Machine Vision and Applications (2008) 19:195–207
DOI 10.1007/s00138-007-0092-0

ORIGINAL PAPER

Morphology-based text line extraction

Jui-Chen Wu · Jun-Wei Hsieh · Yung-Sheng Chen

Received: 13 September 2006 / Revised: 3 March 2007 / Accepted: 24 April 2007 / Published online: 3 August 2007

Abstract This paper presents a morphology-based text line extraction algorithm for extracting text regions from cluttered images. First of all, the method defines a novel set of morphological operations for extracting important contrast regions as possible text line candidates. The contrast feature is robust to lighting changes and invariant against different image transformations like image scaling, translation, and skewing. In order to detect skewed text lines, a moment-based method is then used for estimating their orientations. According to the orientation, an x-projection technique can be applied to extract various text geometries from the text-analogue segments for text verification. However, due to noise, a text line region is often fragmented to different pieces of segments. Therefore, after the projection, a novel recovery algorithm is then proposed for recovering a complete text line from its pieces of segments. After that, a verification scheme is then proposed for verifying all extracted potential text lines according to their text geometries. Experimental results show that the proposed method improves the state-of-the-art work in terms of effectiveness and robustness for text line detection.

Keywords Morphological operations · Text line extraction · Text verification · Video understanding · Document analysis

1 Introduction

Extracting text lines from images or videos is an important problem in many applications like document processing [1,2], image indexing, video content summary [3–5], video retrieval [6], video understanding [7], and so on. Usually, texts embedded in an image or a frame capture important media contexts such as player's name, title, date, story introduction, and so on. Therefore, the task can provide various advantages for annotating an image or a video and thus improves the accuracy of a content-based indexing system to search desired media contents. In addition, the information can be used for content filtering so that commercial programs can be found and removed out for video summary. Moreover, when analyzing video audios, the recognition result of text line can provide extra refinements for correcting the errors of speech recognition.

Texts usually have different appearance changes like font, size, style, orientation, alignment, texture, color, contrast, and background [1]. All the changes will make the problem of automatic text extraction become complicated and difficult. There are many researchers who have devoted themselves to investigating different methods for tackling the above problems [1–7]. For example, Sato et al. [4] proposed a caption detection system to detect and recognize text characters embedded in video captions. Zhong et al. [8] assumed that the characters embedded in an image usually have similar color, and then proposed a color reduction technique for locating them from complex images. Lienhart et al. [9] made another assumption that text lines usually have high contrast to background, and then applied a motion analyzer to extract text line locations. In addition to the above character properties, the change between character boundaries also forms a good feature for text line extraction. Hasan and Karam [10] used several morphological operations to extract this feature for text line localization. Wong and Chen [11] computed this feature by accumulating the maximum gradient differences of pixels line by line for obtaining all potential text segments in the processed image. This feature can also

J.-C. Wu · J.-W. Hsieh (✉) · Y.-S. Chen
Department of Electrical Engineering, Yuan Ze University,
135 Yuan-Tung Road, Chung-Li 320, Taiwan, ROC
e-mail: shieh@saturn.yzu.edu.tw

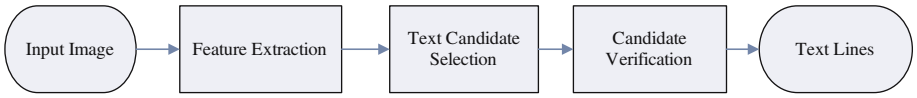


Fig. 1 Flowchart of the proposed system

be extracted from frequency domain. For example, Sin et al. [12] took advantages of Fourier spectrum to extract high frequency components for text line detection. Mao et al. [13] used wavelet transform to obtain edge maps at different resolutions for locating all possible text lines. The feature can also be obtained using a training process. For example, Kim et al. [14] used support vector machines (SVM) to learn important text features. Xiangrong et al. [15] used an Adaboost algorithm to build a stronger classifier for text line detection. The training scheme has superiorities in detecting normal text lines but often fails to detect skewed text lines. The training approach needs lots of training samples to train each configuration and becomes inefficient when more orientations of text line are detected.

In this paper, a novel text line detection scheme is proposed for locating different text lines from cluttered images. The major contribution of this paper is to devise a novel morphology-based technique for extracting important text contrast features from the processed images. The feature is invariant against various geometrical image changes like translation, rotation, and scaling. Even though the lighting condition or text color has changes, the feature still can be maintained. Thus, the proposed morphology-based method works robustly under different image alterations. After that, a coarse-to-fine text verification scheme is proposed to verify each text-analogue segment. The coarse scheme uses two constraints including the size and the width-to-height ratio between texts to filter out all impossible text candidates. Then, a finer verification scheme is applied to verify all remained text candidates using their detailed character features. Since each text line has different orientations, this paper first uses a moment-based method to find its longest axis. Then, we can benefit from an x-projection technique to find text character geometries for the finer verification. After text verification, due to noise, some character regions still will be missed or a complete text is fragmented into pieces of segments. To avoid the text missing or fragmentation problem, a recovery algorithm is then proposed to adjust text boundary so that all missed text components can be recovered. Without any training process, all possible text lines can be correctly verified. The proposed method can well detect various text lines even though they are skewed. In addition, no matter how cluttered the background is, all desired text regions can be very accurately located. The average accuracy of text line detection is 95.4%. Experimental results have shown the superiority of the proposed method in text line detection.

The rest of this paper is organized as follows. In the next section, the flowchart of the proposed method is first illustrated. Then, details of feature extraction using morphological operations are described in Sect. 3. The scheme of text line candidate extraction is discussed in Sect. 4 and details of the algorithm of text region verification are illustrated in Sect. 5. Section 6 reports the experimental results. Finally, a conclusion will be presented in Sect. 7.

2 Overview of the proposed system

This paper presents a novel technique for automatically locating text lines from cluttered images. Figure 1 shows the flowchart of the proposed system. The system consists of three major parts, i.e., feature extraction, candidate selection, and verification. In what follows, details of each part are described.

Feature extraction Text lines embedded in images often have quite high contrast to the background. In addition, their widths and heights are usually uniform and horizontally aligned. The relative contrast between texts and their background is an important feature for text line detection. Thus, this paper proposes a novel morphology-based scheme for extracting the high contrast feature for locating all possible text lines.

Text candidate selection After feature extraction, we can use a labeling technique to select all possible text lines from the analyzed image. For handling skewed text lines, a moment-based method is then applied to find their orientations. After that, a novel rule-based selection scheme is proposed to select all potential segments. Furthermore, a novel merging algorithm is proposed for merging the segments together if they belong to the same text line.

