Image and Vision Computing 28 (2010) 1590–1599

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Text detection in images using sparse representation with discriminative dictionaries

Ming Zhao [a], Shutao Li [a,*], James Kwok [b]

[a] College of Electrical and Information Engineering, Hunan University, Changsha, 410082, China
[b] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China

ARTICLE INFO

Article history:
Received 23 August 2009
Received in revised form 31 March 2010
Accepted 6 April 2010

Keywords:
Text detection
Sparse representation
Discriminative dictionary

ABSTRACT

Text detection is important in the retrieval of texts from digital pictures, video databases and webpages. However, it can be very challenging since the text is often embedded in a complex background. In this paper, we propose a classification-based algorithm for text detection using a sparse representation with discriminative dictionaries. First, the edges are detected by the wavelet transform and scanned into patches by a sliding window. Then, candidate text areas are obtained by applying a simple classification procedure using two learned discriminative dictionaries. Finally, the adaptive run-length smoothing algorithm and projection profile analysis are used to further refine the candidate text areas. The proposed method is evaluated on the Microsoft common test set, the ICDAR 2003 text locating set, and an image set collected from the web. Extensive experiments show that the proposed method can effectively detect texts of various sizes, fonts and colors from images and videos.

1. Introduction

With the rapid development of digital devices, images and videos are now popular media in our daily lives. Texts, which are often embedded in images and videos, contain lots of semantic information useful for video comprehension. They can thus play an important role in content-based multimedia indexing and retrieval. In recent years, the automatic detection of texts from images and videos has gained increasing attention. However, the large variations in text fonts, colors, styles, and sizes, as well as the low contrast between the text and the often complicated background, often make text detection extremely challenging.

A lot of efforts have been put on addressing these problems [1–11], and these can be roughly divided into four categories. The first category uses connected component analysis (CCA) [1], in which pixels with similar colors are grouped into connected components, and then into text regions. CCA is fast. However, it fails when the texts are not homogeneous and text parts are not dominant in the image. The second category is based on edges [2], which assume high-contrast differences between the text and background. It is fast, and can have a high recall. However, it often produces many false alarms since the background may also have strong edges similar to the text. The third category is based on textures [3], and assumes that texts have specific texture patterns. It is more time-consuming and can fail when the background is cluttered with text. The fourth category is based on frequencies [4], in which the text is extracted from the background in the frequency (e.g., wavelet) domain. This is also time-consuming, and the frequency representation may not be better than the spatial representation. Recently, there is a lot of interest on using pattern classification techniques (such as AdaBoost [7], support vector machines [5,8,10], belief propagation [11] and neural networks [9]) for text localization. With the help of elaborately designed features that incorporate various properties of the text (such as geometry, color, texture and frequency), these techniques are often successful in discriminating text from its background.

On the other hand, the use of a sparse representation has recently drawn much attention in diverse classification applications. These include face recognition [12], signal classification [13] and texture classification [14]. With the assumption that natural images admit a sparse decomposition in some redundant basis (or so-called dictionary), various representations, including the curvelets, wedgelets, bandlets and various variants of wavelets, have been proposed. Recent works showed that these non-parametric dictionaries can significantly outperform the standard, off-the-shelf dictionaries [15,16].

Recently, Pan et al. [17] proposed the use of a sparse representation for text detection. It extracts text-like edges from an image by using a dictionary obtained by K-SVD [20]. However, as the K-SVD dictionary is designed for coding and denoising, it can be confused by complex backgrounds with text-like areas. In this paper, we overcome this deficiency by using a discriminative dictionary. Specifically, two overcomplete dictionaries are trained. The first dictionary provides a sparse representation for the text and a non-sparse representation for the background, while the second dictionary does the opposite. Subsequently, the image can be separated into text and background regions by comparing the reconstruction errors from each of these dictionaries. Finally, text detection can be efficiently performed by

* Corresponding author. Tel.: +86 731 88828850; fax: +86 731 88822224.
E-mail addresses: mingzhao_frank@yahoo.cn (M. Zhao), shutao_li@yahoo.com.cn (S. Li), jamesk@cse.ust.hk (J. Kwok).

0262-8856/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2010.04.002

applying the adaptive run-length smoothing algorithm [24] and projection profile analysis [27]. Contrary to the existing approaches, the proposed method utilizes edge information and sparse representation to locate text candidates, which makes it provide high precision rate and recall rate for texts with various sizes, fonts, colors and language. The proposed method has a three-step refinement scheme, which works well for extracting texts from complex and textured backgrounds. However, because of using horizontal and vertical directions wavelet basis and projection analysis, our method may not perform well on skewed texts detection.

The rest of this paper is organized as follows. In Section 2, the sparse representation and discriminative dictionary are briefly reviewed. In Section 3, we introduce the training of the discriminative dictionaries for text detection. Details of the proposed text detection method are presented in Section 4, and the experimental results are shown in Section 5. Finally, the conclusion is discussed in Section 6.

