Available online at www.sciencedirect.com
SCIENCE DIRECT®

Image and Vision Computing 23 (2005) 565–576

www.elsevier.com/locate/imavis

Fast and robust text detection in images and video frames

Qixiang Ye[a,*], Qingming Huang[b], Wen Gao[a,b], Debin Zhao[c]

[a]*Institute of Computing Technology, Chinese Academy of Sciences, China*
[b]*Graduate School, Chinese Academy of Sciences, China*
[c]*Department of Computer Science, Harbin Institute of Technology, China*

Received 12 April 2004; received in revised form 12 October 2004; accepted 14 January 2005

Abstract

Text in images and video frames carries important information for visual content understanding and retrieval. In this paper, by using multiscale wavelet features, we propose a novel coarse-to-fine algorithm that is able to locate text lines even under complex background. First, in the coarse detection, after the wavelet energy feature is calculated to locate all possible text pixels, a density-based region growing method is developed to connect these pixels into regions which are further separated into candidate text lines by structural information. Secondly, in the fine detection, with four kinds of texture features extracted to represent the texture pattern of a text line, a forward search algorithm is applied to select the most effective features. Finally, an SVM classifier is used to identify true text from the candidates based on the selected features. Experimental results show that this approach can fast and robustly detect text lines under various conditions.

Keywords: Text detection; Multiscale wavelet feature; Feature combination; SVM classification

1. Introduction

Images and videos on webs and in databases are increasing. It is a pressing task to develop effective methods to manage and retrieve these multimedia resources by their content. Text, which carries high-level semantic information, is a kind of important object that is useful for this task. For example, text in web images can reflect the content of the web pages. Text on book and journal covers can be helpful to retrieve these digital resources [1]. Caption text in news videos usually annotates information on where, when and who of the happening events [2]. Sub-title in sport videos often annotates information of score, athlete and highlight. Scene text often suggests the presence of a fact such as advertisement board, traffic warning, etc. Compared with the other image features, text is embedded into images or scenes by human, which can directly reveal the image content in a certain point of view without requiring complex computation. Therefore, it has inspired a lot of research on text detection and recognition in images and videos [1–15].

The role of text detection is to find the image regions containing only text that can be directly highlighted to the user or fed into an optical character reader module for recognition. It is an essential step for text recognition. In some cases, text detection becomes even meaningful by itself. For example, finding the appearance of a caption in news video can help to locate the beginning of a news item.

Although a lot of approaches have been developed on text detection in real applications [1–15], fast and robust algorithms for detecting text under various conditions need to be further investigated. Here 'robust' is referred to the following criteria:

(1) Good performance (high recall rate and low false alarm rate) for text of various font-size, font-color, orientations, languages and in complex background;
(2) Constant performance with minimized need for manually fine-tuning the parameters or rebuilding a model.

* Corresponding author. Address. Ke Xue Yuan South Road, Zhong Guan Cun, Hai Dian District, P.O. Box: 2704#, 100080 Beijing, China. Tel.: +86 10 82612763 801; fax: +86 10 58858301.
E-mail address: qxye@ict.ac.cn (Q. Ye).

0262-8856/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.01.004

To develop a fast and robust text detection algorithm is a nontrivial task since there exist such difficulties as:

(1) Text may be embedded in complex background;
(2) It is difficult to find effective features to discriminate text with other text-like things, such as leaves, window curtains or other general textures;
(3) Text pattern varies with different font-size, font-color and languages;
(4) Text quality decreases due to noise and image encoding/decoding procedure.

Considering all the above issues, we present a text detection algorithm by fully taking advantage of the following text properties:

(1) Dense intensity variety;
(2) Contrast between text and its background;
(3) Structural information;
(4) Texture property.

In the algorithm, instead of classifying an image block into text or nontext block by supervised classification models [6,12], a new coarse-to-fine detection framework is proposed by applying different text properties in different detection stages. In the coarse detection, candidate text regions are firstly obtained using property (1) and (2) by assuming that all text regions have dense intensity variety and contrast with its background. And then these regions are separated into text lines using property (3). In the fine detection, property (4) is used to discriminate text with other nontext patterns whose dense intensity variety is similar to that of text. We extract four kinds of texture features to identify text lines from the candidate ones including wavelet moment features, wavelet histogram features, wavelet co-occurrence features and crossing count histogram features. A feature selection algorithm is used to find effective features and an SVM classifier is employed to perform text/nontext classification task. Fig. 1 is the flow chart of the proposed algorithm.

Compared with existing approaches, the main advantages of the proposed algorithm are as follows:

(1) *Fast detection*. The method is in a coarse-to-fine framework, which avoids performing computational intensive classification on the whole image block by block. The feature selection procedure reduces feature dimension and improves the classification efficiency.
(2) *Multiscale text detection*. The pattern of text may vary a lot with the change of text font-size. In this paper, multiscale wavelet features are extracted and used to ensure that text with different font-size be correctly detected without performing the scale down operation on the original image. Although Li et al. [12] also used the wavelet features for text detection, they calculated the mean, second- and third-order central moment features in the first three decomposition levels (scales) for text blocks detection while in our method we detect text by using the wavelet features in the suitable scale (as explained in Section 4.1).
(3) *Combination of texture features*. In previous works, researchers use the traditional texture features to discriminate text with nontext patterns. In this paper, four kinds of features are combined to do this task. A forward feature selection algorithm is employed to find the degree of importance for different features.
(4) *Truly robust text detection*. The method can detect text in different sizes and colors. It is also insensitive to text line orientation by using oriented region growing templates. The fine classification procedure reduces false alarms and makes the method effective even in a complex background. Furthermore, texture features are extracted in a whole text line instead of a small image block. These features are much more robust than features from a small image block that may contain insufficient texture information for classification.

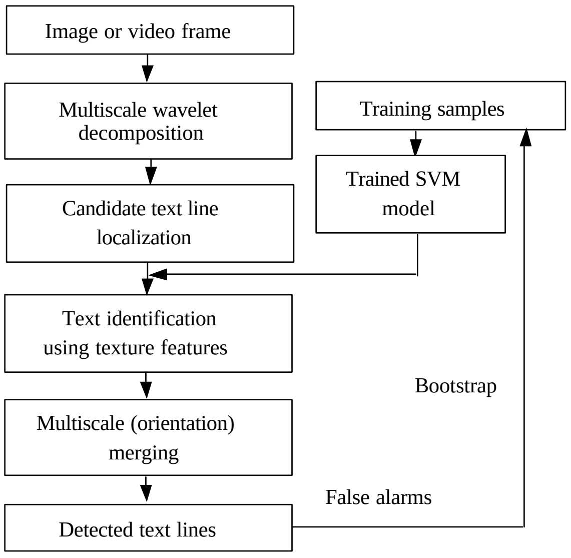
Image or video frame
Multiscale wavelet decomposition
Candidate text line localization
Text identification using texture features
Multiscale (orientation) merging
Detected text lines
Training samples
Trained SVM model
Bootstrap
False alarms
Fig. 1. Flow chart of the proposed method.