Candidate verification Once all potential text lines have been selected, a verification procedure will be then proposed for filtering out all impossible candidates. The criteria of text verification include the regularities of character size, the ratio between character width and height, and the period of characters. We propose an x-projection technique to extract all the regularities from each text line. After verification, an extending technique is further used for adjusting text boundaries so that all the missed text pixels can be well recovered.

3 Feature extraction

In order to well detect desired text lines from the cluttered background, a novel morphology-based approach will be presented in this section to extract high contrast regions as

$S_{1,5}$ = | 1 | 1 | 1 | 1 | 1 |

(a)

$S_{3,7}$ =

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

(b)

Fig. 2 Two kinds of structure elements

text candidates. The used high-contrast feature is invariant against text orientations and highly tolerant to noise. Before introducing the proposed method, some morphological operations should be first described.

Let $S_{m,n}$ denote a structure element with the size $m \times n$, where m and n are odds and larger than zero. Figure 2 shows two kinds of structure element. Let $I(x, y)$ denote a gray-level input image. According to the definition of $S_{m,n}$, the smoothing, dilation, erosion, closing, opening, and other operations are defined as follows:
smoothing operation:

$$E_{S_{m\times n}}(I(x,y)) = \frac{1}{mn}\sum_{i=-m/2}^{m/2}\sum_{j=-n/2}^{n/2} I(x+i, y+j)S_{m,n}(i,j), \quad (1)$$

dilation operation:

$$I(x,y) \oplus S_{m,n} = \max_{|i|\le m/2, |j|\le n/2} I(x-i, y-j)S_{m,n}(i,j), \quad (2)$$

erosion operation:

$$I(x,y) \odot S_{m,n} = \min_{|i|\le m/2, |j|\le n/2} I(x-i, y-j)S_{m,n}(i,j), \quad (3)$$

closing operation:

$$I(x,y) \bullet S_{m,n} = (I(x,y) \oplus S_{m,n}) \odot S_{m,n}, \quad (4)$$

opening operation:

$$I(x,y) \circ S_{m,n} = (I(x,y) \odot S_{m,n}) \oplus S_{m,n}, \quad (5)$$

differencing operation:

$$D(I_1, I_2) = |I_1(x,y) - I_2(x,y)|, \quad (6)$$

and thresholding operation:

$$T(I(x,y)) = \begin{cases} 255, & if \ I(x,y) > T; \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

As mentioned previously, for reading easily, text lines are specially designed with high contrast to their background color. Even though their lightings have changes, the feature still can be maintained and invariant to several geometrical transformations like camera translation, rotations, and scaling. Thus, the feature plays an important role in text line detection. In this paper, we use a series of morphological operations to extract points which have high gradients to their background as the contrast feature. Figure 3 shows the whole procedure of our novel morphology-based technique to extract the feature. Firstly, in order to eliminate noise, a smoothing operation with a structure element $S_{3,3}$ is first applied. Then, the closing and opening operations with a structure element $S_{1,7}$ are performed into the smoothed image so that the output images I_c and I_o can be obtained, respectively. Furthermore, for detecting text boundary edges, a differencing operation is further applied into I_c and I_o. In order to make these edges more compactly and closely, a closing operation is then used so that all characters embedded in a text line can form a connected segment. After that, a thresholding operation is used for converting the analyzed image into a binary map. Then, a labeling process is executed to extract the text-analogue segments. After that, a set of potential text lines can be obtained for further verification. Figure 4 shows an example of the above morphology-based technique to extract text regions as text line candidates. Figure 4a and b are the original images. Figure 4c and d are their corresponding results of morphological operations using our proposed method. Clearly, all possible text candidates were well detected.

Hasan and Karam [10] also used several morphological operators to detect text lines. However, there are many differences between our method and their approach. Firstly, they used only the dilation and erosion operations to extract edges as text features. Instead of using these operations, we use the closing and opening operators to detect text lines. In addition, we use a horizontal structure element to extract contrast feature. Compared with their method, our approach performs more robustly in the abilities to deal with noise,

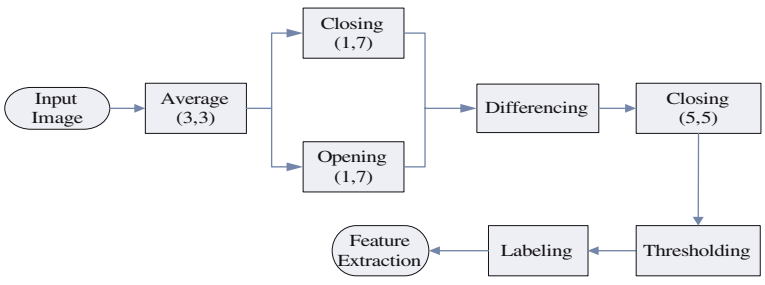


Fig. 3 Flowchart of the proposed method to extract contrast features for text line detection

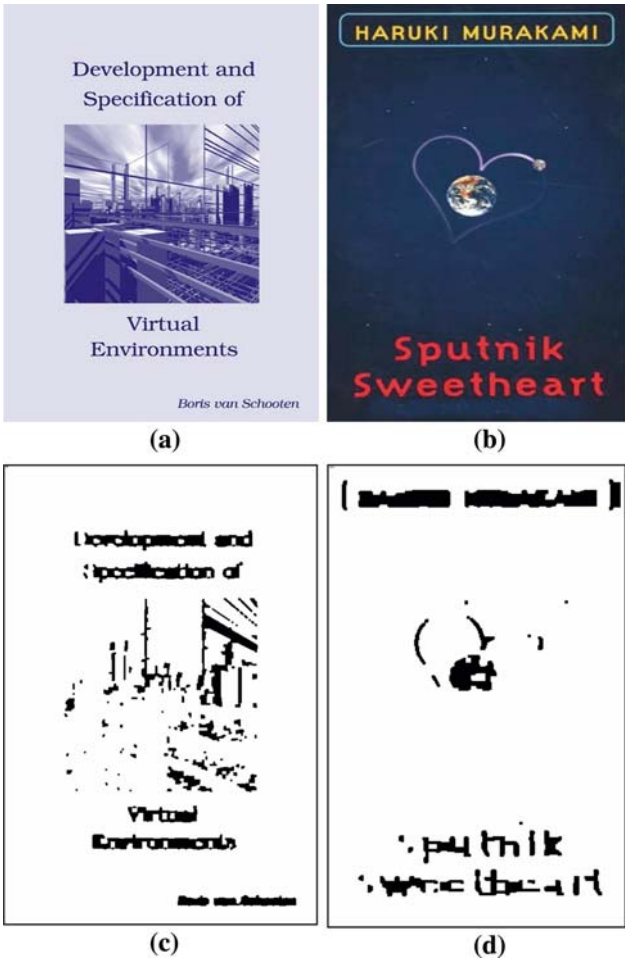


Fig. 4 Result after morphological operations. **a** and **b** Original images. **c** and **d** results of extracted contrast features obtained from **a** and **b**, respectively

cluttered background, and text fragmentation. Secondly, their approach used the closing operation after differencing. But our method uses the closing operation before differencing. Our approach will make text features more compactly and closer to each other. Thirdly, their approach did not discuss the problems when text lines are fragmented, missed, or skewed. All these problems will lead to the failure of text line detection and will be tackled later in this paper.