2. Sparse representation and discriminative dictionary

Sparse representation models have recently been used in image understanding tasks such as texture segmentation and feature selection [18,19]. The sparse representation of a signal over an overcomplete dictionary is achieved by optimizing an objective function that includes two terms: one measures the signal reconstruction error and the other measures the signal sparsity. Suppose that the data x in R^n admits a sparse approximation over an overcomplete dictionary $\mathbf{D}$ (where $\mathbf{D}\in R^{n\times K}$ with $K\gg n$) with K atoms. Then x can be approximately represented as a linear combination of a few atoms from $\mathbf{D}$. There are a number of algorithms that can be used to learn $\mathbf{D}$. One of the most popular algorithms is K-SVD [20], in which an overcomplete dictionary is obtained by solving the following optimization problem:

$$\min_{\alpha,\mathbf{D}} \sum_{l=1}^{M} ||x_l-\mathbf{D}\alpha_l||_2^2 \quad s.t.||\alpha_l||0\le L, \tag{1}$$

where $\mathbf{x}_l$ is the signal, $\mathbf{D}$ in $R^{n\times K}$ is the dictionary to be learned (each of its atoms is a unit vector in the l^2 norm), $\boldsymbol{\alpha}_l$ in R^K is the sparse representation of the lth signal using the dictionary $\mathbf{D}$, and $||\bullet||_0$ is the l_0 norm (that counts the number of nonzero entries in the vector argument). Thus, Eq. (1) finds the optimal dictionary with the lowest reconstruction error, subject to the constraint that each signal must have fewer than L atoms in its decomposition. Computationally, an iterative procedure can be used. First, the dictionary $\mathbf{D}$ is initialized, and then the training procedure alternates between sparse coding and dictionary update.

While K-SVD can obtain a good reconstructive representation, it may not perform well in a classification setting. As a remedy, Maira et al. proposed an algorithm that learns multiple discriminative dictionaries [21]. Given N classes of signals $S_i, i=1,\ldots,N$, it attempts to learn N dictionaries $\mathbf{D}_i$, with one dictionary per class. This yields the following optimization problem:

$$\min_{\{\mathbf{D}_j\}_{j=1}^N} \sum_{i=1}^{N} \sum_{l\in S_i} c_i^{\lambda}\left(\left\{R^*\left(x_l,\mathbf{D}_j\right)\right\}_{j=1}^{N}\right) + \lambda\gamma R^*(x_l,\mathbf{D}_i), \tag{2}$$

$$\text{where } R^*(\mathrm{x}_l,\mathbf{D})\equiv \min_{\alpha_l} ||\mathrm{x}_l-\mathbf{D}\alpha_l||_2^2 \; s.t.\phi(\alpha_l)\le 0, \tag{3}$$

where, $R^*(\mathbf{x}_l, \mathbf{D}_i)$ is the reconstruction error of the signal $\mathbf{x}_l$ using dictionary $\mathbf{D}_i$, γ controls the tradeoff between reconstruction and discrimination. Moreover, C_i^{λ} is a *softmax* cost function (with parameter λ), which is the multi-class version of the *logistic regression* function, which serves to make dictionary $\mathbf{D}_i$ a better representation than dictionaries $\mathbf{D}_j$ $(i\neq j)$ for signals from class S_i.

Computationally, the optimization procedure is implemented as a sequence of truncated Newton iterations with respect to the dictionaries. It is shown in [21] that each such iteration of updating dictionary $\mathbf{D}_j$ is equivalent to solving a problem of the form:

$$\min_{\substack{D'\in\mathbb{R}^{n\times k}\\ i\neq j}} \sum_{i=1}^{N} \sum_{l\in S_i} w_l||\mathrm{x}_l-D'\alpha_{lj}||_2^2, \tag{4}$$

where α_{lj} 's are the coefficients of the decompositions of $\mathbf{x}_l$ using dictionary $\mathbf{D}_j$, and w_l are the weights coming from a local linear approximation of C_i^{λ}. They have to be recomputed at each step of the optimization procedure. Empirically, the resultant sparse discriminative representation performs better in classification tasks.

3. Discriminative dictionaries for text detection

In this paper, two overcomplete dictionaries are trained. The first dictionary provides a sparse representation for the text, while the second one provides a sparse representation for the background. To train the text dictionary, we choose as training samples isolated machine-printed characters extracted from 5 synthesized document images. These images contain 1500 commonly used Chinese characters, 26 English letters and 10 Arabic numbers of various fonts and sizes. Two of the document images are shown in Fig. 1(a). Moreover, since we will mainly consider the detection of English and Chinese texts in the experiments, so only English and Chinese characters are included in the training set. As will be seen in the experiments, this still allows the detection of texts with similar shapes as Chinese or English (such as French). Obviously, this can also be extended to the detection of texts in other languages by simply including characters of those languages into the training set. As for the background dictionary, 56 non-text real scene images shown in Fig. 1(b) are used to construct the training set. These images are collected from the websites which include natural landscape, buildings, human beings, animals and vehicles.