The rest of the paper is organized as follows. Section 2 is a literature survey of related works. We present the coarse detection procedure in Section 3 and describe the fine detection process in Section 4. Experimental results are presented in Section 5. The paper is concluded with a discussion of future work in Section 6.

2. Related works

Dozens of text detection algorithms are presented in the past years using the text properties described in Section 1. Various kinds of features are explored to capture these properties.

Edge (gradient) is a preferred feature. In [3], Smith et al. detect text by finding text box 'horizontal rectangle structure of clustered sharp edges'. Their algorithm is scale dependent, that is, only text with certain font-size can be detected. Sato et al. [4] also use edge features and structural constraint to detect captions in video frames. Wu et al. [5] use nine second-order Gaussian derivatives to extract vertical strokes in horizontal aligned text regions. Strokes are connected into 'chip' if they are connectable and there exists a path between them whose length is less that three times the height of the stroke. The chip will be further checked by structural properties like values of height, width and width/height. Lienhart and Wernicke [6] locate text in images and video frames using the image gradient feature and a neural network classifier. Chen et al. [7] and Ye et al. [8] use the Canny edge feature and morphological 'close' operation to detect candidate text blocks. Chen et al. [7] use the 'dist-map' feature to verify the candidate text to reduce the false alarms. The above methods often have a high recall rate but produce many false alarms since background blocks may also have strong edge (gradient) just as text does. And pure structural property is not competent for eliminating false alarms.

Jain and Yu [9] propose a classical text detection algorithm based on connected component analysis. The method can detect text in web images, video frames and some document images such as newspapers. In the algorithm, after the original image is decomposed into a multi-value map, connected components of color are selected, which will be regarded as text if they are accepted by any of the following strategies: (1) inter-component features (geometry and connection property), (2) projection profile features in both horizontal and vertical orientations. The probability of missing text is minimized at the cost of increasing false alarms. Zhong et al. [10] also use the connected component analysis to locate text in complex color images. Although the method is a classical one and is effective for most of the text detection tasks, it fails when there are characters of different colors in a text line. In addition, the authors do not mention the scale problem.

Texture analysis is employed in some works to discriminate text with nontext [1,11–13]. Li et al. [11,12] use mean, second- and third-order central moments in wavelet domain as the texture features and a neural network classifier is applied for text block detection. In their detection results, small isolated areas are filtered out and large text blocks are connected into text regions. Zhong et al. [1] detect text in JPEG/MPEG compressed domain using texture features from DCT coefficients. They first detect blocks of high horizontal spatial intensity variation as text candidates, and then refine these candidates into regions by spatial constraints. The potential caption text regions are verified by the vertical spectrum energy. But its robustness in complex background may not be satisfying for the limitation of spatial domain features. Kim et al. [13] propose a texture-based method using support vector machine (SVM). Classification models trained by an SVM on original gray values of a pixel's neighborhood are used to identify a pixel as text or nontext pixel. Text pixels will be post-processed by an adaptive mean shift (CAMSHIFT) algorithm and connected into text chips. Although the SVM-based learning approach makes the algorithm fully automatic, it is still difficult to discriminate text with nontext using pure texture features in complex background since the feature is insufficient to discriminate text with general textures. In [6,11–13], classification on each of the pixel (image block) will be a time-consuming task.

Some researchers use video temporal information to detect video captions by considering that the appearance of a caption will bring on difference in successive frames. Tang et al. [14] first filter the difference of successive frames by four edge operators and then divide the output into K blocks. Furthermore, they use the means and variances of these filtered blocks and a fuzzy clustering neural network (FCNN) classifier for text identification. Although temporal information can be used to track text [12] and enhance its quality for recognition [15], it may not be a good feature for text detection because precise shot boundary detection is needed before identifying the appearance of a text line, which will increase the computation complexity. Furthermore, if a text line appears on a shot boundary, temporal information will fail to detect it.

Based on the above discussion, we can see that although a lot of research has been carried out in text detection, more robust and effective method still needs to be developed.

3. Coarse detection

Coarse detection is to find all of the possible text lines in an image. In this procedure, the wavelet energy feature is used to locate candidate pixels and a new region growing method is applied to connect the candidate pixels into regions. Text in non-horizontal orientations is detected by using orientated region growing templates. Non-horizontal candidate will be rotated into the horizontal orientation for further processing. Obtained regions are then separated into text lines by structural property.

3.1. Multiscale wavelet decomposition

Daubichie4 wavelet transformation [16] is employed in our work for its good location performance [17], which is computed by applying separable filter banks on the image as

$$\begin{aligned}
I_n(b_i,b_j) &= [G_x*[G_y*I_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(b_i,b_j)\\
D_{n1}(b_i,b_j) &= [H_x*[G_y*I_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(b_i,b_j)\\
D_{n2}(b_i,b_j) &= [G_x*[H_y*I_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(b_i,b_j)\\
D_{n3}(b_i,b_j) &= [H_x*[H_y*I_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(b_i,b_j),
\end{aligned} \quad (1)$$

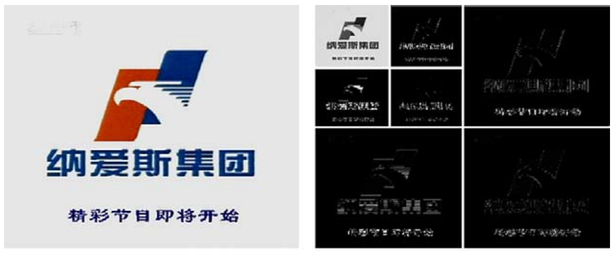

Fig. 2. Two-level wavelet transformation. (a) The original image with text in different font-size and (b) the wavelet transformation images in two levels.

where * denotes the convolution operator, $(\downarrow_{1,2})$ sub-sampling along rows (columns) and I_0 is the original image, H and G are high and low bandpass filters, respectively. b_i and b_j are the locations in two directions in a decomposition level. $\{I_n, D_{nk}\}_{k=1,2,3,\ n=1,2,\ldots,l}$ is the multiscale representation of depth l of the original image. I_n are the low resolution images at level n, D_{nk} are the wavelet coefficients obtained by bandpass filtering which contain the intensity variety information in level n. Fig. 2 shows that text with different sizes are emphasized in different decomposition levels (scales). It also strongly implies that text in different sizes can be extracted from the different levels of the wavelet decomposition images.

