document image into many sub-block images, each of which is classified into different LSBs. Each LSB has one object with a similar value. The texts/objects with similar illumination or color are embedded in one of the LSBs of a sub-block image. Although the block-based clustering algorithm can extract the texts from different backgrounds, the texts in the same paragraph will be divided into many $K \times L$ blocks which are the LSBs of the different sub-block images. The other objects in the document image also have the same problem that the structure information of the objects is destroyed. This is the shortcoming of a local approach. Hence, some statistical and spatial features of adjacent LSBs are introduced into the jigsaw-puzzle layer construction algorithm, in order to assemble all LSBs into a single text paragraph or object, as a jigsaw puzzle. The jigsaw-puzzle layer construction algorithm can recover structural information about different objects. The construction algorithm matches the LSBs of two adjacent sub-block images. The matching process is described in the section of the jigsaw-puzzle layer construction algorithm. Figure 1 shows the results of two adjacent sub-block images after the block-based clustering algorithm has been applied. Figure 1(a) shows a part of test image 5 (in Fig. 9). Figures 1(b) and 1(c) are two sub-block images from Fig. 1(a). Figures 1(d), 1(f) and 1(h) exhibit the LSBs in Fig. 1(b). Figures 1(e), 1(g), 1(i) and 1(j) show the LSBs in Fig. 1(c). It is evident that the sub-block images of Figs. 1(b) and 1(c) have three and four different objects, respectively. After the block-based clustering algorithm has been applied, Figs. 1(b) and 1(c) are segmented automatically into three and four LSBs, respectively.

The MLSM employs the block-based clustering algorithm and the jigsaw-puzzle layer construction algorithm to segment different objects of a document image into distinct object layers. For instance, Figs. 1(d) and 1(e) belong to the same object layer. The different object layers can be obtained after the MLSM has been applied, and the texts with various colors are embedded in different object layers. Then the text line extraction algorithm can collect all of the texts from these object layers, regardless of whether they overlap a monochromatic background or a complex one.

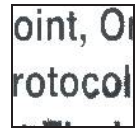
(a) Partial image of test image 5

(b) Sub-block image L (size = 96×96)



(c) Sub-block image R (size = 96×96)

(d) *LSB* of the sub-block image L



(e) *LSB* of the Sub-block image R

(f) *LSB* of the sub-block image L

(g) *LSB* of the sub-block image R

(h) *LSB* of the sub-block image L

(i) *LSB* of the sub-block image R

(j) *LSB* of the sub-block image R

Fig. 1. Example of the results after the block-based clustering algorithm has been applied.

The number of object layers depends on the maximum number of LSBs associated with the sub-block image.

The "joint division factor"(JDF) measures the separability between two adjacent clusters in the block-based clustering algorithm. When the number of clusters

exceeds two, the mean JDF is employed to measure the separability of the clusters. The pool of layered sub-blocks, referred to as the *Pool*, collects all undetermined LSBs. A cluster of determined LSBs constitutes an object layer for further analysis.

2.1. *The block-based clustering algorithm*

A color transformation is applied to transfer a color document image to the YUV space, and the Y-plane (grayscale image) of the original document image is applied to perform segmentation. The Y-plane is used to cluster the objects for two reasons. First, so that the processing speed can be increased, because the Y-plane is one-third of the color image. Second, the method is suitable for application to color and monochromatic images, because many document images are stored monochromatically.

After a color document image has been transformed into a monochromatic one, the converted grayscale image retains the textures of the original color image. However, the difference between the gray value of the texts and that of the overlapping background image may still be small. Therefore, a block-based clustering algorithm is presented to cluster the grayscale sub-block images. The clustering analysis is an unsupervised method for separating text from a sub-block image. The sub-block image is classified into different clusters.

The proposed block-based clustering algorithm is described as follows.

Step 1. Partition the $M \times N$ grayscale image A into q sub-block images $x_n(i,j)$. Each sub-block $x_n(i,j)$ is of size $K \times L$, where $i = 1 - \lceil \frac{M}{K} \rceil$, $j = 1 - \lceil \frac{N}{L} \rceil$ and $n = 1 - \left(\lceil \frac{M}{K} \rceil \times \lceil \frac{N}{L} \rceil \right)$. When the full image A cannot be divided into a whole number of sub-blocks, then the bounding blocks, in the right column and the bottom row will be smaller than the standard sub-block images.

Step 2. Calculate the mean, m, and the standard deviation, σ, for 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 and σ_n.

Step 3. If $\sigma_n < \mathrm{TH}_\sigma$, terminate the clustering process. Else, if $\sigma_n > \mathrm{TH}_\sigma$, then $x_n(i,j)$ is split based on the mean and standard deviations of the processed sub-block image. Define two centers, C'_{n1} and C'_{n2}, as

$$C'_{n1} = m_n + 0.5 \times \sigma_n, \quad \text{and} \quad C'_{n2} = m_n - 0.5 \times \sigma_n. \tag{1}$$

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

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

Then, $x_n(i,j)$ is partitioned into two clusters $\psi_k (k = 1, 2)$ as,

$$\psi_1 : \left\{ x_n(i,j) | D'_{ij,1} \le D'_{ij,2} \right\}, \quad \text{and} \quad \psi_2 : \left\{ x_n(i,j) | D'_{ij,1} > D'_{ij,2} \right\}. \tag{3}$$

Step 5. As stated in Ref. 9, let $\sigma^2_{B1,2}$ and $\sigma^2_{T1,2}$ be the between-class variance and the total variance, respectively. The joint division factor (JDF) between two

adjacent clusters is defined as,

$$\mathrm{JDF}_{1,2} = \frac{\sigma_{B1,2}^2}{\sigma_{T1,2}^2}, \tag{4}$$

$$\sigma_{T1,2}^2 = \sum_{r\in(\psi_1\cup\psi_2)} (r-\mu_{T1,2})^2 P_r, \quad \mu_{T1,2} = \sum_{r\in(\psi_1\cup\psi_2)} rP_r, \tag{5}$$

$$\sigma_{B1,2}^2 = \omega_1\omega_2(\mu_1\mu_2)^2, \quad \omega_1 = \sum_{r\in\psi_1} P_r, \quad \omega_2 = \sum_{r\in\psi_2} P_r, \tag{6}$$

$$\mu_1 = \frac{\sum_{r\in\psi_1} rP_r}{\omega_1}, \quad \mu_2 = \frac{\sum_{r\in\psi_2} rP_r}{\omega_2}, \quad P_r = \frac{n_r}{n}, \tag{7}$$

where n_r is the number of pixels with gray-level r, and n is the total number of pixels in ψ_1 and ψ_2.

Step 6. Calculate the mean m_{nk} and standard deviation σ_{nk} of the two clusters $\psi_k(k=1,2)$.

If $\mathrm{JDF}_{1,2} < \mathrm{TH}_{\mathrm{JDF}}$ and $\sigma_{n\_\max} > \mathrm{TH}_\sigma$, where $\sigma_{n\_\max}$ is the maximum of the $\sigma_{nk}(k=1,2)$, then split the cluster of $\sigma_{n\_\max}$.

Else, terminate the clustering process.

Step 7. Step 6 yields three clustering centers C_{nk} $(k=1,2,3)$. And, $x_n(i,j)$ is partitioned into three clusters ψ_k $(k=1,2,3)$.

Step 8. Calculate the mean, m_{nk}, and the standard deviation, σ_{nk}, of each cluster; and calculate the joint division factors, $\mathrm{JDF}_{1,2}$, $\mathrm{JDF}_{2,3}, \ldots, \mathrm{JDF}_{k-1,k}$ among the clusters ($k=1,2,\ldots,x$ and $x>2$).

If $\sqrt{\frac{\mathrm{JDF}_{1,2}^2+\mathrm{JDF}_{2,3}^2+\cdots+\mathrm{JDF}_{k-1,k}^2}{k-1}} < \mathrm{TH}_{\mathrm{JDF}}$ and $\sigma_{n\_\max} > \mathrm{TH}_\sigma$, then split the cluster of maximum σ_{nk} into two clusters. Repeat Step 8.

Else, terminate the clustering process.

Repeat Steps 2–8 until all of the sub-block images $x_n(i,j)$ have been processed. This work uses $\mathrm{TH}_{\mathrm{JDF}} = 0.9$, $\mathrm{TH}_\sigma = 14$ and $K = L = 96$.

With regards to K and L, smaller sub-block images are segmented with greater detail. The small objects can be segmented more clearly, but at the cost of greater computation time to yield the final result using the Jigsaw-Puzzle Layer Construction Algorithm. Therefore, the maximum values of parameters K and L must be selected such that the objects in document images can be clearly segmented. The maximum values of parameters K and L depend on the size of the smallest texts in the document images. In this work, $K = L = 96$ is determined by conducting experiments that involve numerous image samples, such that almost all objects in the document images are separated. Consequently, all objects are segmented into individual clusters in the order of darkest to lightest.