4 Text candidate selection and merging

In Sect. 3, a novel morphology-based scheme has been proposed to extract high-contrast regions as potential text segments. However, a text line is not always horizontally aligned and sometimes fragmented into several small segments. In order to detect a skewed text line, in what follows, a moment-based method is first used for estimating its orientations. Then, a merging technique is proposed to link all the missed fragments.

4.1 Moment-based orientation estimation

Given a binary region $R(x, y)$, the central moments of R can be defined as

$$(\mu_{p,q})_R = \frac{1}{|R|} \sum_{(x,y)\in R} (x - \bar{x})^p (y - \bar{y})^q,$$

where $(\bar{x}, \bar{y}) = \frac{1}{|R|}(\sum_{(x,y)\in R} x, \sum_{(x,y)\in R} y)$ and $|R|$ is the area of R. Here, if a pixel (x, y) belongs to R, the value of $R(x, y)$ is one; otherwise, its value is zero. Then, as shown in Fig. 5, the orientation θ_R of R can be obtained using the equation:

$$\theta_R = \arg \min_{-\pi<\theta\le\pi} \sum_{(x,y)\in R} [(x - \bar{x}) \sin\theta - (y - \bar{y}) \cos\theta]^2. \tag{8}$$

Setting the term $\frac{1}{\partial\theta} \sum_{(x,y)\in R} [(x - \bar{x}) \sin\theta - (y - \bar{y}) \cos\theta]^2$ to zero, we can get

$$\theta_R = \frac{1}{2} \tan^{-1} \left[\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right]. \tag{9}$$

With θ_R, even though a text line is not horizontally aligned, we still can well detect it from the background.

4.2 Text candidate selection

Let R be a potential text line extracted from our morphology-based scheme. Due to noise, many impossible non-text regions will be also extracted. This section will use a rule-based scheme to coarsely eliminate impossible candidates. Then, in Sect. 5, a finer verification scheme will be proposed to more accurately verify all the remained text segments.

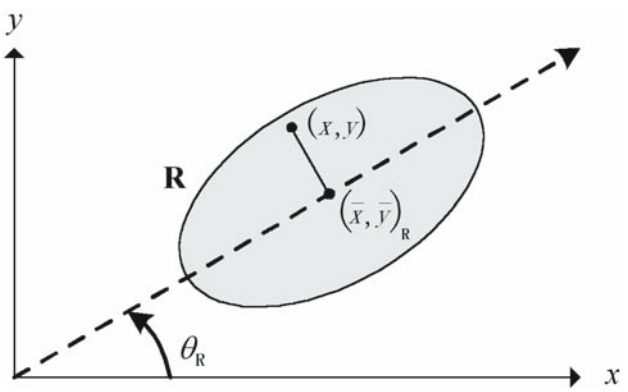


Fig. 5 The gravity center $(\bar{x}, \bar{y})_R$ and orientation θ_R of an object R

Assume that R^θ is the rotated version of R with its orientation θ_R. In addition, w_{R^θ} and h_{R^θ} denote the width and height of R^θ, respectively. Since a text line has a longer width than its height, the first rule requires the ratio $\gamma(R^\theta)$ between w_{R^θ} and h_{R^θ} being larger than 1.5. In addition, the density of R should be larger enough, i.e., $\text{den} = \frac{\text{Area of} R}{w_{R^\theta} \times h_{R^\theta}} > 0.1$. The final rule requires the area of R being not too small. According to the above requirements, R is a text line candidate if it satisfies the following three rules:

Rule 1: den should be larger than 0.1;
Rule 2: the ratio $\gamma(R^\theta)$ between w_{R^θ} and h_{R^θ} should be larger than 1.5;
Rule 3: the area of R should be larger than a threshold, i.e., 2.5 pixels.

4.3 Text line merging

After filtering out impossible text regions using the above rule-based approach, this section will propose a novel merging scheme to deal with the problem of text fragmentation. Given two regions R_i and R_j, if they come from the same text line, this section defines four similarity measures for determining whether they should be merged. In practice, if R_i and R_j belong to the same text line, their heights and orientations should be similar. In addition, their centroids and major axes are close to each other. Therefore, if R_i and R_j belong to the text line, the first criterion requires the orientation difference between them being less than 10°, i.e.,

$$d_\theta(R_i, R_j) = \min(|\theta_{R_i} - \theta_{R_j}|, |\theta_{R_i} - \theta_{R_j} + 360°|, |\theta_{R_i} - \theta_{R_j} - 360°|) < 10°, \quad (10)$$

where θ_{R_i} and θ_{R_j} are the major orientations of R_i and R_j, respectively. The threshold 10° is not related to the sizes of image and font. In addition, we also require the heights of R_i and R_j being similar enough, i.e.,

$$d_h(R_i, R_j) = \frac{2|h_{R_i^{\theta_{R_i}}} - h_{R_j^{\theta_{R_j}}}|}{h_{R_i^{\theta_{R_i}}} + h_{R_j^{\theta_{R_j}}}} < 0.15. \quad (11)$$

Since Eq. (11) is a relative constraint, the threshold 0.15 is not subject to the size of image and font.