3.2. Candidate text pixels detection

Considering intensity variety (property 1 in Section 1) around the text pixels, the wavelet coefficients around the pixels should have large values (as shown in Fig. 3(b)). Then, we define the wavelet energy feature of a pixel at (i, j)

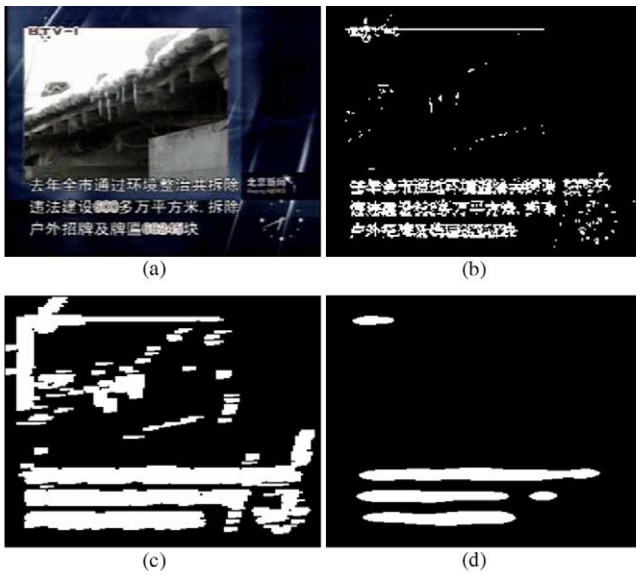

Fig. 3. Candidate text region detection. (a) Original image, (b) candidate pixels in the first scale, (c) horizontal candidate regions in first level by 'close' operation and (d) by density-based region growing method.

in level n as

$$E_n(b_i, b_j) = \left(\sum_{k=1}^{3} [D_{nk}(b_i, b_j)]^2 \right)^{1/2} \tag{2}$$

for candidate pixels detection by integrating the wavelet coefficients in the three high frequency subbands (LH, HL and HH subbands). The wavelet energy feature $E_n(b_i, b_j)$ reflects the intensity variety around a pixel in level n. A pixel will be a candidate text pixel in level n if its wavelet energy feature is larger than a dynamic threshold, which is described as

$$C_n(i,j) = \begin{cases} 1 & \text{if } (E_n(b_i, b_j)) > T_C \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where $C_n(i, j)$ is the map of candidate text pixels in level n (as shown in Fig. 3(b), candidate pixels in the first level are projected into original image). T_C is a threshold determined as

$$T_C = \begin{cases} t_B & \text{if } t_B > t_C, \\ t_C & \text{otherwise,} \end{cases} \tag{4}$$

where t_B is the basic threshold whose value is set to be 30.0 which is proved to be the minimum value that a text pixel can be perceived by human from its background. t_C is determined by energy histogram $(H(i))$ curve analysis of the whole image (Fig. 4). The value of t_C is calculated by

$$\sum_{i=t_C}^{\mathrm{WE_{max}}} H(i) \Big/ \sum_{j=0}^{\mathrm{WE_{max}}} H(j) = t_{\mathrm{Area}}. \tag{5}$$

In Eq. (5), t_C ensures that pixels whose histogram lies in shadow area (Fig. 4) be detected as candidate pixels. Empirically, we found that text pixels in an image rarely exceed 15% of the whole image pixels. Therefore t_{Area} is selected as 0.15. The determination of T_C in this way can ensure that the candidate detection works robust in image of different contrast. For the image of low contrast, t_B is selected for T_C, which ensures that most of the background pixels be excluded. With the increment of image contrast and complexity, t_C is selected and adaptively calculated. And the larger the image contrast is, the bigger t_C should be. Pixels whose contrast is higher than T_C are selected as candidates.

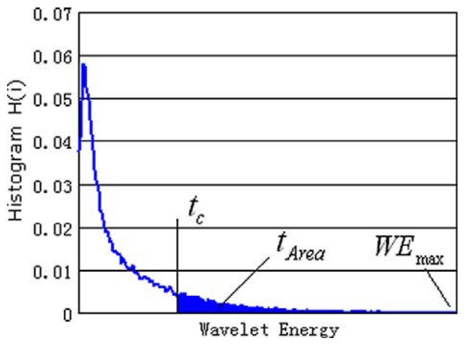

Fig. 4. Wavelet energy histogram of an image.

3.3. Density-based region growing

A text region is made of a 'cluster' of text pixels. None but 'dense' text pixels can construct a text region and the isolated candidate pixels are often noises. Morphological operation such as 'close' operation is often used to connect text pixels (blocks) into text regions [2,7,8]. In the operation, all of the pixels near to each other will be connected despite whether they form a 'cluster' of text pixels or not (Fig. 3(c)). In this paper, 'density-based' region growing method is proposed to fulfill this task.

A pixel P will be a seed pixel if the percentage of candidate pixels in its neighborhood is larger than the threshold T_D. The neighborhood is a 16×10 template in all levels. To find text lines in non-horizontal orientations, horizontal template is rotated by 30 degrees each time to get six templates (Fig. 5). T_D is set as 0.35 in our experiments. A pixel P' is considered to be density-connected with pixel P if P' is within the neighborhood of P and P is a seed pixel. By these definitions, the region growing method is described as follows:

(1) Search the unlabeled candidate pixels to find a seed pixel;
(2) If a seed pixel P is found, a new region is created. Then, we iteratively collect unlabeled candidate pixels that are density-connected with P, and label these pixels with the same region label;
(3) If there are still seed pixels, goto (1);
(4) Label each found region as a text region. Merge the pixels that are not included in any text region with the background.

Fig. 3(d) is an example of text regions found by the proposed 'density-based' region growing algorithm in the first scale and horizontal orientation. It can be seen that the result is much better than that of morphological algorithm (Fig. 3(c)).

The orientation of text lines in a candidate region can be considered to be consistent with the orientation of region growing template as illustrated in Fig. 5. Therefore, candidates in non-horizontal orientation can be rotated to (approximately) horizontal orientation for further processing. In the following, we only need to process the (approximately) horizontal text.

3.4. Getting candidate text lines

Many detected text regions contain multi-line text. A projection profile operation [6] is employed to separate

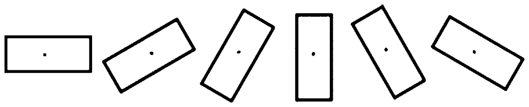
Fig. 5. Region growing templates for text in different directions.