After all sub-block images are clustered, many clusters are decomposed from each sub-block image. Each cluster has a partial image of its original sub-block image. Each cluster may contain more than one connected region. For instance, two letters, i and j, and many Chinese characters have more than one connected

region. Hence, if a sub-block image is divided into k clusters, it will probably have more than k connected regions. A particular analytical method — the Jigsaw-Puzzle Layer Construction Algorithm — is applied to them. Furthermore, observation of a sub-block image and its resultant clusters generated from a block-based clustering algorithm reveals that the clusters look like "sub-layers" of the original sub-block image. Hence, a cluster is called a "Layered-Sub-Block", or LSB. All LSBs generated in the preceding clustering process are collected into a "*Pool*", to which the jigsaw-puzzle layer construction algorithm is applied.

2.2. *Jigsaw-puzzle layer construction algorithm*

A sub-block image may be composed of one or more object images with various intensity features. Those object images may be parts of a larger object, one or many character patterns with various intensities, or one piece of background texture. The block-based clustering algorithm decomposes the sub-block image into several layered sub-block images, LSBs, ordered from darkest to lightest, corresponding to the original sub-block image. Some statistical and spatial features of the adjacent LSBs are introduced into the jigsaw-puzzle layer construction algorithm to assemble all LSBs of the same text paragraph or object. This section describes an algorithm for constructing the object layers from the LSBs, generated from the block-based clustering algorithm introduced in the preceding section. Before the algorithm is described, some definitions are given.

The term *4-adjacent* refers to the situation in which each LSB has four sides between the adjacent sub-block images that border the top, the bottom, the left or the right side of the LSB. Each side of the LSB may be joined to several adjacent LSBs derived from a single adjacent sub-block image; one of them may match the LSB on the side. An object layer is assembled by the LSBs, by matching adjacent LSBs and by connecting the finite chains of all LSBs. Text strings are mostly printed horizontally or vertically in documents, so the text strings of document images are horizontally or vertically continuous. As a result the *4-adjacent* property can be appropriately considered for evaluating the connectedness of the LSBs. The matchness of the *4-adjacent* LSBs is determined by evaluating their continuity. The pixels of each LSB are represented as a specific subset of all pixels in the corresponding sub-block image. An LSB may consist of several connected regions. The pixels in the connected regions are said to be valid pixels, and the other pixels in the LSB are said to be invalid pixels.

The notation $\mathrm{LSB}(i, j, k)$ means that the kth LSB is decomposed from the sub-block image $x_n(i, j)$. If the $\mathrm{LSB}(i, j, k)$ is matched and belongs to the object layer L_q, then it is represented as $\mathrm{LSB}_q(i, j, k)$, where the subscript q refers to the qth object layer. The valid pixel value at (x, y) in the $\mathrm{LSB}(i, j, k)$ is denoted as Pix(LSB, x, y), $x = 0 - (K - 1)$ and $y = 0 - (L - 1)$. Two measurements of the continuity between two LSBs are defined. First, the average distortion of all touching valid pixels at the boundary between two 4-adjacent LSBs, called side-match distortion, is

represented as D_{SM}. For example, two horizontally adjacent LSBs have the dimensions $K \times L$; the left one is expressed as LSB_l and the right one as LSB_r. Their pixel values on the horizontal touching boundary can be described by $\mathrm{Pix}(\mathrm{LSB}_l, K-1, y)$ and $\mathrm{Pix}(\mathrm{LSB}_r, 0, y)$, $y = 0-(L-1)$. Please note that only the valid pixels are considered, whereas the D_{SM} accounts for only the values of the boundary pixels. The horizontal touching boundaries between the two adjacent LSBs form a vertical side. The valid pixels that are located symmetrically on both boundaries of the vertical side are considered to be establishing a valid side connection. The number of pairs of valid pixels in the valid side connection reflects the connectedness of the two adjacent LSBs, and is given by $N_{vs}(\mathrm{LSB}_l, \mathrm{LSB}_r)$. Hence, the D_{SM} of two horizontally adjacent LSBs is computed as,

$$D_{\mathrm{SM}}(\mathrm{LSB}_l, \mathrm{LSB}_r) = \frac{\sum_{y=0}^{L-1} |\mathrm{Pix}(\mathrm{LSB}_l, K-1, y) - \mathrm{Pix}(\mathrm{LSB}_r, 0, y)|}{N_{vs}(\mathrm{LSB}_l, \mathrm{LSB}_r)}. \tag{8}$$

The D_{SM} is the mean difference between the valid pixels in the valid side connection. If the value of the D_{SM} is small, then the two adjacent LSBs will be the same object layer. The range of D_{SM} values is 0–255.

Second, the difference between the mean values of the two LSBs, called the *inter*-LSB *distortion*, is defined as the D_{LM}. Similarly, only the valid pixels of the LSBs are taken into account for the D_{LM}; the number of valid pixels of a specific LSB is $N_{vp}(\mathrm{LSB})$. The D_{LM} is given by

$$D_{\mathrm{LM}}(\mathrm{LSB}_l, \mathrm{LSB}_r) = |m(\mathrm{LSB}_l) - m(\mathrm{LSB}_r)|, \tag{9}$$

where $m(\mathrm{LSB})$ is the mean of all valid pixels associated with this LSB, and is computed as

$$m(\mathrm{LSB}) = \frac{\sum_{x=0}^{K-1} \sum_{y=0}^{L-1} \mathrm{Pix}(\mathrm{LSB}, x, y)}{N_{vp}(\mathrm{LSB})}. \tag{10}$$

D_{LM} is the mean difference between two LSBs. If the D_{LM} is small, then the two LSBs will be associated with the same object layer. The range of D_{LM} values is 0–255.

Similarly, the D_{SM} and D_{LM} of two vertically adjacent LSBs can be deduced from Eqs. (8) and (9), respectively. A smaller computed value of D_{SM} or D_{LM} corresponds to a stronger continuity or similarity among adjacent LSBs.

Based on the above definitions, the match grade is defined as

$$\text{match grade} = \max(D_{\mathrm{SM}}, D_{\mathrm{LM}}), \tag{11}$$

where the D_{SM} and D_{LM} are calculated from the unclassified $\mathrm{LSB}(i, j, k)$ and the representative $\mathrm{LSB}_q(i', j', k')$. The D_{SM} of Eq. (8) can be regarded as the local mean difference on the adjacent sides of the two LSBs, and D_{LM} can be treated as the global mean difference between the two LSBs. Therefore, the maximum value of D_{SM} and D_{LM} is defined as the match grade. The best matching pair of LSBs can be determined from the minimal match grade.

The jigsaw-puzzle layer construction algorithm is constructed using two procedures — **the decision procedure for constructing a new object layer**, and **the matching procedure**. Table 1 presents the proposed algorithm.

The proposed algorithm begins by analyzing the initially unclassified LSBs in the *Pool*. The *Pool* will be analyzed repeatedly until every unclassified LSB has being associated with a particular object layer. Before beginning a new iteration

Table 1. Psuecode procedure of the jigsaw-puzzle layer construction algorithm.

```
N ← 0                                   (N is current number of existing object layers)
Inital_flag ← 1                         (denote the initialization of layer construction)
while the Pool is not empty do
{
    Decision procedure for constructing of a new object layer
    {
      If Inital_flag = 1
      {
              L_N ← LSB(0,0,0)          (set up and initialize first object layer.)
              N ← 1
        Inital_flag ← 0                 (complete the initialization by constructing first layer)
      }
      Else
      {
          For each unclassified LSB(i,j,k) in the Pool do
          {
              Find the SID, LID and SLD of the unclassified LSB
          }
          If the smallest SID(LSB(i,j,k)) ≤ Th_Sl
          {
              LSB[s], s ≥ 5 ← The unclassified LSBs with the smallest SID
            LSB_min-SLD ← The unclassified LSB with smallest SLD computed
                                from the LSB[s]
              Classify the LSB_min_SLD in its corresponding object layer L_SI
              Remove the LSB_min_SLD from the pool
          }
          Else
          {
              LSB_max-LID ← The LSB with the maximum LID
            L_N ← LSB_max-LID
              N ← N + 1
              Remove the LSB_max−LID from the Pool
          }
    }
}
Matching procedure
{
     for each unclassified LSB(i,j,k) in the Pool do
     {
        for each existing object layers L_N do
        {
            for each 4-adjacent neighboring LSB_N, all the LSB_N ∈ L_N,
              of the unclassified LSB(i,j,k) do
            {
```

Table 1. (*Continued*)

```
                    if the LSB_N satisfy the pre-match condition
                    {
                        Label the LSB_N as the representative LSB of the object layer
                }
                }

                    if more than one representative LSB is involved in this process
                    {
                        LSBq ← the representative LSB with the minimal match grade
                        Found_flag ← 1
                    }
                    else if one representative LSB is involved in this process
                    {
                        LSB'_q ← the representative LSB
                        Representative_flag ← 1
                    }
                    else
                    {
                        Representative_flag ← 0
                    }
                    if Representative_flag = 1
                    {
                    candidate_insert(LSB'_q)
                }
                }
                candidate_decide( ) → L_w
                {
                    L_w ← LSB(i, j, k)
                }
            }
        }
    }
```

of analyzing the unclassified LSBs in the *Pool*, the algorithm conducts a procedure called the "decision procedure for constructing a new object layer", which will be described later, to determine whether a chosen seeded LSB should establish a new object layer or whether it is associated with an existing object layer similar to the chosen seeded LSB. Then, the "matching procedure" finds the matched object layers of the unclassified LSBs. After an unclassified LSB in the *Pool* has been classified into an object layer, it is then removed from the *Pool*.