Let Cen_{R_i} and Cen_{R_j} be the centroids of R_i and R_j, respectively. The third criterion requires Cen_{R_i} and Cen_{R_j} being closer to each other; that is,

$$|\text{Cen}_{R_i} - \text{Cen}_{R_j}| < 2(h_{R_i^{\theta_{R_i}}} + h_{R_j^{\theta_{R_j}}}). \quad (12)$$

Let L_R be the longest axis of R denoted by this equation: $y = m_R x + b_R$. Then, the distance between the major axes L_{R_i} and L_{R_j} of R_i and R_j can be defined as follows:

$$d_L(R_i, R_j) = \frac{|y_{\text{Cen}_{R_j}} - m_{R_i} x_{\text{Cen}_{R_j}} - b_{R_i}|}{2\sqrt{1 + m_{R_i}^2}} + \frac{|y_{\text{Cen}_{R_i}} - m_{R_j} x_{\text{Cen}_{Ri}} - b_{R_j}|}{2\sqrt{1 + m_{R_j}^2}}, \quad (13)$$

where x_{Cen_R} and y_{Cen_R} are the coordinates of Cen_R in the x and y directions, respectively. The fourth criterion requires the distance $d_L(R_i, R_j)$ being < 5 pixels, i.e.,

$$d_L(R_i, R_j) < 5. \quad (14)$$

Based on Eqs.(10)–(12), and (14), we can determine whether R_i and R_j should be merged or not. Thus, different text candidates can be well selected for further verification. The threshold 5 is subject to font size. When large font size is handled, it should be changed (become larger). However, the problem can be easily tackled if a pyramid structure is used. In this structure, the input image is gradually reduced into a smaller one with a scale factor (like 0.9). Then, different text lines even with a larger font size can be well detected from the pyramid structure. Figure 6 demonstrates the result of text line merging. Figure 6a and b are the results of contrast area detection using our morphology-based method. Figure 6c is the result of text line merging.

5 Text line verification

Once all potential text lines have been extracted, this section will propose a finer verification process for verifying all the remained text candidates. The verification process uses an x-projection technique to extract desired text features for verifying the above remained text candidates. In what follows, details of the x-projection technique are first discussed. Then, for avoiding a text line being fragmented into different pieces, a novel recovery algorithm will be proposed to recover the missed text data together.

5.1 X-projection

In Sect. 4, a set of potential text candidates has been extracted using our proposed morphology-based method. Before verification, each gray text region should be binarized into a binary map using a threshold T_R. In this paper, the " *minimum within-group variance*" dynamic thresholding method [16] is adopted for finding T_R. Then, given a text candidate R, after binization an x projection technique is used to find its various text geometries. In practice, for all characters embedded in R, they should satisfy the following requirements:
A1: their widths should be similar;
A2: their heights should be similar;

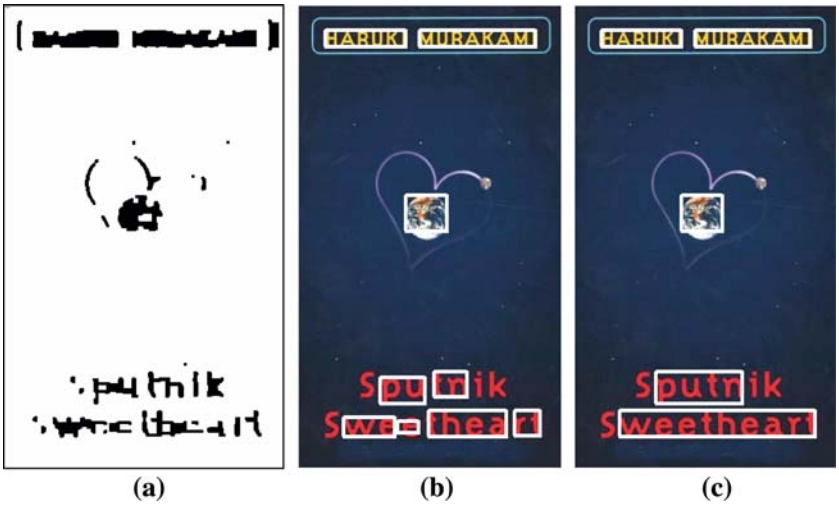


Fig. 6 Results of text line merging and selection. **a** Result of high contrast area detection. **b** Locating result of **a**. **c** Merging result of **b**

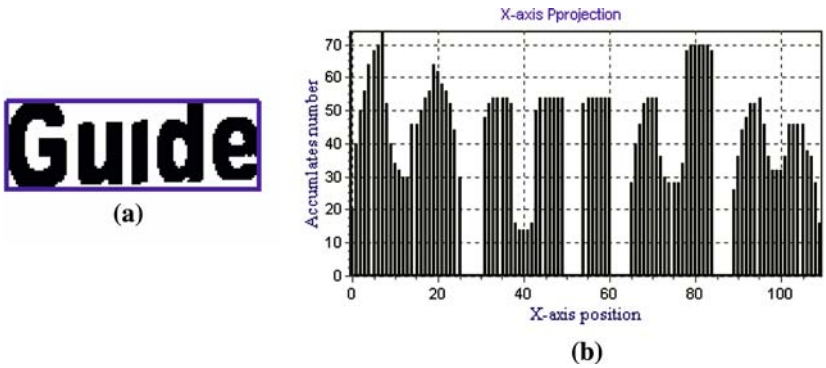


Fig. 7 Character analysis. **a** Original image; **b** result of the x-projection obtained from **a**

A3: their centers should be aligned along a straight line.

The projection technique can well separate R into different characters if R is a real text line. The technique tries to project all text pixels of R on its longest axis and then accumulates the number of character pixels column by column. Like Fig. 7b is the result of x-projection got from Fig. 7a. The minimum valleys can provide important information for separating R into different characters like 'g', 'u', 'i', 'd', and 'e'. Assume that $XP_R[i]$ is an array to record the x-projection result of R. Then, the minimum valleys in XP_R can be detected by seeking the points whose values in XP_R are locally minimum and less than a threshold, i.e., $\frac{h_{R^\theta}}{8}$. Here, h_{R^θ} is the height of the rotated version of R using its orientation θ. Using the minimum valleys, different characters C_i can be extracted from R. Assume that $\overline{W_R}$ and $\overline{h_R}$ are the average width and height of all characters C_i in R. Then, the width and height variances of characters in R can be calculated, respectively, as follows

$$\sigma^2_{W,R} = \frac{1}{N^C_R}\sum_{C_i \in R}\left(W_{C_i} - \overline{W_R}\right)^2 \quad \text{and}$$

$$\sigma^2_{h,R} = \frac{1}{N^C_R}\sum_{C_i \in R}\left(h_{C_i} - \overline{h_R}\right)^2, \tag{15}$$

where N^C_R is the number of characters in R. In addition, all the character centers in R form a line and satisfy the equation:

$$y = m^C_R x + b^C_R. \tag{16}$$

The values of m^C_R and b^C_R can be easily obtained using a line fitting technique [17]. Then, the linearity of R can be measured by

$$\text{Linearity}(R) = \frac{1}{|N^C_R|}\sum_{C_i \in R}\frac{\left|y_{C_i} - m^C_R x_{C_i} + b^C_R\right|}{\sqrt{\left(m^C_R\right)^2 + 1}}. \tag{17}$$

Thus, if R is a text line, it should satisfy

$$\sigma^2_{R,w} < T_w, \sigma^2_{R,L} < T_h, \quad \text{and } linearity(R) < T_l. \tag{18}$$

The values of T_w, T_h, and T_l can be obtained from thousands of training text samples. Then, different text lines can be correctly verified if they satisfy all requirements in Eq. (18).