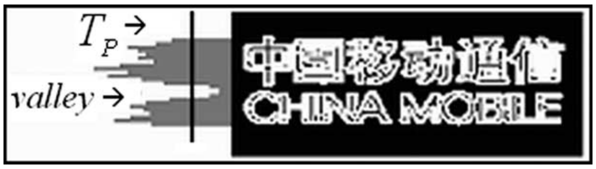

Fig. 6. An example of horizontal projection profile. Multi-line text can be separated at the 'valley' of the project profile.

these regions into text lines. A horizontal projection profile is defined as the sums of the candidate pixels over rows. Fig. 6 shows an example of projection profile of a candidate pixel map. To separate the two text lines in the example, we need to find the 'valley' on the profile where the profile value is smaller than a threshold T_P and then segment the two text lines at the valley. T_P is calculated as

$$T_P = (\mathrm{Avg}_{\mathrm{profile}} + \mathrm{Min}_{\mathrm{profile}})/2.0, \tag{6}$$

where $\mathrm{Min}_{\mathrm{profile}}$ and $\mathrm{Avg}_{\mathrm{profile}}$ are the minimum value and average value of the profile, respectively. This scheme can also be extended to more than two text lines separation.

For each of the candidate text lines, simple structural information will be used to reduce the apparent false alarms. It is observed that text height should be larger than 8 pixels to be seen clearly by human. The candidate whose height is smaller than 8 pixels or width/height <1.0 is discarded as nontext.

4. Fine detection

All image blocks with abrupt intensity variation may be falsely detected as candidate text lines, especially for general textures such as window curtains, leaves, etc. In the following, texture features are extracted to identify true text from the candidate ones.

4.1. Feature extraction in suitable scale

Text can be regarded as a kind of texture pattern, but the texture properties such as regularity and directionality are weak. It just contains some character strokes that form a text line in a special orientation. Then, only one kind of texture feature is insufficient to model text's texture pattern. In this paper four kinds of features are combined to represent a text line, three of which are extracted in wavelet domain and one in gradient image. The gradient features are extracted in the original image and the wavelet texture features are extracted in the level where the candidate text lines are located, say the suitable scale.

Text of large font-size can be seen as a relatively low frequency signal while text of small font-size as a relatively high frequency signal. According to the wavelet decomposition theory [16], low frequency signal will have similar wavelet response in deep wavelet decomposition scale as

Table 1
Features used for text/nontext classification

Feature set	Feature description	Number of features	Number of feature selected
Wavelet moments features	Mean, second-, third order moments	9	6
Wavelet histogram features	Wavelet energy histogram and direction histogram	20	10
Wavelet co-occurrence features	Energy, entropy, homogeneity and correlation	180	16
Scan line features	Crossing count histogram	16	9
Total		225	41

high frequency signal in shallow scale. Therefore, representing text wavelet response in suitable scale will unify the text pattern and reduce the burden on selecting representative training samples for text of different font-size. If a text line has the most candidate pixels (projected to original image) in scale n, it can be considered to have the best wavelet response in this scale. Then n is considered to be the suitable scale in which texture features are extracted to represent this text line.

The descriptions of the features are listed in Table 1.

(1) *Wavelet moment features*. Text and nontext have different intensity variance and spatial grey value distributions. Wavelet mean and central moment features are extracted to reflect this character. The features are the primary texture features in wavelet domain. Li et al. [12] have proved the effectiveness of these features in text block identification. For a $M\times N$ text line T, the mean (m), second-order (μ_2) and third-order (μ_3) central moments are calculated as

$$m(T)=\frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}T(i,j),$$
$$\mu_2(T)=\frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(T(i,j)-m(T))^2, \tag{7}$$
$$\mu_3(T)=\frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(T(i,j)-m(T))^3,$$

where $T(i, j)$ represents wavelet coefficients of pixel (i, j). The features in (7) are calculated in the high frequency (HL, LH and HH) subbands. There are totally 9 features (3 subbands$\times$1 level$\times$3 features).

(2) *Wavelet histogram feature*. Histogram is a kind of effective feature to represent the first order distribution of signatures. Two kinds of histogram features, wavelet energy histogram (WEH(i), $i=0,\ldots,15$) and wavelet direction histogram (WDH(i), $i=0,1,2,3$) are used to represent the energy and direction distribution of a text line. To calculate the WEH(i), wavelet energy of all pixel is quantized into 16 levels by

$$\mathrm{WE}_q=\mathrm{WE}\times 16/(\mathrm{WE}_{\max}-\mathrm{WE}_{\min}), \tag{8}$$

where WE is the wavelet energy of a pixel, $\mathrm{WE}_{\max}$ and $WE_{\min}$ are the maximum and minimum energy values of the image, respectively. We then compute the wavelet energy histogram on the quantized energy value. The value of WEH(i) is the percentage of the pixels whose quantized energy is equal to i. For a text line, the bins at the front and tail of WEH(i) should be large while bins in the middle parts of WEH(i) should be small (Fig. 7(a)). This is caused by the contrast between text and its background. For a general texture, the WEH(i) may not be so (Fig. 7(b)).

WDH(i) contains horizontal ($i=1$), vertical ($i=2$), dialog ($i=3$) and non-direction ($i=0$) bins. We declare that a candidate pixel has a horizontal (vertical, dialog) direction in level n when D_{n1}(D_{n2}, D_{n3}) is the largest one among $\{D_{nk}\}_{k=1,2,3}$. All non-candidate pixels will be non-direction pixels. The value of the bin in direction i is the percentage of pixels on the direction compared with the number of all pixels in the text line. WDH(i) can be used to discriminate text with general textures with distinct directionality as shown in Fig. 7(b).

(3) *Wavelet co-occurrence features*. Histogram features are first order statistics. When the first order statistics are insufficient, second order statistics can be used to improve texture discrimination power by describing correlations among adjacent pixels [18]. Co-occurrence features for three high frequency subbands are used. The element (i, j) of the co-occurrence matrix $C(d, \theta)$ is defined as the joint probability of a wavelet coefficient $D_{nk}=i$ co-occurring with a coefficient $D_{nk}=j$ on a distance d in the direction θ [19].

The features of energy, entropy, inertia, local homogeneity and correlation in co-occurrence matrixes are

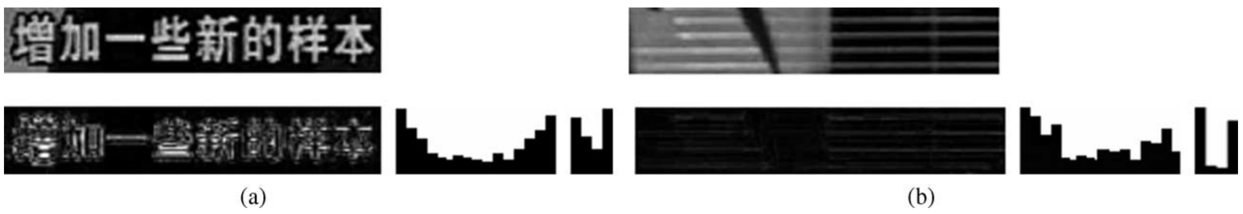

Fig. 7. The first row in (a) is a text image, the second row is its wavelet energy map and the WEH and WDH histograms. The first row in (b) is a nontext image, the second row is its wavelet energy map and the WEH and WDH histograms.