When the *Pool* is first analyzed, the first unclassified LSB$(0, 0, 0)$ should establish a new object layer, since initially there is no object layer present. Accordingly, the LSB$(0, 0, 0)$ becomes a new object layer L_0, and is represented as LSB$_0(0, 0, 0)$ in the decision procedure for constructing a new object layer. In the matching procedure, all of the remaining unclassified LSBs in the *Pool* are scanned. Whenever an unclassified LSB(i, j, k) that is *4-adjacent* to one or more existing object layers is detected, the prematch condition is tested to determine the reasonable object layers for the unclassified LSB(i, j, k).

The prematch condition is defined as $D_{\mathrm{LM}}(\mathrm{LSB}(i,j,k), \mathrm{LSB}_q(i',j',k')) \leq \mathrm{Th}_{\mathrm{LM}}$, where $\mathrm{Th}_{\mathrm{LM}} = 14$ is a pre-defined threshold. When the condition is met, the object layer L_q becomes a candidate for the unclassified $\mathrm{LSB}(i,j,k)$, and the representative $\mathrm{LSB}_q(i',j',k')$ of the L_q is involved in the measure of determining the match grade. The purpose of prematch is to filter out the unreasonable object layers so as to eliminate the useless process of matching a grade that is incapable of being matched with the $\mathrm{LSB}(i,j,k)$, and thereby reduce the required computational power. All representative LSBs of the reasonable object layers are identified and inserted into the candidate list. It is worth noting that if there are two or more LSBs in a single object layer L_q which are *4-adjacent* to the unclassified $\mathrm{LSB}(i,j,k)$ and meet the prematch condition, then the one with the minimal match grade is chosen as the representative $\mathrm{LSB}_q(i',j',k')$ of the object layer L_q. After all the representatives of the object layers have been obtained, the match grades of the unclassified $\mathrm{LSB}(i,j,k)$ are compared with all the representatives in the candidate list, and then the best match representative LSB_w is determined by choosing the one with the minimal match grade; the unclassified $\mathrm{LSB}(i,j,k)$ are classified into the L_w accordingly.

Now let's again consider the matching procedure. After the L_0 has been established, one object layer exists now. The $\mathrm{LSB}(1,0,0)$ is analyzed; so let's assume that the $\mathrm{LSB}_0(0,0,0)$ is *4-adjacent* and satisfies the prematch condition with $\mathrm{LSB}(1,0,0)$. Only the $\mathrm{LSB}_0(0,0,0)$ is present in the L_0, so the $\mathrm{LSB}_0(0,0,0)$ is selected as the representative LSB of the L_0, and is inserted into the candidate list. No other object layer exists at this time, so the L_0 is automatically determined to be the best match object layer, and consequently $\mathrm{LSB}(1,0,0)$ is classified into L_0 and removed from the *Pool*. Then, the two procedures are repeated until all unclassified LSBs in the *Pool* have been analyzed once in this iteration. The two procedures are described in detail below.

2.2.1. *Decision procedure for constructing a new object layer*

The procedure finds a selected LSB to (i) establish and initialize a new object layer, or (ii) classify it into the existing object layer most similar to the selected LSB. The decision procedure is implemented in order to reach an optimum decision for constructing or extending an object layer. The decision is made according to the analysis of the following characteristics, and is depicted in Fig. 2.

Several definitions and measures must be explicated before this procedure is detailed. The minimum gray intensity distance between one unclassified $\mathrm{LSB}(i,j,k)$ and the object layer L_p, $\mathrm{ID}(\mathrm{LSB}(i,j,k), L_p)$, is determined by the minimum intensity difference between the unclassified $\mathrm{LSB}(i,j,k)$ and the nearest $\mathrm{LSB}_p(i',j',k')$ of all LSBs that belong to L_p; and is determined as

$$\mathrm{ID}(\mathrm{LSB}(i,j,k), L_p) = \min_{\forall \mathrm{LSB}_p \in L_p} D_{\mathrm{LM}}(\mathrm{LSB}(i,j,k), \mathrm{LSB}_p(i',j',k')), \tag{12}$$

where D_{LM} is defined in Eq. (9).

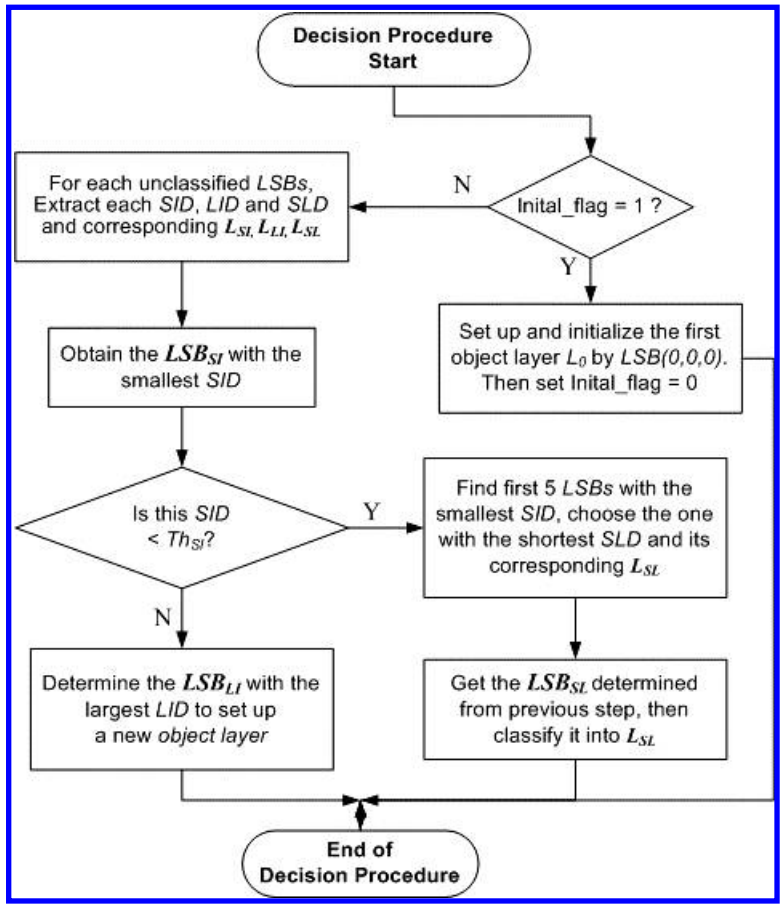


Fig. 2. Flowchart of the decision procedure for constructing or extending an object layer.

The Euclidean distance between one unclassified $\mathrm{LSB}(i,j,k)$ and the object layer L_p, given by $\mathrm{LD}(\mathrm{LSB}(i,j,k),L_p)$, is computed by the Euclidean distance between the unclassified $\mathrm{LSB}(i,j,k)$ and the closest LSB_p of all LSBs that belong to L_p. It is determined as

$$\mathrm{LD}(\mathrm{LSB}(i,j,k),L_p) = \min_{\forall \mathrm{LSB}_p \in L_p} D_l(\mathrm{LSB}(i,j,k),\mathrm{LSB}_p(i',j',k')) \tag{13}$$

$$\text{where } D_l(\mathrm{LSB}(i,j,k),\mathrm{LSB}_p(i',j',k')) = \sqrt{(i-i')^2+(j-j')^2)}. \tag{14}$$

The smallest gray intensity distance between an unclassified $\mathrm{LSB}(i,j,k)$ and all currently existing object layers is defined as

$$\mathrm{SID}(\mathrm{LSB}(i,j,k)) = \min_{\forall n}(\mathrm{ID}(\mathrm{LSB}(i,j,k),L_n)), \tag{15}$$

where n is the index of existing object layers. The object layer with the shortest gray intensity distance determined by the $\mathrm{SID}(\mathrm{LSB}(i,j,k))$ is given by L_{SI}, and its LSB

with the minimum gray intensity distance obtained from the $\mathrm{ID}(\mathrm{LSB}(i,j,k), L_{\mathrm{SI}})$ is $\mathrm{LSB}_{\mathrm{SI}}$. $\mathrm{SID}(\mathrm{LSB}(i,j,k))$ is the smallest difference in the gray intensity between the unclassified $\mathrm{LSB}(i,j,k)$ and all existing object layers. If the $\mathrm{SID}(\mathrm{LSB}(i,j,k))$ value of an unclassified $\mathrm{LSB}(i,j,k)$ is very small, then the gray intensity of the unclassified $\mathrm{LSB}(i,j,k)$ is very similar to the gray intensity of the $\mathrm{LSB}_{\mathrm{SI}}$ in the object layer L_{SI}. The regions of some text blocks or certain homogeneous objects have similar gray intensity, so the unclassified $\mathrm{LSB}(i,j,k)$ may be part of the object layer L_{SI}. The unclassified $\mathrm{LSB}(i,j,k)$ should not establish a new object layer so as to prevent a certain text block or homogeneous object with similar gray intensity to split into more than one object layer.