5.2 Text line recovery

In the previous section, we have used the x-projection technique to verify and extract different text lines. However, due to noise or lighting changes, some characters still would be missed from the above text analysis. Therefore, this section will propose a novel recovery algorithm for adjusting the

boundaries of a text line R so that its all missed data can be well recovered.

Assume that R^θ is the rotated version of R with its major orientation θ. The whole recovery algorithm considers the longest axis of R as the x-axis and the center of R^θ as the original. Then, all the text characters in R^θ will be horizontally aligned. As described in the previous section, characters embedded in a text line usually have similar sizes, fonts, and intensities. The proposed extending technique will use these character similarities for recovering all missed text pixels from R. Let $l_{R^\theta}, r_{R^\theta}, t_{R^\theta}$, and b_{R^θ} denote the most left, right, top, and bottom coordinates of R^θ in the x and y directions, respectively. The recovery algorithm is iteratively performed. In each repetition, we create two new regions R_l and R_r extended from both sides of R^θ with the same height of R^θ and a fixed width. The fixed width is proportion to the average width of characters embedded in R^θ. Then, the threshold T_R, which was decided in Sect. 5.1 for binarizing R, is used to binarize R_l and R_r. According to the binary maps, different isolated characters can be found. An isolated character is a connected component which has similar width and height to $\bar{w}_{R,C}$ and $\bar{h}_{R,C}$. If no character is further isolated, the iteration to adjust text boundaries is terminated. Then, the final recovered region is the desired text line. Let W_I and H_I denote the width and height of the input image I, respectively. In what follows, details of the text recovery algorithm are described.

5.2.1 Text line recovery algorithm

Input: a text region R, the average character width $\bar{w}_{R,C}$ and height $\bar{h}_{R,C}$ in R.
Output: a new recovered region $\bar{R}$.
Step 1: According to the orientation θ_R of R, obtain its rotated version R^θ.
Step 2: Obtain the most left, right, top, and bottom coordinates of R^θ, i.e., $l_{R^\theta}, r_{R^\theta}, t_{R^\theta}$, and b_{R^θ}, respectively, by considering the longest axis of R^θ as the x axis.
Step 3: // left extension
S3.1: Create a new region $R^{\text{New}}_{\text{left}}$ with the boundary coordinates: $l = \max(0, l_{R^\theta} - 5\bar{w}_{R,C})$, $r = l_{R^\theta}, t = \max(0, t_{R^\theta} - \bar{h}_{R,C}/5)$, and $b = \min(b_{R^\theta} + \bar{h}_{R,C}/5, H_I)$.
S3.2: Binarize $R^{\text{New}}_{\text{left}}$ using T_R found in Sect. 5.1 for binarizing R.
S3.3: Check whether there are isolated characters in $R^{\text{New}}_{\text{left}}$.
S3.4: If any isolated character C_i is found, $l_{R^\theta} =$ the most left boundary of C_i and go to Step 3; otherwise, go to step 4.
Step 4: // right extension
S4.1: Create a new region $R^{\text{New}}_{\text{right}}$ with the boundary coordinates: $l = r_{R^\theta}, r = \min(r_{R^\theta} + 5\bar{w}_{R,C}, W_I), t = \min(t_{R^\theta} - \bar{h}_{R,C}/5, 0)$, and $b = \max(b_{R^\theta} + \bar{h}_{R,C}/5, H_I)$.
S4.2: Binarize $R^{\text{New}}_{\text{right}}$ using T_R.
S4.3: Check whether there are isolated characters appearing in $R^{\text{New}}_{\text{right}}$.
S4.4: If any isolated character C_i is found, $r_{R^\theta} =$ the most right boundary of C_i and go to Step 4; otherwise, go to step 5.
Step 5: Obtain $\bar{R}$ with the new boundary coordinates: $l_{R^\theta}, r_{R^\theta}, t_{R^\theta}$, and b_{R^θ}.

6 Experimental results

In order to analyze the performance of our proposed approach, an image database including 100 images was used for testing. For well testing our method, these images have various appearance changes like contrast changes, complex backgrounds, lightings, different fonts, and sizes. Figure 8 shows the results of text line detection when normal images were handled. Since the backgrounds were simple, all text lines were correctly detected. Figure 9 shows another case of text line extraction when text regions have low contrast to the background. In Fig. 9a, the background had various intensity changes. In Fig. 9b and c, text lines had low contrasts

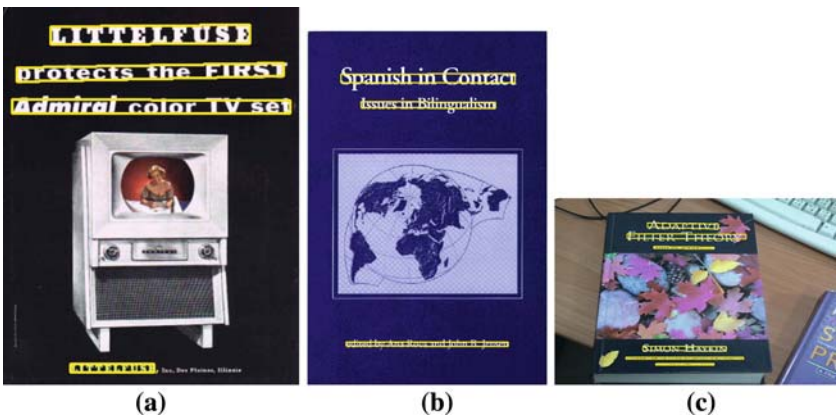


Fig. 8 Results of text detection when normal images were handled. Different text lines were well detected

Fig. 9 Results of text line extraction when text regions had low contrasts to the background. **a** Various intensity changes in the background. **b** and **c** Text lines having low contrasts to the background