calculated as

$$\text{Energy}: \quad E(d,\theta) = \sum_{i,j} C^2(d,\theta),$$

$$\text{Entropy}: \quad H(d,\theta) = \sum_{i,j} C(d,\theta)\log C(d,\theta),$$

$$\text{Inertia}: \quad I(d,\theta) = \sum_{i,j} (i-j)^2 C(d,\theta),$$

$$\text{Local homogeneity}: \quad L(d,\theta) = \sum_{i,j} \frac{1}{1+(i-j)^2} C(d,\theta),$$

$$\text{Correlation}: \quad C(d,\theta) = \frac{\sum_{i,j}(i-\mu_x)(j-\mu_y)C(d,\theta)}{\sigma_x\sigma_y}, \tag{9}$$

where μ_x, μ_y and σ_x, σ_y are means and variances of the concurrence matrix $C(d,\ \theta)$. θ is the direction of co-occurrence that is selected as 0, 45, 90 and 135 degree. d is the co-occurrence distance, which is set to be 1, 3 and 5 pixels in this paper. By calculating features in the 12 co-occurrence matrixes in 3 wavelet subbands (HL, LH and HH), we obtain (3 subbands $\times$ 12 co-occurrences $\times$ 5 features) 180 co-occurrence features.

(4) *Crossing count histogram feature*. The above-mentioned features do not consider the periodicity of text along the text line. For the limited length of a text line, frequency analysis is not suitable for capturing the periodicity. Therefore, normalized crossing count histogram (CCH) on gradient projection map (GPM) is used to do this task. CCH is the histogram statistics of crossing count of all horizontal scan lines. To calculate CCH, gradient values (Fig. 8(b)) are first projected into one dimension data along the horizontal direction to obtain gradient projection map (Fig. 8(c)). After Gaussian smoothing, it can be seen that there are regularity and periodicity in the GPM. A crossing

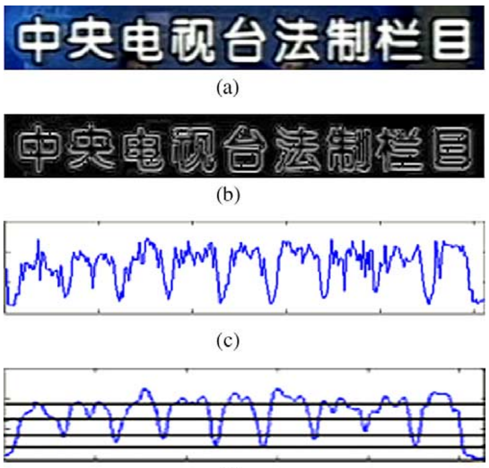

Fig. 8. Gradient crossing count histogram features. (a) The original image, (b) the gradient image, (c) the gradient projection along horizontal direction and (d) the smooth gradient projection with several scan lines.

count is calculated as the number of times the pixel value in GMP changes from 0 (white) to 1 along a horizontal raster scan line (Fig. 8(d) shows an example containing four scan lines). Suppose CC(k), $k=1,2,\ldots,N$ is the crossing count of a horizontal scan line k and N is the scan line number. The value of CCH$'(k)$ is calculated as

$$\text{CCH}'(k) = \frac{\text{CC}(k)}{\sum_{i=1}^{N}\text{CC}(i)}. \tag{10}$$

For example, if the maximum of gradient projection is 300, we use $N=300$ lines to scan the GMP and obtain 300 bins CCH$'(k)$, $k=1,2,\ldots,300$. By normalizing CCH$'(k)$ into 16 bins as

$$\text{CCH}(i) = \frac{1}{16}\sum_{k=i\frac{N}{16}}^{(i+1)\frac{N}{16}} \text{CCH}'(k), \quad i = 1,2,\ldots,16, \tag{11}$$

where ($i(N/16)$, $(i+1)(N/16)$) represents a non-overlapped window, we obtain the normalized crossing count histogram.

The CCH can reflect the distribution of the crossing counts of all scan lines and then coarsely reflect the periodicity of GMP on text line orientation.

4.2. Feature selection

We extract a total of 225 features from a text line (Table 1). Although all of these features can be used to distinguish text with nontext, some features may contain more information than others. Using only a small set of the most powerful features will reduce the time for feature extraction and classification. Furthermore, the existing research has shown that when the number of training samples is limited, using a large feature set may decrease the generality of a classifier [20]. Therefore, feature selection is performed before they are fed into the classifier.

For selecting the effective features for SVM classifier (which will be introduced in the next section), a forward search algorithm [20] is used to perform the feature selection task.

The feature set F is first divided into selected feature set F_S and unselected feature set F_U, and then selected one by one using the following procedure:

(1) Set $F_S=\phi$ and $F_U=F$;
(2) Label all of the features in F_U untested;
(3) Select one untested feature f from F_U and label it as tested;
(4) Put f and F_S together to form the temporary testing feature set $\tilde{F}_S$;
(5) Evaluate the classification performance of $\tilde{F}_S$;In this procedure, 1000 positive (text) samples and 2500 negative (nontext) samples are randomly selected from the training set for train and classification tasks. The samples are divided into 10 shares, one is selected for testing and others for training. The operation is repeated until all samples have been used as training

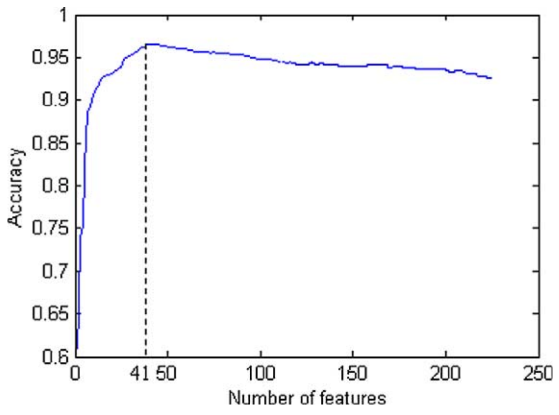

Fig. 9. Feature selection. The best classification result is achieved when 32 features are selected.

and testing samples once. The average accuracy for all iterations is set as the estimated accuracy for the testing feature set. Accuracy is defined as

$$\text{Accuracy} = \frac{\text{Number of correctly classified samples}}{\text{Number of samples}}. \tag{12}$$

(6) If there are still untested features in F_U, goto (3);
(7) Find a feature $\hat{f}$ such that when we add it into the feature set $\tilde{F}_S$, the highest classification accuracy will be obtained

$$\hat{f} = \text{argmax Accuracy}(\hat{F}_S)$$

and then move $\hat{f}$ from F_U to F_S;
(8) If there are still untested features in F_U, goto (2) and if F_U is empty, the procedure exists.