The greatest gray intensity distance between an unclassified $\mathrm{LSB}(i,j,k)$ and all currently existing object layers is defined as

$$\mathrm{LID}(\mathrm{LSB}(i,j,k)) = \max_{\forall n}(\mathrm{ID}(\mathrm{LSB}(i,j,k), L_n)). \tag{16}$$

The object layer with the greatest gray intensity distance determined by $\mathrm{LID}(\mathrm{LSB}(i,j,k))$ is represented as L_{LI}. $\mathrm{LID}(\mathrm{LSB}(i,j,k))$ is the largest difference between the gray intensity of the unclassified $\mathrm{LSB}(i,j,k)$ and that of all existing object layers. If the $\mathrm{SID}(\mathrm{LSB}(i,j,k))$ values of all unclassified $\mathrm{LSB}(i,j,k)$ are very large then the unclassified $\mathrm{LSB}(i,j,k)$ are dissimilar to the existing object layers. Hence, the unclassified $\mathrm{LSB}(i,j,k)$ which has the largest $\mathrm{LID}(\mathrm{LSB}(i,j,k))$ should be selected as the seeded LSB to establish a new object layer.

The minimum Euclidean distance measured between an unclassified $\mathrm{LSB}(i,j,k)$ and all currently existing object layers is defined as

$$\mathrm{SLD}(\mathrm{LSB}(i,j,k)) = \min_{\forall n}(\mathrm{LD}(\mathrm{LSB}(i,j,k), L_n)), \tag{17}$$

and the object layer with minimum Euclidean distance as determined by the $\mathrm{SLD}(\mathrm{LSB}(i,j,k))$ is L_{SL}. $\mathrm{SLD}(\mathrm{LSB}(i,j,k))$ is the minimum Euclidean distance between an unclassified $\mathrm{LSB}(i,j,k)$ and all existing object layers.

In this procedure, all unclassified LSBs in the *Pool* are processed to extract their corresponding SIDs, LIDs and SLDs to all existing object layers, according to the definitions given above. The unclassified LSB, with a similar gray intensity and which is closest to its corresponding object layer, is selected according to the SID and SLD values. No more than five unclassified LSBs with the least SID values are selected, and they must satisfy the condition — $\mathrm{SID}(\mathrm{LSB}(i,j,k)) \leq \mathrm{Th}_{\mathrm{SI}}$. The SLDs between the five unclassified LSBs and their corresponding L_{SL} are computed. The texts or the homogeneous objects may contain several connected regions, therefore each unclassified LSB should be part of the corresponding L_{SI}. Then, the unclassified LSB with the smallest SLD becomes the selected LSB and is classified into its corresponding L_{SL}.

If the SID values of all unclassified LSBs exceed $\mathrm{Th}_{\mathrm{SI}}$, $\mathrm{SID}(\mathrm{LSB}(i,j,k)) > \mathrm{Th}_{\mathrm{SI}}$, so that none of the unclassified LSBs are similar to any of the existing object layers, then establishing a new object layer by determining the seeded LSB with the largest LID from the *Pool* is appropriate for initializing a new object layer.

The $\mathrm{Th_{SI}}$ is the predefined threshold, and is set to 14. The setting of the $\mathrm{Th_{SI}}$ value will influence the number of the resultant object layers. If it is too small, then the number of object layers will increase, and some homogeneous region may be split into more than one object layer, such as broken text lines, whereas if the value is too large, then different object regions may be merged into the same object layer.

2.2.2. *Matching procedure*

The matching procedure that assigns each unclassified LSB into the existing object layer is as follows. It analyzes the unclassified LSBs from darkest to lightest, left to right, and top to bottom. All unclassified LSBs are put in a "*Pool*", and are analyzed in the order described above.

The algorithm utilizes a list to track the representative LSBs of the object layers and to determine to which object layer the unclassified LSB should belong. The representative LSB_q of the object layer L_q must be one of the LSBs that are in the L_q and *4-adjacent* to the current unclassified LSB. When an unclassified LSB is analyzed to determine which object layer is the best match for it, a choice can be made from any of the object layers. The list stores the representative LSBs of the candidate object layers, each of which provides one representative LSB. Then, the match grades between the unclassified LSB and all representative LSBs, which are *4-adjacent* to the current unclassified LSB, can be calculated to determine which object layer matches best for the unclassified LSB. The match grade is the criterion used to calculate how well the unclassified $\mathrm{LSB}(i, j, k)$ matches a candidate object layer L_q. Before the match grade is computed, the prematch condition is applied to determine whether $\mathrm{LSB}_q(i', j', k')$ representing object layer L_q is a candidate unclassified $\mathrm{LSB}(i, j, k)$.

The match grade is determined by analyzing the D_{LM} and D_{SM} values of the unclassified $\mathrm{LSB}(i, j, k)$ and the representative $\mathrm{LSB}_q(i', j', k')$. Some noise pixels affect the valid pixels in the valid side connection. As a result the D_{SM} may be invalid when the D_{SM} value is small. Therefore, the D_{SM} is determined in two cases. (1) When N_{vs} is large enough to reflect the suitability of the side information of the two adjacent LSBs, the D_{SM} is taken into consideration for the match grade, such that

$$N_{vs}(\mathrm{LSB}(i, j, k), \mathrm{LSB}_q(i', j', k') \geq \mathrm{Th}_{vs}, \tag{18}$$

where Th_{vs} is a predefined threshold. (2) Otherwise, the D_{SM} factor is disabled by setting D_{SM} to zero, in the "max" operation applied in Eq. (11). In case the two adjacent LSBs include character patterns with thin strokes across their sides, Th_{vs} can be reasonably defined as 5% of the K or L values. $K = L = 96$ is used experimentally, as described before, so $\mathrm{Th}_{vs} = 5$ is obtained herein.

The two operations to be applied to the candidate list are defined as;

(i) *candidate_insert* ($\mathrm{LSB}_q(i', j', k')$): inserts a representative $\mathrm{LSB}_q(i', j', k')$ of the object layer L_q into the candidate list.

(ii) *candidate_decide*( )→ L_w: computes the match grades of all representative $\mathrm{LSB}_q(i', j', k')$ in the candidate list, and then determines the best match object layer L_w with the minimal match grade among all candidates. Accordingly, the current unclassified $\mathrm{LSB}(i, j, k)$ is classified into the object layer L_w.

After the proposed MLSM algorithm has been executed, all LSBs are classified into appropriate object layers. Consequently, N object layers, $L_0, L_1, \ldots, L_{N-1}$ are generated. Each object layer has a set of LSBs. An object image is created from all pixels in the object layer. Figure 3(a) shows the image of a CD cover. Figure 3(b) depicts the divided $M \times N$ sub-blocks corresponding to Fig. 3(a). Figures 3(c)–3(e) present the object images obtained from Fig. 3(a) after the MLSM algorithm has been applied. The object images in which all character patterns, foreground objects and background components are well separated, can easily be analyzed in detail. The following section discusses the extraction of the text-lines from each object layer using the text extraction algorithm.

3. Text Extraction Algorithm

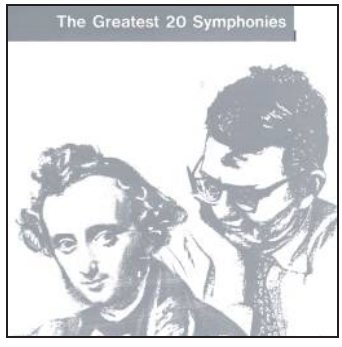
After the MLSM is implemented, the entire document image is decomposed into various object layers. Each object layer may include considerable information about characters, foreground objects, background textures or some other objects. Each object layer will be binarized by setting the valid pixels in the object layer to "1" and setting the invalid pixels to "0". Notably, this study focues on horizontal text lines. The bounding boxes of all connected-components are extracted by the connected-component extraction step. The blocks of characters must be identified and organized into text lines or text regions. The connected-component-based projection profile method is applied to separate all bounding boxes into various "general text lines", GTLs. Each GTL includes a group of bounding boxes.

The definitions used in the text extraction algorithm are as follows.