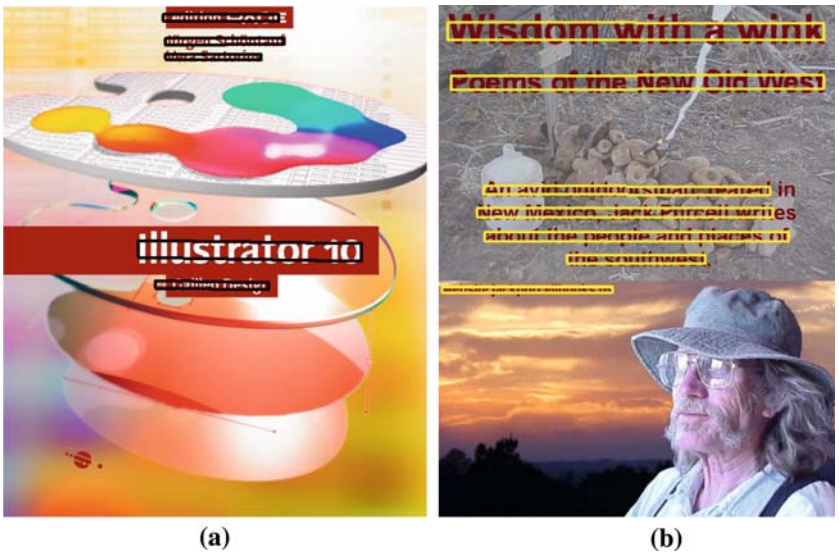


Fig. 10 Results of text line detection when cluttered backgrounds were handled. **a** Colorful background. **b** Text lines embedded in a textured background

to the backgrounds. No matter what cases were handled, our method still performed well to detect all descried text line. Figure 10 is the detection result when text lines were embedded in a cluttered background. In Fig. 10a, the background is colorful. In Fig. 10b, text lines were embedded in a textured background. Clearly, no matter what the background is, our method works very well to detect all desired text regions.

In real conditions, different text fonts and sizes will be embedded together in the same image. The changes of font and size will also affect the accuracy of text line detection. Figure 11 shows the results of text line detection when different fonts and sizes were handled. No matter what characters and fonts are embedded, the proposed method still works successfully to detect them. A more challenging work is to detect text line from video sequences since text lines have quite distortions after compression. Figure 12 and 13 show the results of text line detection when video frames were handled. Figure 14 shows another difficult task when text lines had different orientations. Even though the text lines were not horizontally aligned, they were still correctly detected. Clearly, no matter what cases are handled, all the described text lines can be well extracted using our proposed methods.

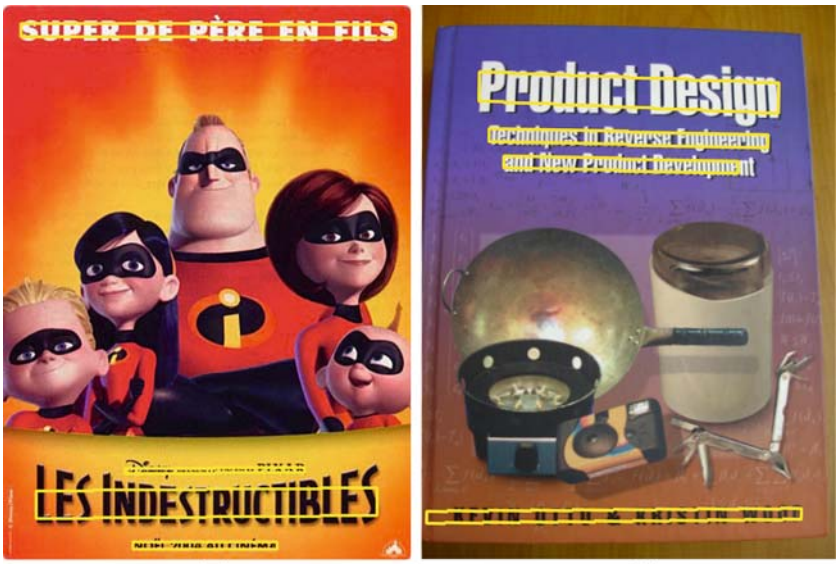
In another set of experiments, we also compared our proposed approach with two other approaches, i.e., Hasan and Karam [10] and the maximum gradient approach [11]. In order to evaluate the performance of each method, the precision and recall measures were used. Recall is the ratio of the number $\mathrm{Num}_{\mathrm{Correct}}$ of correct text lines detected by the algorithm to the total number $\mathrm{Num}_{\mathrm{actual}}$ of actual text lines appearing in the test images, i.e.,

$$\mathrm{Recall} = \mathrm{Num}_{\mathrm{Correct}} / \mathrm{Num}_{\mathrm{actual}}.$$

In addition, precision is the ratio of the number of text lines correctly detected by the algorithm to the total number $\mathrm{Num}_{\mathrm{Detected}}$ of detected text lines; that is,

$$\mathrm{Precision} = \mathrm{Num}_{\mathrm{Correct}} / \mathrm{Num}_{\mathrm{Detected}}.$$

For well analyzing these algorithms, we further divided our testing database into four categories according to their backgrounds (simple or cluttered) and their embedded text line orientations (normal or skewed). Figure 15 shows the comparison results among the three methods when different font sizes were handled. Hasan and Karam's method [10] also used several morphological operators to detect text lines. However, they used only the dilation and erosion



(a) (b)

Fig. 11 Results of text line extraction when different fonts and sizes were embedded in the same image

(a) (b)

(c)

Fig. 12 Results of text line extraction when video frames were handled

operators to extract text features. Then, each candidate is verified according to its density of text pixels. Their verification algorithm depends strongly on the geometries and sizes of individual characters. Therefore, their method is very sensitive to noise, cluttered backgrounds, and text fonts. As to Wong and Chen's [11] approach, their maximum gradient technique is sensitive to complicated background and text fonts. Consequently, when different text fonts and font sizes

Fig. 13 Results of text line extraction when video frames were handled. All text lines were well detected

were embedded, many incorrect regions would be extracted and lead to may false alarms. As to our method, even though the processed text lines have different fonts and sizes, it still works very well to detect all desired text lines.

Figure 16 shows another set of performance comparison when skewed text lines were handled. Fig. 16a is the detection result using Hasan and Karam's approach [10]. Their verification method is easily affected by noise and does not deal with fragmented text lines. Therefore, text fragments tended to be detected. Figure 16b is the result obtained using Wong and Chen's method [11]. Their method used a horizontally scanning method to search all regions with local maximum gradients as potential text candidates. The local maximum gradient is very sensitive to complicated background and text orientations. Therefore, although the text lines were correctly located, many false alarms were also detected. However, since our method is invariant to text orientations, all desired text lines were successfully detected using our approach.