Fig. 9 is the accuracy curve of feature selection. It can be seen on the curve that when the number of selected features increases, the accuracy increases sharply at first but will slightly decrease when the number pass 41. Then the first 41 features should be selected. The selected features are listed in the last column in Table 1. It can be seen that almost all of the wavelet moment features are selected, which shows that these features are the most important features. About half CCH features are selected, which show that they have the middle discriminating power. There are less histogram and co-occurrence features selected. In the following, training and classification are performed on the 41 selected features.

4.3. Training and classification

Compared with other classifiers such as neural network and decision tree (C4.5), SVM is easier to train, needs fewer training samples and has better generalization ability. Considering the limited number of training sample, we use SVM classifier in our work.

SVM was proposed by Vapnik [21] and obtained excellent results in various data classification tasks in recent years especially in two-class problems [22,23], including text detection [13]. Traditional classification techniques use empirical risk minimization and lack a solid mathematical justification. The SVM classifier uses the structural risk minimization to find the hyperplane that optimally separates two classes of objects. The kernel function of SVM is a second polynomial for its better performance than other kernels in this task.

The SVM was trained on a dataset consisting of 3200 text and 8000 nontext labeled samples. Fig. 10 shows some of the training examples. As stated in [22,23], although positive samples are easy to be obtained, it is more difficult to get representative negative samples since they span a vast space. Then, after obtaining a trained model, a 'bootstrap' process (as shown in Fig. 1) is used to improve the performance of the classifier. False alarms will be added into the nontext set for re-training.

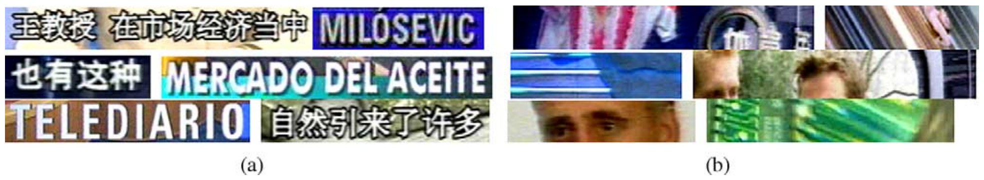

Fig. 10. Examples of the training data, (a) the text data (b) nontext data.

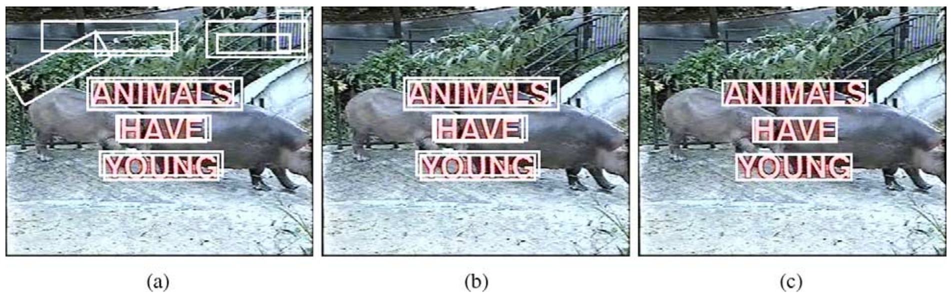

Fig. 11. Text detection results. (a) The coarse detection result, (b) the result after fine detection and (c) the result after multiscale (direction) integration.

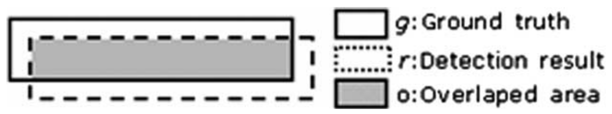

Fig. 12. Illustration of the overlap of a ground truth box and detection result.

Fig. 11 shows an example of coarse and fine detection results. It can be seen that false alarms are reduced in the fine classification process (Fig. 11(b)).

4.4. Multiscale (orientation) merging

Sometimes, a text line can be detected in more than one orientation or scale and the detection results overlap in the original image (Fig. 11(b) is an example). When the overlapped area of two candidates is larger than half area of either of them, a candidate should be deleted. If a text line is detected in two orientations, the one with larger width is kept. If a text line is detected in two scales, the one with larger probability to be text is kept. The probability of a candidate to be text is obtained in the SVM classification process as the method in [24]. Fig. 11(c) shows the result after multiscale (orientation) merging (Fig. 12).

5. Experimental results

Two testsets are used for experiments. One is our Lab testset containing 177 images from Webs, broadcast videos frames and images captured by digital camera. The other is Microsoft common testset containing 44 images [25]. All of the video frames in test have a size of 400×328 pixels. The testsets consist of a variety of cases, including text in different font-size, font-color, directions and languages, light text on dark background, text on textured background, text of poor quality, etc.

The proposed method performs robust on a majority of the test images. Fig. 13 illustrates some examples of the detected text lines. It can be seen from the results that most of the text is well detected despite of large and small font-size, colors and languages. The results show that even in a cluttered background, the proposed method can work well for overlay text and frontal scene text.

Fig. 14 illustrates some failure examples. An isolated character '2' in Fig. 14(a) is not detected for the reason that it does not construct a text line and then does not have the text texture property. Two text lines are missed in Fig. 14(b) because the large affine distortion makes the text outline non-rectangle. A text line of quite small font-size is missed in Fig. 14(c). In the experiments, we find that text whose height is smaller than 9 pixels is prone to be missed. If a text string contains less than two characters, it is probably to be missed. Since most of the text has a height larger than 10 pixels and contains more than two characters, this failure is trivial in practice. In Fig. 14(d), there is a man-made structure falsely detected as text line but we can see that it has all the properties as text, including contrast with background and texture properties. Other man-made structures such as building windows, guardrails, etc. are also prone to be falsely detected as text lines. We predict that these false alarms can be eliminated by the feedback from OCR result (like the recognition rate and repetition rate of characters) in the future work.

Based on the idea of performance evaluation in [26], we define the following simple criteria to evaluate the text

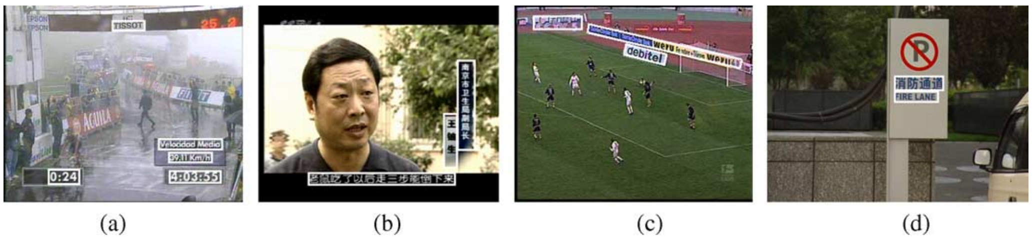

Fig. 13. Examples of detected text lines in images and video frames.