(a) CC_i is the ith connected-component of the binarized object layer.
(b) CG is a group of connected-components, $\mathrm{CG} = \{\mathrm{CC}_i, i = 0, 1, 2, \ldots, p\}$, and its bounding box is the union of the bounding boxes of all CCs belonging to this CG. Accordingly, the following definitions, which utilize the bounding box of CC, are similar to those based on that of the CG. In the notation in (c)–(f), the term CC can be directly substituted for CG, because both these terms can be applied to their bounding boxes. Similar applications of such notation involving CG are therefore not described.
(c) The bounding box of CC_i has top, left, bottom and right coordinates denoted by $t(\mathrm{CC}_i)$, $l(\mathrm{CC}_i)$, $b(\mathrm{CC}_i)$ and $r(\mathrm{CC}_i)$, respectively, where $t(\mathrm{CC}_i) < b(\mathrm{CC}_i)$ and $l(\mathrm{CC}_i) < r(\mathrm{CC}_i)$.
(d) The width and height of the bounding box of CC_i are $W(\mathrm{CC}_i)$ and $H(\mathrm{CC}_i)$, respectively.

(a) Original image

(b) Sub-block images that correspond to the original image



(c) Object layer 1

(d) Object layer 2

(e) Object layer 3

Fig. 3. Example of the MLSM (image size = 1361 × 1333).

(e) The horizontal and vertical distances between two bounding boxes are defined as

$$D_h(\mathrm{CC}_i, \mathrm{CC}_j) = \max[l(\mathrm{CC}_i), l(\mathrm{CC}_j)] - \min[r(\mathrm{CC}_i), r(\mathrm{CC}_j)], \quad (19)$$

and $$D_v(\mathrm{CC}_i, \mathrm{CC}_j) = \min[b(\mathrm{CC}_i), b(\mathrm{CC}_j)] - \max[t(\mathrm{CC}_i), t(\mathrm{CC}_j)]. \quad (20)$$

If the two bounding boxes are overlapping in the horizontal or vertical direction, then the value of $D_h(\mathrm{CC}_i, \mathrm{CC}_j)$ or $D_v(\mathrm{CC}_i, \mathrm{CC}_j)$ will be negative.

(f) The measures of overlap between horizontal and vertical projections of the two bounding boxes are defined as

$$P_h(\mathrm{CC}_i, \mathrm{CC}_j) = -D_h(\mathrm{CC}_i, \mathrm{CC}_j)/\min[W(\mathrm{CC}_i), W(\mathrm{CC}_j)] \tag{21}$$

$$\text{and} \quad P_v(\mathrm{CC}_i, \mathrm{CC}_j) = -D_v(\mathrm{CC}_i, \mathrm{CC}_j)/\min[H(\mathrm{CC}_i), H(\mathrm{CC}_j)]. \tag{22}$$

Based on the functions and notation as defined above, the text extraction method is detailed as follows. The method comprises two procedures.

The horizontal segmentation procedure $\boldsymbol{H}$**-seg**$(\mathbf{CG}_{in})$ (where the subscript "*in*" refers to "the input CG") is applied as follows:

(1) Project all the bounding boxes of the CCs in the CG_{in} horizontally onto the vertical y-axis.
(2) Sort all the CCs in the CG_{in} according to their corresponding $t(\mathrm{CC}_i)$, where all $\mathrm{CC}_i \in \mathrm{CG}_{in}$. Then, scan the horizontal projections of these CCs on the y-axis and determine the "shadow segments" of these CCs on the y-axis. CCs that are said to share the same shadow segment must have bounding boxes that overlap on the y-axis when horizontally projected, and can be detected when $P_v(\mathrm{CC}_i, \mathrm{CC}_j) > 0$.
(3) For each shadow segment, group the CCs covered by the same shadow segment into an individual CG.
(4) After the above steps are performed, many CGs will have been generated; these are CG_K, where $K = 0, 1, 2, \ldots, k-1$. For each CG_K, implement the vertical segmentation procedure V-seg(CG_K).

The vertical segmentation procedure $\boldsymbol{V}$**-seg**$(\mathbf{CG}_K)$ is conducted as follows:

(1) Project all the bounding boxes of CCs of the CG_K vertically onto the x-axis.
(2) Sort all of the CCs in the CG_K according to their corresponding $l(\mathrm{CC}_i)$, where all $\mathrm{CC}_i \in \mathrm{CG}_K$. Then, scan the vertical projections of these CCs onto the x-axis and determine their shadow segments. The CCs whose vertical projections on the x-axis share a single shadow segment can be identified when $P_h(\mathrm{CC}_i, \mathrm{CC}_j) > 0$.
(3) Group the CCs that are covered by the same shadow segment into an individual CG. Repeat for each shadow segment.
(4) Determine the two merging conditions of the adjacent CGs, CG_{K1} and CG_{K2}, which are (i) whether the horizontal space between the two adjacent CGs is sufficiently small; that is

$$D_h(\mathrm{CG}_{K1}, \mathrm{CG}_{K2}) < \max(\mathrm{Avg\_}W(CG_{K1}), \mathrm{Avg\_}W(CG_{K2})), \tag{23}$$

where Avg_W(CG) is the average width of all CCs that belong to this CG; (ii) whether the average heights of the CCs that belong to the two CGs are similar,

to determine whether the ratio of the two average heights should be within a reasonable range

$$0.67 \leq \text{Avg\_}H(\text{CG}_{K1})/\text{Avg\_}H(\text{CG}_{K2}) \leq 1.5. \tag{24}$$

If the foregoing two conditions are met, then merge the two adjacent CGs.

(5) After the above steps have been implemented, many CGs of CCs will have been produced; these are CG_L, where $L = 0, 1, 2, \ldots, l-1$. If only one resultant CG_0 is obtained, then terminate the segmentation procedure; otherwise, for each CG_L, perform $H - \text{seg}(\text{CG}_L)$.

As stated above, the text extraction procedure is applied to all CCs when a particular object image is being processed by recursive segmentation, using H-seg and V-seg procedures. The sets of all CCs extracted from the processed object image are defined as the CGs; Fig. 4 presents the text extraction algorithm. The resultant CGs are the "general text lines", or GTLs. Figure 4(a) shows the result of applying the H-seg procedure to the CCs of Fig. 3(e): five CGs are obtained. The five CGs are then in turn applied to the V-seg procedure. The first CG is considered as an example, and the results are shown in Fig. 4(b). Then, the CGs obtained from the V-seg procedure are divided into two CGs, according to condition (24). The two CGs derived from the V-seg procedure undergo the H-seg procedure; neither can be divided into more CG. Therefore, the two CG are the resultant GTL and are obtained by applying the text-line decision rules.

A set of knowledge-based decision rules are provided to determine whether each GTL is a text line or a nontext region. If one GTL satisfies the rules required for a text-line, it is identified as a text-line. The shape and content of the bounding box are determined from the GTL the features of the text line, including the transition pixel ratio, the foreground pixel density ratio and the block size. A "1" represents a valid pixel and a "0" represents an invalid pixel. The transition pixel is at the boundary of the foreground pixels. For instance, in the following bi-level image,

0	0	0	1	1	0	1	0	0	1	1	1	0	0	0
			▲	▲		▲			▲		▲			

the pixels marked "▲" are the transition pixels, of which five are present.

The horizontal transition pixel ratio of the GTL block is,

$$T_h = \frac{\text{Total number of the transition pixels of the GTL}}{\text{Col}_N}, \tag{25}$$

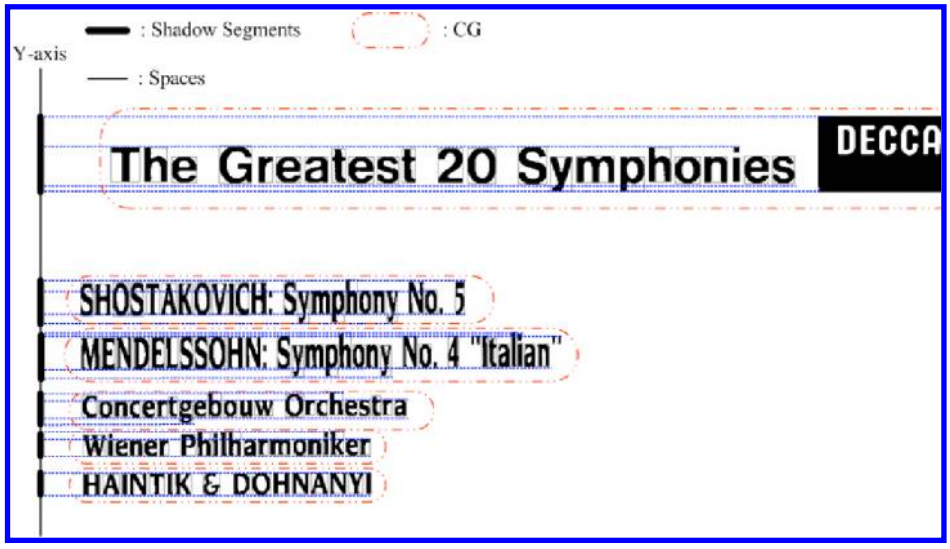
where Col_N is the number of the pixel columns in which the valid pixels are present.