For comparing the accuracy of our approach to other methods, we collected 254 normal text lines and 106 skewed text lines in the simple background category. In addition, 52 normal text lines and 49 skewed text lines were collected in the cluttered background category. Table 1 shows the overall accuracy comparisons among the above three methods. Since Hasan and Karam's method did not well deal with fragmented text lines, their method performed the worst among the three methods. As to Wong and Chen's approach [11], they used a line scanning method to find pixels having maximum gradients as potential text lines. Many false alarms would be caused when complicated backgrounds or skewed text lines were handled. Therefore, they had a lower accuracy of text line detection than our proposed approach. Table 2 summaries the average precision and recall analysis among these three approaches. Hasan and Karam [10] use the density of text pixels to verify each possible text regions. The criterion is not robust when images with texture backgrounds are handled. Under this circumstance, a higher threshold should be set for filtering out most of impossible text lines and thus the detection accuracy can be maintained highly. However, this setting will also cause a higher rejection rate. Consequently, in Table 2, their method got a lowest recall rate

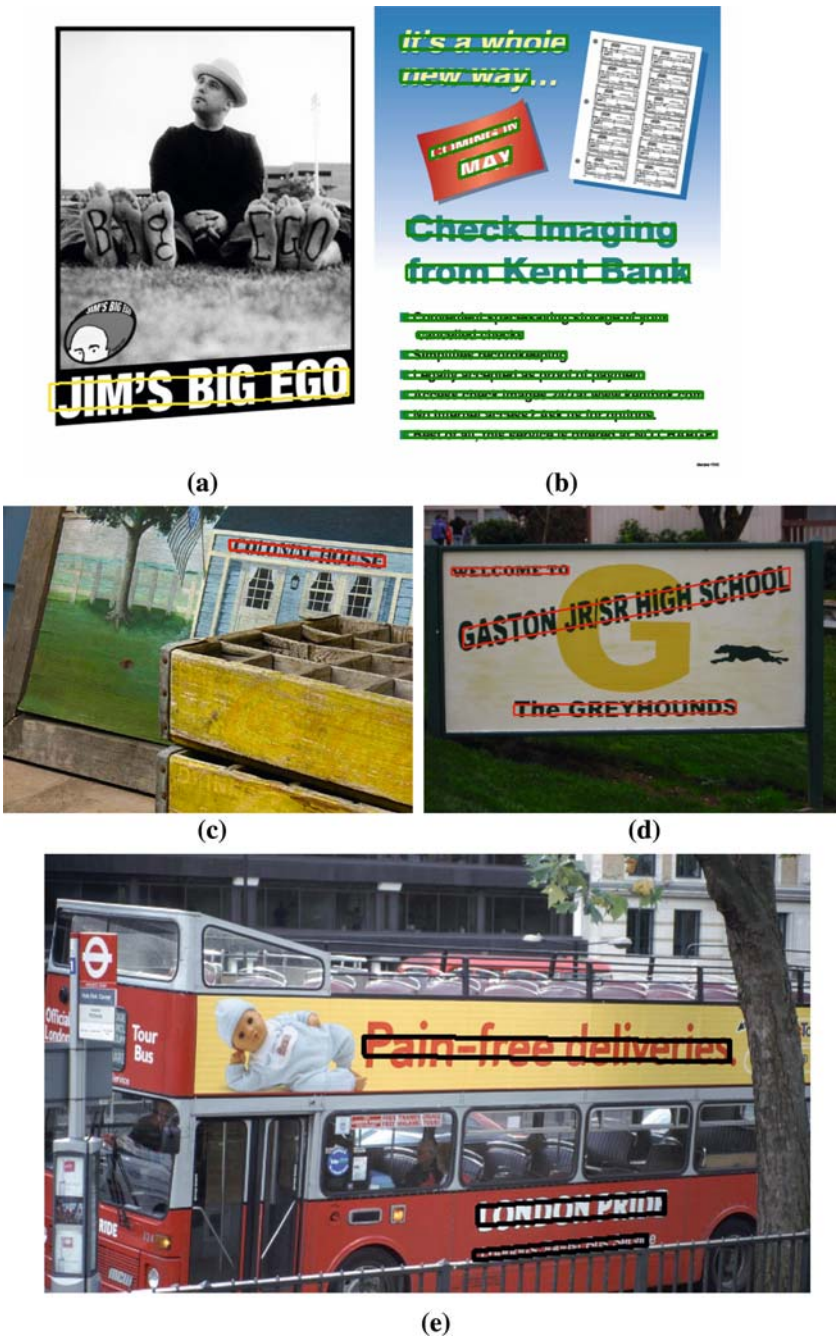


Fig. 14 Results of text line extraction when skewed text lines were handled

among these three methods. The recall rate can be increased if a lower threshold is chosen. However, a lower threshold also leads to a lower precision rate. As to Wong and Chen's [11] approach, their method is vulnerable to cluttered backgrounds and skewed text lines. Therefore, they had a lower precision and recall rates than our method. In this experiment, our method performed the best among all the above comparisons. The average accuracy of our proposed system is 95.4%.

In addition to the recall-precision analysis, we also use the well-known probability of error (PE) to estimate the error rate of each method with the form:

$$PE = P(T) \times P(B|T) + P(B) \times P(T|B),$$

where $P(B|T)$ is the error probability in classifying text regions as background, and $P(T|B)$ the error probability in classifying background regions as text lines. $P(T)$ and $P(B)$ represent the prior probabilities of text line and background, respectively. Table 3 summaries the PE values of the three methods. Clearly, our method got the lowest error rates. All the experiments have proved the superiority of our proposed method in text line detection.