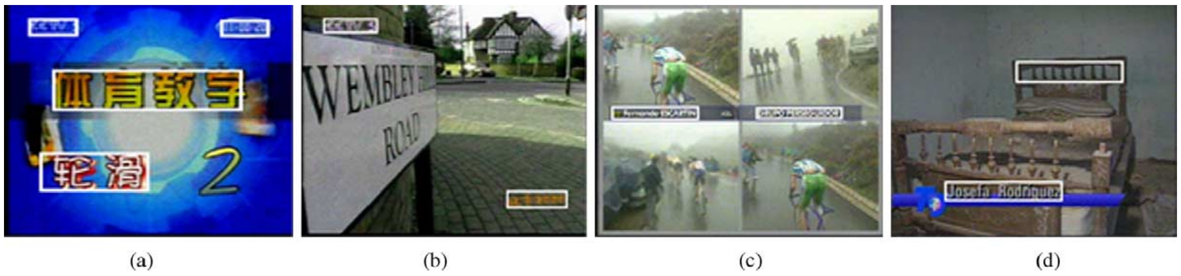

Fig. 14. Images with failures and false alarms. (a–c) Examples containing missed text and (d) contains a false alarm.

detection results

$$\delta(r,g,o) = \begin{cases} 1 & \text{if } o/g > 0.95 \text{ and } o/r\ 0.75 \\ 0 & \text{else} \end{cases} \tag{13}$$

where r, g and o represent the detected text rectangle area, the ground truth area and their overlap area, respectively (as shown in Fig. 12. $\delta(r, g, o)=1$ means that a text line is correctly detected and $\delta(r, g, o)=0$ missed. In (13), $o/g>0.95$ and $o/r>0.75$ means that the overlap area covers more than 95% ground truth box and more than 75% detected rectangle. If o covers no text pixels, a false alarm is produced.

Ground truth is marked by hand from the testsets. There are totally 689 text lines marked in the two testsets for performance evaluation. Given the marked ground truth and detected result by the algorithm, we can automatically calculate the recall and false alarm rate by

$$\text{Recall} = \frac{\text{Number of correctly detected text}}{\text{Number of text}}, \tag{14}$$

$$\text{False alarm rate} = \frac{\text{Number of falsely detected text}}{\text{Number of detected text}}. \tag{15}$$

As listed in Table 2, 94.2% recall rate and 2.3% false alarm rate are reported on our lab testset. 94.1% recall rate and 2.4% false alarm rate are reported on Microsoft testset. The large decrease of false alarm rate from the coarse to fine detection (Table 2) shows the validity of the selected features and SVM-based classification.

The method has an average detection speed of more than 8 images per second on a Pentium IV 1.6G CPU, which means that video text can be detected in real time by sampling several frames per second (supposing that video text will last at least 20 frames to be seen clearly).

In order to provide an idea about the quality of these results, experiments are done on Microsoft common testset to compare the proposed method with two representative methods that are implemented according to [6,12]. It can be seen (Table 3) that our method is faster. Both the recall rate and false alarm rate perform better than that of [12]. Although the recall rate is a little bit lower than that of [6], the false alarm rate (2.4%) is much lower than the method.

To show the advantage of the selected feature set, we compare this feature set with features used in [6,12] by an SVM classifier. Comparison results (Table 4) show that the features used in this paper work better than the wavelet moment features and gradient features. This also shows that the feature combination and selection in this paper is effective.

Experiments are done to compare SVM with decision tree (C4.5) and B-P Neural Network (BPNN). The reason why C4.5 and Neural Network are selected for comparison is that they succeed in lots of classification tasks in recent years and both of them use a 'decision boundaries' to solve a binary-classification problem [20] just as SVM classifier does. Classification accuracy is set as the criteria to evaluate the classifiers. It can be seen in Fig. 15 that SVM performs better than C4.5 on all of the training sets. The performance of BPNN exceeds the SVMs when the number of training text/nontext is larger than 1600/4000, but the performance of SVM is much better than BPNN when the training set is relatively small. This proves that SVM has the best generalization ability in the three compared classifiers in terms of this problem. SVM is suitable for the text/nontext classification problem when it is hard to get a large number of representative training examples.

In tested images, there is text in different languages including Chinese, English, etc. The Chinese text can represent Eastern languages such as Japanese, Korean, etc.

Table 2
Performance of coarse and fine detection

	Recall rate (%)	False alarm rate (%)
Coarse detection	97.4[a]	10.1[a]
	97.2[b]	11.5[b]
Fine detection	94.2[a]	2.3[a]
	94.1[b]	2.4[b]

[a] The result on our testset.
[b] The result on Microsoft common testset.

Table 3
Performance comparison of three algorithms

	Recall rate (%)	False alarm rate (%)	Speed (images/s)
Our algorithm	94.2	2.4	8.3
Algorithm [12]	91.4	5.6	1.5
Algorithm [6]	94.3	8.1	2.2

Table 4
Performance comparison of features

Features	Accuracy (%)
Features used in this paper	96.8
Wavelet moment features used in [12]	93.2
Gradient features used in [6]	94.5

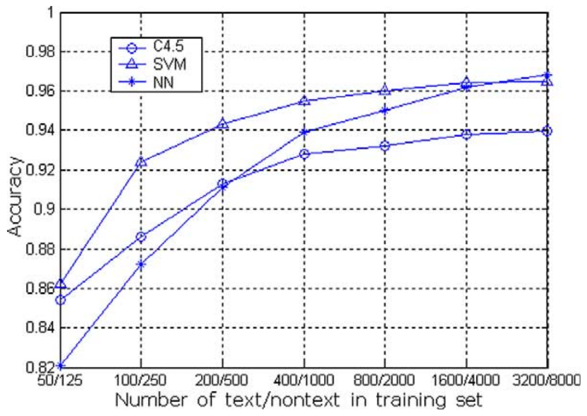

Fig. 15. Performance comparison of classifiers.

Table 5
Performance comparison on different languages

	Recall rate (%)	False alarm rate (%)
Chinese text	94.0	2.4
English text	94.3	2.4
Combining text	94.2	2.4

and the English can represent Western language such as French, German, etc. The recall rate for Chinese and English texts are 94.0% and 94.3%, respectively (Table 5). This shows that our algorithm is insensitive to both Western and Eastern languages. Whereas, performance on English-like Western languages is better because their intensity variety is denser than that of Chinese-like Eastern languages. Furthermore, texture property of English-like text is more distinct because it is commonly made up of limited characters.