The valid pixel density of the GTL is defined as,

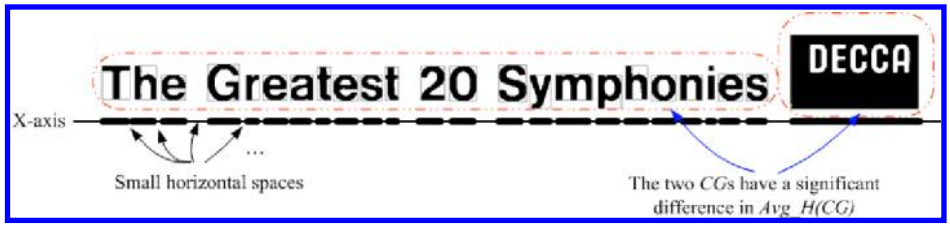
$$B = \frac{\text{Total number of valid pixels of the GTL}}{A}, \tag{26}$$

where A is the area of the bounding box of the GTL.

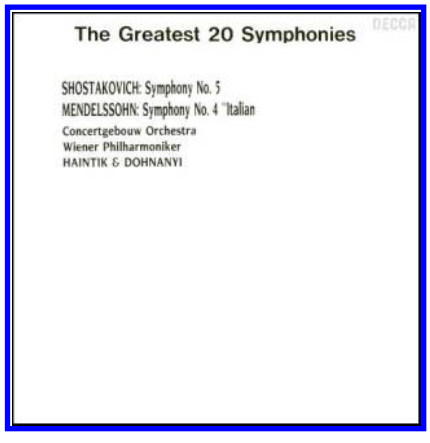
The width and height of the bounding box of the GLT, and the number of the CCs that belong to the GTL are W, H and N_c, respectively.



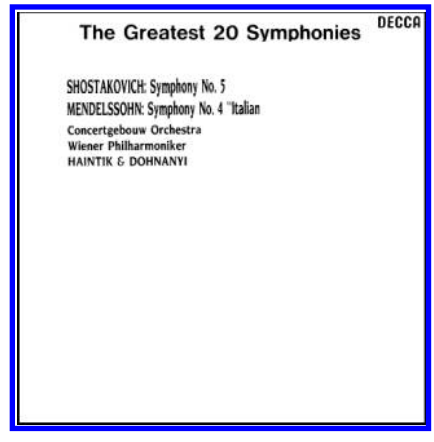
(a) Result of applying the *H-seg* procedure on the *CC*s in Fig. 3(e)



(b) Result of the *V-seg* procedure applied to the first *CG* in Fig. 4(a)



(c) Text plane following application of the text extraction algorithm



(d) Binary image of the text plane

Fig. 4. Example of the text extraction algorithm.

Based on these features as defined above, a GTL block is identified as a text-line block if all the following decision rules are satisfied.

$$\text{(i)} \quad 1.1 < T_h < 4.0 \tag{27}$$

$$\text{(ii)} \quad 0.2 < B < 0.7 \tag{28}$$

$$\text{(iii)} \quad W/H \geq 2.0 \tag{29}$$

$$\text{(iv)} \quad 0.5(W/H) \leq Nc \leq 8.0(W/H) \quad \text{and} \quad 2 \leq Nc \tag{30}$$

$$\text{(v)} \quad \sum\nolimits_i A_i/A \geq 0.4, \text{ where } A_i \text{ is the area of the } i\text{th CC of the GTL.} \tag{31}$$

The ratio of the number of transition pixels used under condition (i) is used to evaluate the complexity of the area of the GTL. The valid pixel density in condition (ii) measures the density of the valid pixels. The conditions (iii)–(v) determine whether the CCs in the GTL well aligned: if a series of CCs constitutes a text line, they should be well aligned. The foregoing decision conditions are obtained by analyzing many experimental results of processing document images having text strings with various types, lengths and sizes. The constant values utilized under the above decision conditions are determined experimentally, and yield good performance in most general cases.

After all text-lines have been extracted from all of the object layers, all of the text lines are collected as the final segmentation result, as shown in Fig. 4(c). Figure 4(d) represents the binarized text image of Fig. 4(c).

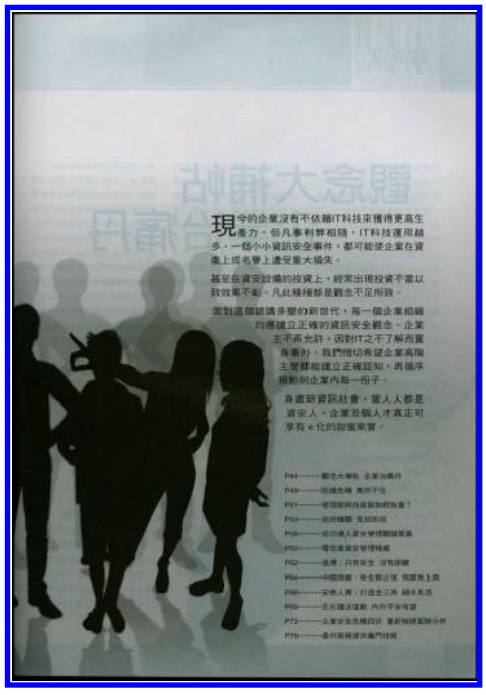
4. Experimental Results and Discussion

This work is focused on 24-bit true color or 8-bit monochromatic full-page document images at 300 dpi. The proposed method of automatic text segmentation was tested on numerous magazine images, cover images and advertisement images. Figures 5(a)–9(a) show parts of the test images. The background images in Figs. 5(a) to 9(a) exhibit the following characteristics. (1) Monochromatic background with/without text; (2) slowly varying background with/without text; (3) highly varying background with/without texts, and (4) complex varying background with/without text of various colors.

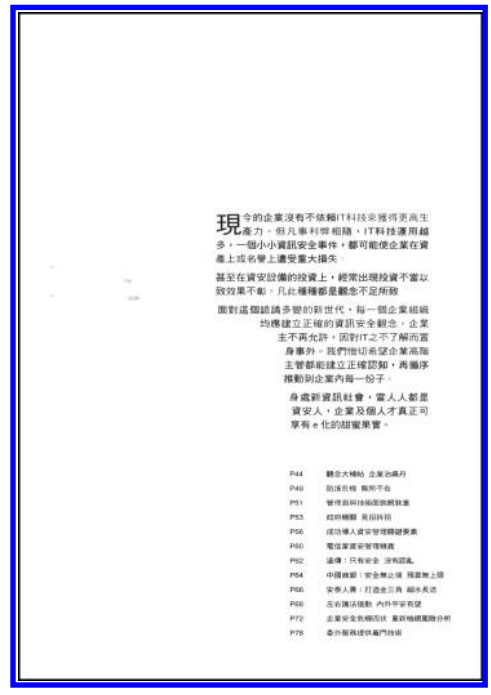
Figures 5(b)–9(b) show the text planes in Figs. 5(a)–9(a) after the proposed text segmentation method has been applied. Figures 5(c)–9(c) present parts of the object layers in Figs. 5(a)–9(a). The ratio of success of the proposed text segmentation method is

$$\text{Ratio of success} = \frac{\text{Number of texts extracted}}{\text{Total number of texts}} \%. \tag{32}$$

The ratios of success in Figs. 5(b)–9(b) are 100%, 98.5%, 99.2%, 100% and 97%, respectively. The proposed text segmentation method was successfully applied to extract texts of different typefaces or sizes, as well as those spread in a compound document image with monochromatic, slowly varying, highly varying and complex varying backgrounds.



(a) Original image



(b) Text plane

(c) Parts of layer planes

Fig. 5. Test image 1 (image size = 2262 × 3263).