Fig. 15 Comparison results of text line detection when the image includes multiple fonts and sizes. **a** Result of Hasan and Karam [10]. **b** Result of Wong and Chen [11]. **c** Result of our proposed method

Fig. 16 Comparison result of text line detection when the image had cluttered background and skewed text lines. **a** Result of Hasan and Karam [10]. **b** Result of Wong and Chen [11]. **c** Result of our proposed method

Table 1 Comparison results among three text line extraction algorithms

Methods	Counts							
	Simple background images				Cluttered background images			
	No. of exact text lines		No. of detected text lines		No. of exact text lines		No. of detected text lines	
	Norm.	Skew	Norm.	Skew	Norm.	Skew	Norm.	Skew
Our method	254	106	244	99	52	49	50	47
Hasan [10]	254	106	166	56	52	49	17	29
Wong [11]	254	106	232	101	52	49	50	34

Table 2 Recall and precision analysis and comparison

Methods	Evaluation			
	Simple background images		Cluttered background images	
	Recall (%)	Precision (%)	Recall (%)	Precision (%)
Our method	95.3	99.4	96.0	95.1
Hasan and Karan [10]	61.6	95.7	45.5	79.3
Wong and Chen [11]	92.5	79.1	83.2	48.6

Table 3 PE performance analysis

Methods	Probability of error
Our method	0.022
Hasan and Karan [10]	0.226
Wong and Chen [11]	0.1595

7 Conclusions

In this paper, we have proposed a novel morphology-based text line extraction algorithm to extract text regions even if they are embedded in a cluttered image. Firstly, a morphology-based scheme was proposed for extracting high contrast areas as text line candidates. Then, an x-projection technique was proposed for extracting different text properties from the extracted text-analogue regions. Then, different impossible text regions can be filtered out based on the extracted text properties. Furthermore, a text recovery algorithm was proposed for recovering a fragmented text line to a complete line. The contributions of this paper can be summarized are follows:

1. A morphology-based method was proposed for extracting high contrast areas as text line candidates. The feature

is invariant to different image changes like lighting, rotation, translation, and complicated backgrounds.
2. An x-projection was proposed for extracting different text properties from a text line. Since the projection was performed adaptively according to text line orientation, text lines even skewed can be well detected and verified.
3. A recovery algorithm was proposed for reconstructing a complete text line from its fragmented segments. Thus, the proposed scheme has high tolerances to noise.

The average accuracy of the proposed system is 95.4%. Clearly, the proposed method has good abilities to detect all kinds of text lines even under cluttered backgrounds.

References

1. Jung, K., Kim, K.I., Jain, A.K.: Text information extraction in images and video: a survey. Patt. Recognit. **37**(5), 977–997 (2004)
2. Dekun, Z., Shi, Y.Q.: Formatted text document data hiding robust to printing, copying and scanning. IEEE Int. Sym. Circuits Syst. **5**, 4971–4974 (2005)
3. Smith, M.A., Kanade, T.: Video skimming for quick browsing based on audio and image characterization. Technical Report CMU-CS-95–186, Carnegei Mellon University, July 1995
4. Sato, T., Kanade, T., Hughes, E.K., Smith, M.A.: Video OCR for digital news archive. 1998 IEEE International Workshop on Content-Based Access of Image and Video Database, pp. 52–60, Bombay India, 1998
5. Lyu, M.R., Song, J., Cai, M.: A comprehensive method for multilingual video text detection, localization, and extraction. IEEE Trans. Circuits Syst. Video Technol. **15**(2), 243–255 (2005)
6. Zhang, N., Tao, T., Satya, R.V., Mukheriee, A.: Modified LZW algorithm for efficient compressed text retrieval. In: Proceeding of International Conference on Information Technology, Coding and Computer, pp. 224–228 (2004)
7. Hoogs, A., Mundy, J., Cross, G.: Multi-modal fusion for video understanding. In: Proceeding 30th Applied Imagery Pattern Recognition, pp. 103–108 (2001)
8. Zhong, Y., Karu, K., Jain, A.K.: Locating text in complex color images. Patt. Recognit. **28**(10), 1523–1536 (1995)
9. Lienhart, R., Stuber, F.: Automatic text recognition in digital videos. In: Proceeding of SPIE, pp. 180–188 (1996)
10. Hasan, Y.M.Y., Karam, L.J.: Morphological text extraction from images. IEEE Trans. Image Process. **9**(11), 1978–1983 (2000)
11. Wong, E.K., Chen, M.: A new robust algorithm for video text extraction. Patt. Recognit. **36**(6), 1397–1406 (2003)
12. Sin, B., Kim, S., Cho, B.: Locating characters in scene images using frequency features. In: Proceedings of International Conference on Pattern Recognition, vol. 3, Canada, pp. 489–492 (2002)
13. Mao, W., Chung, F., Lanm, K., Siu, W.: Hybrid Chinese/English text detection in images and video frames. In: Proceedings of International Conference on Pattern Recognition, vol. 3, Canada, pp. 1015–1018 (2002)
14. Kim, K.I., Jung, K., Park, S.H., Kim, H.J.: Support vector machine-based text detection in digital video. Patt. Recognit. **34**(2), 527–529 (2001)
15. Xiangrong, C., Yuille, A.L.: Detecting and reading text in natural scenes. In: Proceeding of the IEEE Computer Vision and Pattern Recognition, vol.2, pp. 366–373 (2004)
16. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Chapman & Hall, London (1993)
17. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C (1992)

Author Biographies

Jui-Chen Wu was born in Taiwan, R.O.C., on November 22, 1976. She received the B.S. degree from Chung Yuan Christian University and the M.S. degree from Yuan Ze University, Chung-Li, Taiwan, in 2000 and 2002, respectively. She is currently working toward the Ph.D. degree in the Department of Electrical Engineering, Yuan Ze University, Chung-Li, Taiwan, R.O.C. Her research interests include image processing, pattern recognition, and computer vision.

Jun-Wei Hsieh received the B.S. degree in computer science from TongHai University, Taiwan, R.O.C., in 1990, and the Ph.D. degree in computer engineering from the National Central University, Chung-Li, Taiwan, in 1995. From 1996 to 2000, he was a Researcher Fellow at the Industrial Technology Researcher Institute, Hsinchu, Taiwan, where he managed a team to develop video-related technologies. He is presently an Associate Professor at the Department of Electrical Engineering, Yuan Ze University, Chung-Li. His research interests include content-based multimedia databases, video indexing and retrieval, computer vision, and pattern recognition. Dr. Hsieh received the Phai-Tao-Phai and the Best Paper Awards for the IPPR Conference on Computer Vision, Graphics, and Image Processing, 2005 and 2006, respectively.

Y.-S. CHEN was born in Taiwan, on 30 June 1961. He received BS degree from Chung Yuan Christian University in 1983, and the MS and PhD degrees from National Tsing Hua University, Taiwan, in 1985 and 1989, respectively, all in electrical engineering. He received a Best Paper Award from the Chinese Institute of Engineers in 1989, and an Outstanding Teaching Award from the Yuan Ze University in 2005, respectively. He is a member of the IEEE and IPPR of Taiwan, ROC. In 1991, he joined the Electrical Engineering Department of the Yuan-Ze Institute of Technology, Taoyuan, Taiwan, ROC, where he is now a Professor. Since 1998, his name is listed in the Who's Who of the World. His interests include human visual perception, computer vision, circuit system and teaching web design.