6. Conclusions and future work

A new algorithm for detecting text in images and video frames is presented in this paper. The coarse-to-fine detection framework makes the calculation faster compared with previous text detection method in uncompressed domain. The high detection speed will benefit real applications. For example, it will make the text detection really helpful for fast Web content analysis by providing text information in Web images. It will also benefit the real-time video content analysis. The feature selection procedure finds effective texture features to represent the text pattern. The combination of energy, structural and texture properties make the method a truly robust one. The low false alarms rate will ensure the method provide more accurate information for real applications. To the best of our knowledge, this is the first work to compare the text detection performance on different languages. The detection algorithm obtains single-line text instead of text regions containing multi-line text, which will benefit the text recognition in the future work. Although the algorithm is designed mainly for detecting overlay text in images and video frames, it can work properly for most of the frontal scene text.

Currently, we only provide a text detection method. Text should be clearly extracted from its background to obtain a good recognition result for the characters. Temporal information can be used to enhance the quality of video text as in [26–28]. Special technique should be investigated to segment the characters from their background before putting them into an OCR software in the future work as in [29–31]. Domain knowledge [32] should also be integrated into the detection and recognition tasks.

Acknowledgements

The authors would like to thank Dr Shuqiang Jiang for his advice on editing this manuscript and the anonymous reviewers for their constructive comments. This work is partly supported by National Hi-Tech Development Programs of China under grant No. 2003AA142140 and NEC-JDL Context-based Multimedia Analysis and Retrieval Program.

References

[1] K. Sobottka, H. Bunke, H. Kronenberg, Identification of text on colored book and journal covers, International Conference on Document Analysis and Recognition 1999; 57–63.

[2] Y. Zhong, H.J. Zhang, A.K. Jain, Automatic caption localization in compressed video, IEEE Transactions on PAMI 22 (2000) 385–392.

[3] M.A. Smith, T. Kanade, Video skimming for quick browsing based on audio and image characterization, Carnegie Mellon University, Pittsburgh, PA, Technical Report CMU-CS-95-186, July, 1995.

[4] T. Sato, T. Kanade, E.K. Hughes, M.A. Smith, Video ocr for digital news archives, IEEE Workshop on Content Based Access of Image and Video Databases, Bombay, January 1998;.

[5] V. Wu, R. Manmatha, E.M. Riseman, Textfinder: an automatic system to detect and recognize text in images, IEEE Transactions on PAMI 20 (1999) 1224–1229.

[6] R. Lienhart, A. Wernicke, Localizing and segmenting text in images and videos, IEEE Transactions on Circuits and Systems for Video Technology 12 (2002) 256–268.

[7] D.T. Chen, H. Bourlard, J.-P. Thiran, Text identification in complex background using SVM, International Conference on Computer Vision and Pattern Recognition 2001; 621–626.

[8] Q. Ye, W. Gao, W. Wang, W. Zeng, A robust text detection algorithm in images and video frames, Joint Conference of Fourth International Conference on Information Communications and Signal Processing and Pacific-Rim Conference on Multimedia, Singapore 2003;.

[9] A.K. Jain, B. Yu, Automatic text location in images and video frames, Pattern Recognition 31 (1998) 2055–2076.

[10] Y. Zhong, K. Karu, A.K. Jain, Locating text in complex color images, Pattern Recognition 28 (1995) 1523–1535.

[11] H. Li, D. Doermann, O. Kia, Automatic text detection and tracking in digital video, Maryland University LAMP Technical Report 028, 1998.

[12] H. Li, D. Doermann, O. Kia, Automatic text detection and tracking in digital video, IEEE Transactions on Image Processing 9 (2000) 147–156.

[13] K.I. Kim, K. Jung, H. Kim, Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm, IEEE Transactions on PAMI 25 (2003) 1631–1639.

[14] X. Tang, X.B. Gao, J. Liu, H. Zhang, Spatial-temporal approach for video caption detection and recognition, IEEE Transactions on Neural Networks 13 (2002) 961–971.

[15] B. Luo, X. Tang, J. Liu, H. Zhang, Video caption detection and extraction using temporal feature vector, International Conference on Image Processing, Spain, September 2003; 297–300.

[16] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Transactions on PAMI 11 (1989) 674–693.

[17] I. Daubechies, Orthonormal bases of compactly supported wavelets, Communications on Pure and Applied Mathematics 41 (1988) 909–996.

[18] G.V. Wouwer, Wavelets for multiscale texture analysis, PhD Thesis, University of Antwerpen Belgium, 1998, pp. 43–56.

[19] B. Furht, Video and image Processing in Multimedia Systems, Kluwer Academic Publishers, 1995, pp. 226–270.

[20] A.K. Jain, Statistical pattern recognition: a review, IEEE Transactions on PAMI 2 (2001) 4–37.

[21] V. Vapnik, The Nature of Statistical Learning. Theory, Springer, New York, 1995.

[22] K. Sung, T. Poggio, Example-based learning for view-based human face detection, Massachusetts Institute of Technology, Cambridge, MA, A.I. Memo 1521, 1994.

[23] B. Heisele, T. Serre, S. Mukherjee, T. Paggio, Feature reduction and hierarchy of classifiers for fast object detection in video images, International Conference on Computer Vision and Pattern Recognition 2001; 18–24.

[24] T.F. Wu, C.J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research 2001; 975–1005.

[25] X.S. Hua, W.Y. Liu, H.J. Zhang, An automatic performance evaluation protocol for video text detection algorithms, IEEE Transactions on Circuits and Systems for Video Technology 14 (2004) 498–507.

[26] H. Li, D. Doermann, Text enhancement in digital video using multiple frame integration, ACM Multimedia 1999; 385–395.

[27] H. Li, O. Kia, D. Doermann, Text enhancement in digital videos, Proceedings of SPIE99-Document Recognition and Retrieval 1999.

[28] X.S. Hua, P. Yin, H. Zhang, Efficient video text recognition using multiple frame integration, International Conference on Image Processing, New York, September 2002; 22–25.

[29] D. Chen, J.-M. Odobez, H. Bourlard, Text segmentation and recognition in complex background based on Markov random field, Proceedings of the International Conference on Pattern Recognition, April 2002; 227–230.

[30] T. Sato, T. Kanade, E.K. Jughes, M.A. Smith, S. Satoh, Video OCR: indexing digital news libraries by recognition of superimposed captions, ACM Multimedia Systems: Special Issue on Video Libraries 7 (1999) 385–395.

[31] Q.X. Ye, W. Gao, Q.M. Huang, Automatic text segmentation from complex background, IEEE International Conference on Image Processing, Singapore, October 2004; 2305–2308.

[32] D.Q. Zhang, S.F. Chang, A Bayesian framework for fusing multiple word knowledge models in videotext recognition, International Conference on Computer Vision and Pattern Recognition 2003.