The MLSM decomposes the document image into many object layers. All texts are spread into different object layers, according to their illuminations or colors. The text extraction algorithm extracts the text from all of the object layers. Because different object layers may contain text-like blocks in a particular position, the text-extracting algorithm may make an incorrect decision. Therefore, the text extraction

(a) Original image

(b) Text plane

(c) Parts of layer planes

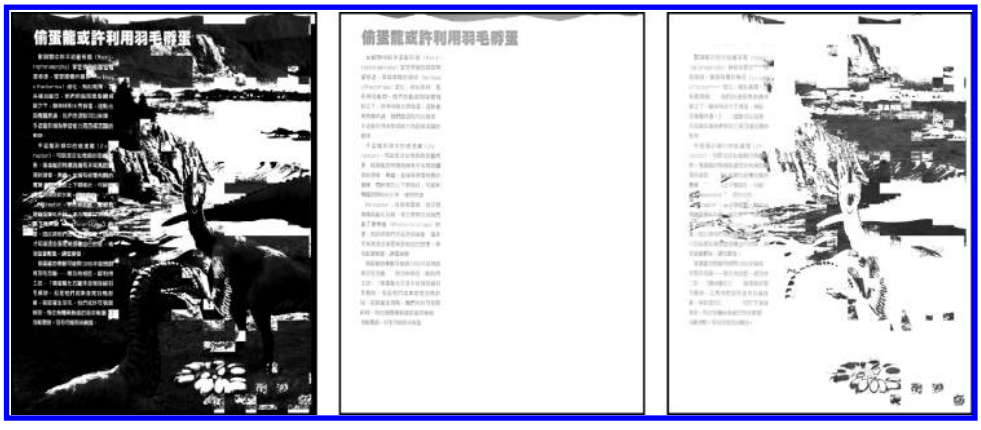
Fig. 6. Test image 2 (image size = 1829 × 2330).

algorithm can be further enhanced. For example, although most of the text in Fig. 6(a) overlays a complex varying background — a map — all of the text that overlaps the map is segmented into one of the object layers in Fig. 6(c). Although the ratios of success in Figs. 6(b), 7(b), and 9(b) are not 100%, the MLSM successfully segments most of the texts.

Based on the results obtained, the texts can be extracted from various backgrounds, independently of whether the texts overlap a simple, slowly or rapidly varying background. This method overcomes the various issues associated with the

(a) Original image

(b) Text plane

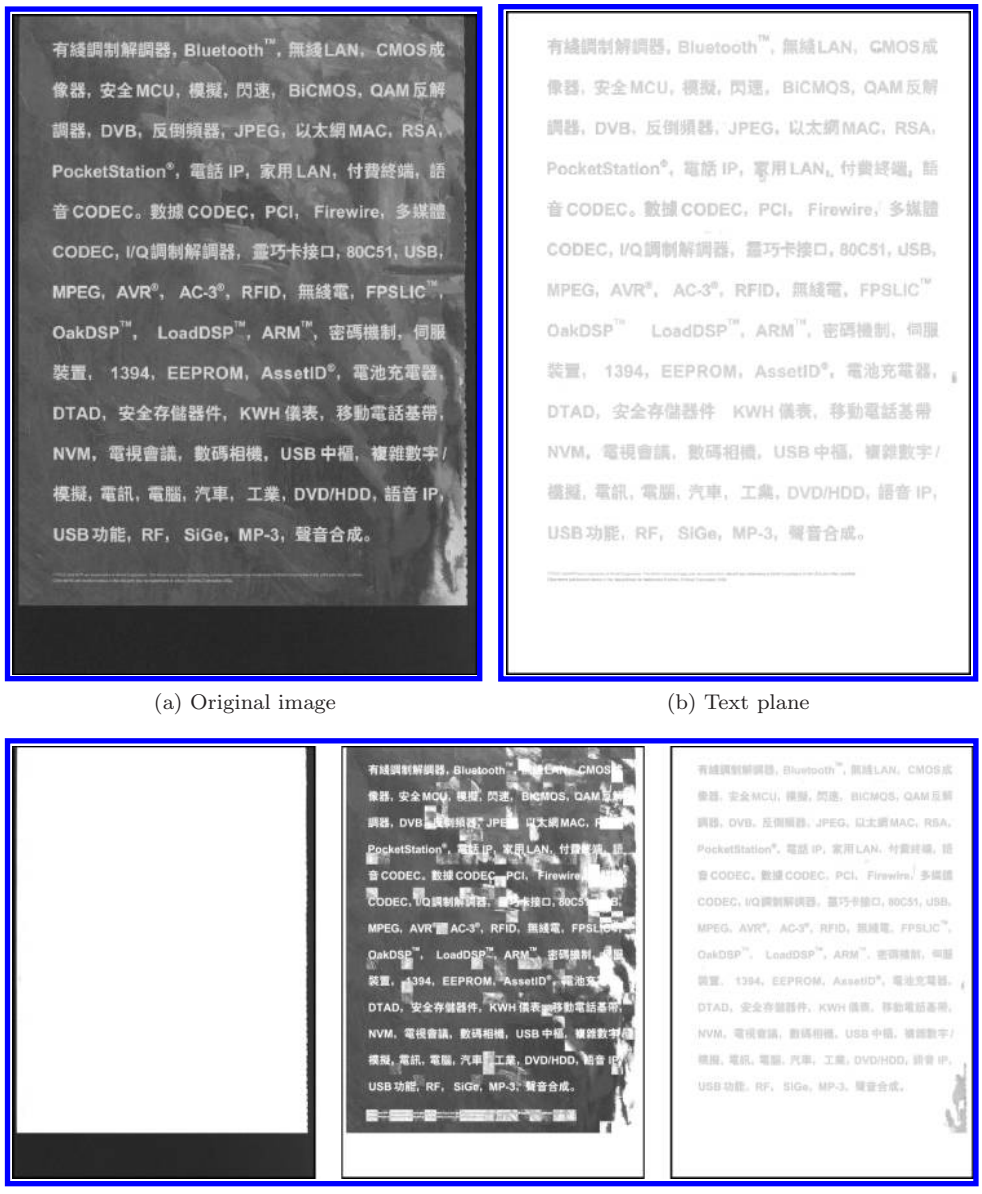


(c) Parts of layer planes

Fig. 7. Test image 3 (image size = 2462 × 3250).

complexity of background images. It is evident that the multilayer segmentation algorithm is an effective solution for extracting text from various document images.

In a block-based clustering algorithm, the parameters $\mathrm{TH_{JDF}}$ and TH_σ are the threshold values for determining which cluster converges under the conditions $\mathrm{JDF} > \mathrm{TH_{JDF}}$ and $\sigma < \mathrm{TH}_\sigma$. The JDF value measures the separability between two adjacent clusters in the block-based clustering algorithm. The JDF value may lie within the range $0 \leq \mathrm{JDF} \leq 1$. The JDF value can be maximized as an objective



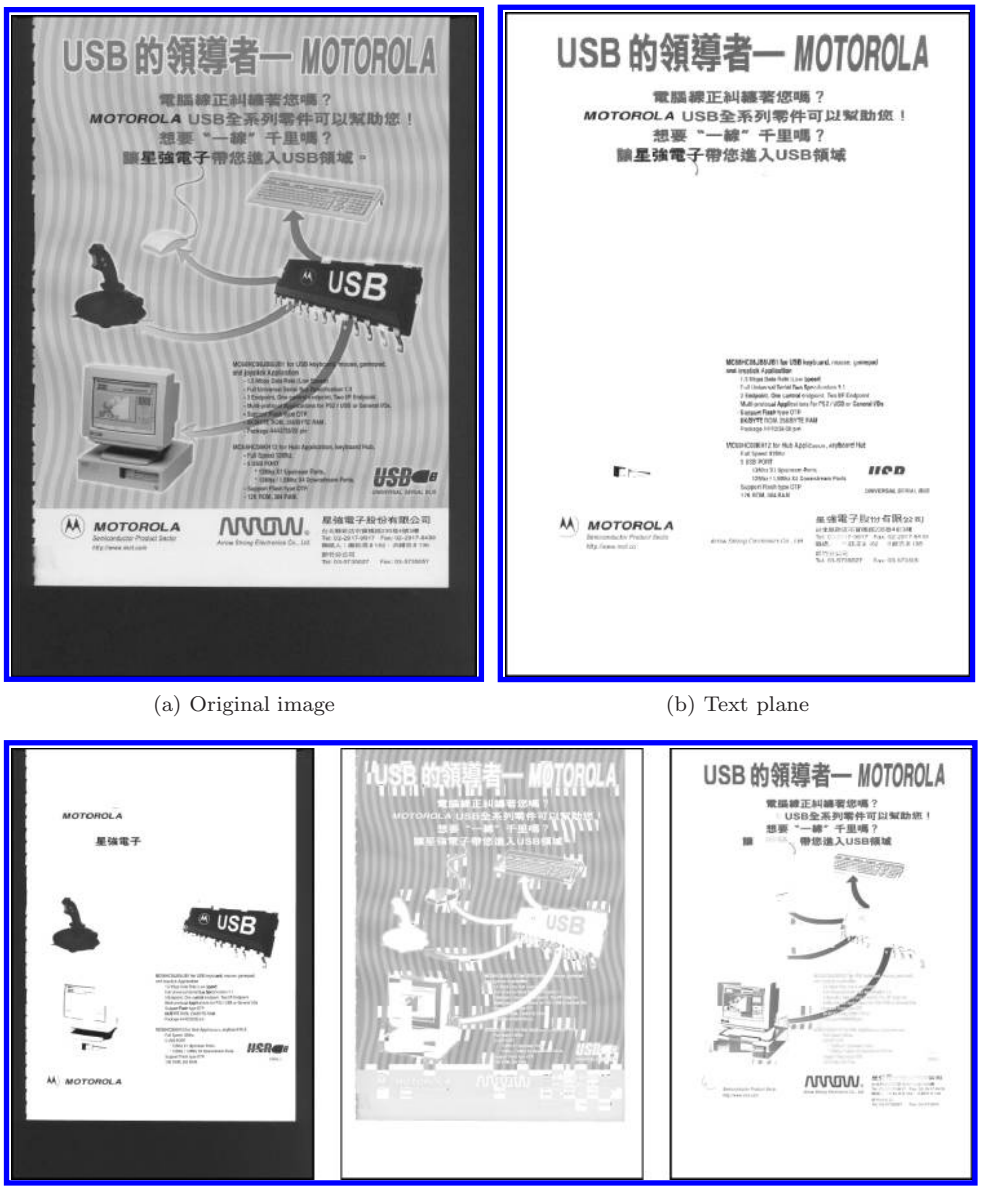
(a) Original image

(b) Text plane

(c) Parts of layer planes

Fig. 8. Test image 4 (image size = 2469 × 3535).

function to optimize the segmentation result. Hence, when the JDF is approximately unity, the two adjacent clusters are ideally and completely separated. When the number of the clusters exceeds two, the mean JDF is used to measure the separability of the clusters. This work uses $\mathrm{TH}_{\mathrm{JDF}} = 0.9$.



(a) Original image

(b) Text plane

(c) Parts of layer planes

Fig. 9. Test image 5 (image size = 2469 × 3535).

The standard deviation, σ, measures the compactness of the pixel values of each cluster. Ideally, the σ approximates zero for a monochromatic object. A pilot experiment is performed to analyze wide distributions, caused by the scanner or the original document, of the pixel values of monochromatic texts in various document

images. The mean variation of the monochromatic texts with size or style is about 0–50. $TH_\sigma = 25$ normally preserves the texts well, but it is not sufficient for reaching the goals of this work. Therefore, this study uses $TH_\sigma = 14$ to yield a better outcome, when the texts overlap a background with rapidly varying texture and similar grayscale. When the TH_σ is under 25, the extracted texts are thinner than the original texts, and the boundaries of the texts are clustered into various object layers, as shown in Fig. 1(j).

TH_σ is set to 14, so the standard deviation, σ, of each LSB is under 14. In other words, if two LSBs are in the same object layer, then the difference between the average values of the two LSBs is under 14. The threshold values of Th_{LM} and Th_{SI} are applied to determine whether the two LSBs belong to the same object layer, based on the difference between their average values. Hence, the threshold values of Th_{LM} and Th_{SI} are set to 14. In the decision procedure for establishing a new object layer, Th_{SI} is applied to determine which unclassified LSB should be merged with an existing object layer or which should be used to establish a new object layer. Under the prematch condition of the matching procedure, the Th_{LM} is used to filter out the unreasonable object layers and thus save computing power.

The segmentation method proposed in this paper was experimentally applied to numerous document images, scanned from book covers, advertisements, brochures and magazines. The MLSM can successfully separate monochromatic objects, text or nontext, from a document image; however, a few texts may not be extracted when the pixel values of the texts are multicolor, slowly changing, or too close to the pixel values of the background. A multicolor or slowly changing text will be fragmented and distributed across different clusters. A text may be merged with its background when the values of the text are too close to those of its background. Reducing the parameter TH_σ (below 14) can separate the text from the overlapping background whose values are very close to the text, thereby solving the problem of merging. However, doing so will cause the text to become fragmented and distributed across different clusters. Hence, an adaptive threshold TH_σ must be developed in the future to solve the merging problem.

5. Conclusions

This study has presented a viable method for extracting texts from a complex compound document image in which texts are overlaid on various background images. The proposed segmentation algorithm applies a multilayer segmentation method to segment the texts from various compound document images, independently of whether the texts overlap the background or not. This method overcomes various issues associated with the complexity of the background image. The experimental results obtained using various document images reveal that the proposed algorithm can successfully segment Chinese and English text strings from various backgrounds, regardless if the texts overlap a simple, slowly or rapidly varying

background. The approach can be applied to improve the effectiveness of compression; the technique has many applications, including compressing color faxes and documents. Additionally, the segmentation algorithm can be applied in Optical Character Recognition (OCR) to search for characters in complex documents with strongly overlapping text and background.

References

1. M. Acharyya and M. K. Kundu, Document image segmentation using wavelet scale-space features, *IEEE Trans. Circuits Syst. Vid. Technol.* **12**(12) (2002) 1117–1127.
2. H. Cheng and C. A. Bouman, Multiscale Bayesian segmentation using a trainable context model, *IEEE Trans. Imag. Process.* **10**(4) (2001) 511–524.
3. H. Choi and R. G. Baraniuk, Multiscale image segmentation using wavelet-domain hidden Markov models, *IEEE Trans. Imag. Process.* **10**(9) (2001) 1309–1321.
4. J. L. Fisher, S. C. Hinds and D. P. D'Amato, Rule-based system for document image segmentation, in *Proc. 10th Int. Conf. Pattern Recognition* (1990), pp. 567–572.
5. H. P. Li, D. Doermann and O. Kia, Automatic text detection and tracking in digital video, *IEEE Trans. Imag. Process.* **9**(1) (2000) 147–156.
6. J. Li and R. M. Gray, Context-based multiscale classification of document images using wavelet coefficient distributions, *IEEE Trans. Imag. Process.* **9**(9) (2000) 1604–1616.
7. R. Lienhart and A. Wernicked, Localizing and segmenting text in images and videos, *IEEE Trans. Circuits Syst. Vid. Technol.* **12**(4) (2002) 236–268.
8. Y. Liu and S. N. Srihari, Document image binarization based on texture features, *IEEE Trans. Patt. Anal. Mach. Intell.* **19**(5) (1997) 540–544.
9. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst. Man Cybern.* **8** (1978) 62–66.
10. M. Pietikinen and O. Okun, Edge-based method for text detection from complex document images, in *Proc. Int. Conf. Document Analysis and Recognition* (2001), pp. 286–291.
11. R. L. Queiroz, Z. Fan and T. D. Tran, Optimizing block-thresholding segmentation for multilayer compression of compound images, *IEEE Trans. Imag. Process.* **9**(9) (2000) 1461–1471.
12. F. Y. Shih, S. S. Chen, D. C. D. Hung and P. A. Ng, Document segmentation, classification and recognition system, in *Proc. IEEE Int. Conf. System Integration* (1992), pp. 258–267.
13. M. Worring and L. Todoran, Segmentation of color documents by line oriented clustering using spatial information, in *Proc. Int. Conf. Document Analysis and Recognition* (1999), pp. 67–70.
14. V. Wu, R. Manmatha and E. M. Riseman, Finding text in images, in *Proc. 2nd ACM Int. Conf. Digital Libraries* (1997), pp. 3–12.
15. H. Yang, M. Kashimura, N. Onda and S. Ozawa, Extraction of bibliography information based on image of book cover, *Int. J. Pattern Recognition and Artificial Intelligence* **14**(7) (2000) 963–978.
16. Q. Yuan and C. L. Tan, Text extraction from gray scale document images using edge information, in *Proc. Int. Conf. Document Analysis and Recognition* (2001), pp. 302–306.
17. J. Zhou and D. Lopresti, Extracting text from WWW images, in *Proc. Int. Conf. Document Analysis and Recognition* (1997), pp. 248–252.

Bing-Fei Wu received the B.S. and M.S. degrees in control engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1992. From 1983 to 1984, he was with the Institute of Control Engineering, NCTU as an Assistant Researcher. From 1985 to 1988, he was with the Department of Communication Engineering at the same university as a Lecturer. Since 1992, he has been with the Department of Electrical Engineering and Control Engineering, where he is currently a Professor. As an active industry consultant, he is also involved in the chip design and applications of the flash memory controller and 3C consumer electronics in multimedia.

His research interests include chaotic systems, fractal signal analysis, multimedia coding, wavelet analysis and applications.

Chung-Cheng Chiu received the Ph.D. degree from the Department of Electrical and Control Engineering at National Chiao Tung University, Hsinchu, Taiwan, in 2004. Since 1993, he is a lecturer at the Department of Electrical Engineering, Chung Cheng Institute of Technology. In 2005, he became an Associate Professor. In 2003, he received Dragon Golden Paper Award sponsored by the Acer Foundation and the Silver Award of Technology Innovation Competition sponsored by the AdvanTech.

His research interests include image processing, image compression, document segmentation and computer vision.

Yen-Lin Chen received the B.S. degree in electrical and control engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2000. He is currently working towards the Ph.D. degree in the same institute. In 2003, he received Dragon Golden Paper Award sponsored by the Acer Foundation and the Silver Award of Technology Innovation Competition sponsored by the AdvanTech.

His research interests include image and video processing, pattern recognition, document image analysis and applications on intelligent transportation system.