Here, we will not utilize the color information of the images. Hence, all the training images obtained above are first converted to grayscale. Afterwards, edges are extracted by the Canny edge detector [22]. Finally, a small sliding window scans the image into patches with a raster-scan order, and all the non-edge patches are discarded. Note that the size of the sliding window is important. If it is too large, the resultant vectors will be high-dimensional, increasing the difficulty and time consumption of the classification process. If it is too small, the edge segments do not contain enough character or background information for discrimination. In the experiments, we found that 16×16 is a good tradeoff. Finally, a total of 200,000 text patches and 200,000 background patches are generated.

We then use the discriminative dictionary training algorithm in [21] to construct two dictionaries, D1 for the text and D2 for the background. In the experiment, each dictionary has 512 atoms. Moreover, 10 iterations of the algorithm are run. At each iteration, we prune the two sets by keeping the "best 90% classified patches" [21]. It is hoped that the overlap between the text and background sets can then be minimized. Finally, 41,178 patches remain in each set after training. The resultant text and background dictionaries are shown in Fig. 2.

4. Text detection via discriminative dictionaries

A flowchart of the proposed text detection algorithm is shown in Fig. 3. There are three key steps, each of which will be described in details below.

4.1. Edge detection with wavelets

Edges in the image are first extracted by the wavelet transform. In general, edges are created by objects which have different local

Fig. 1. Discriminative dictionary training samples: (a) text training samples; (b) background training samples.

intensity profiles due to occlusions, shadows, and textures. To label an edge precisely, it is thus necessary to analyze its local properties, which can be characterized mathematically by singularities in terms of the Lipschitz exponents. The wavelet theory shows that the Lipschitz exponents can be computed from the evolution across scales of the wavelet transform local extrema, and the local extrema of the wavelet transform correspond to the zero crossings of its derivative. Mallat and Zhong [23] pointed out that finding the local maxima of

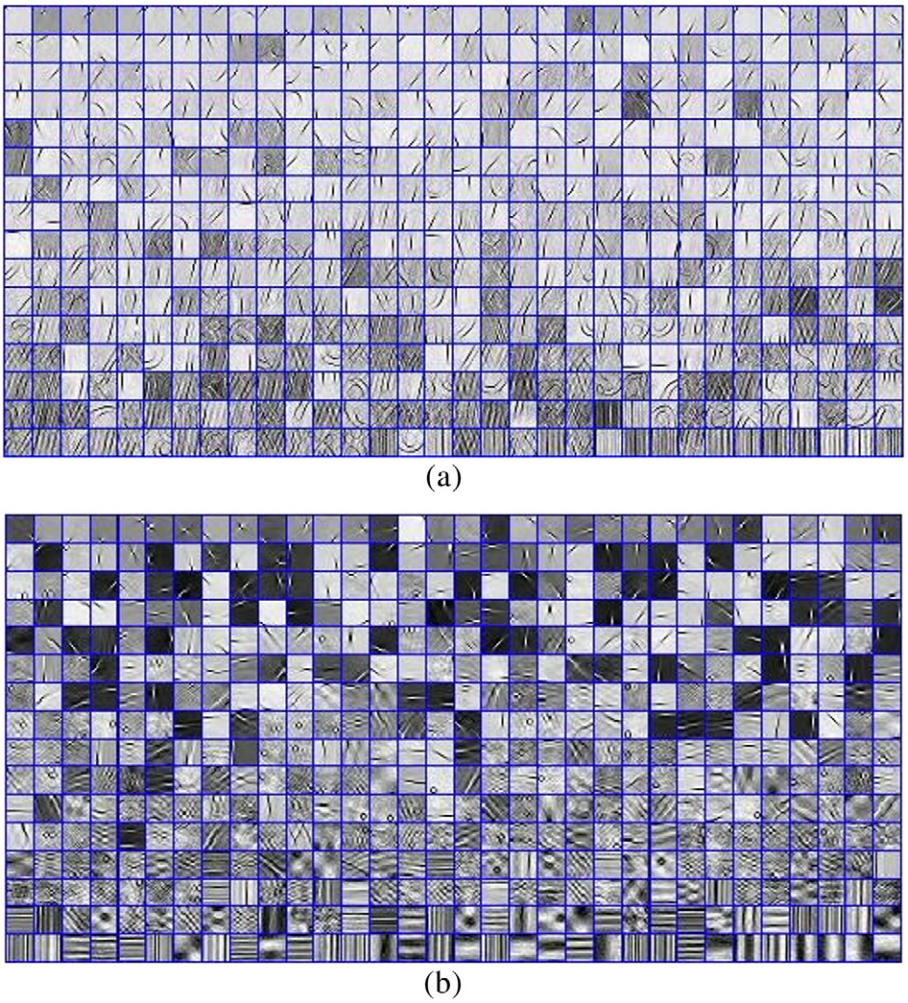

Fig. 2. Text and background dictionaries learned. (a): text dictionary; (b) background dictionary.

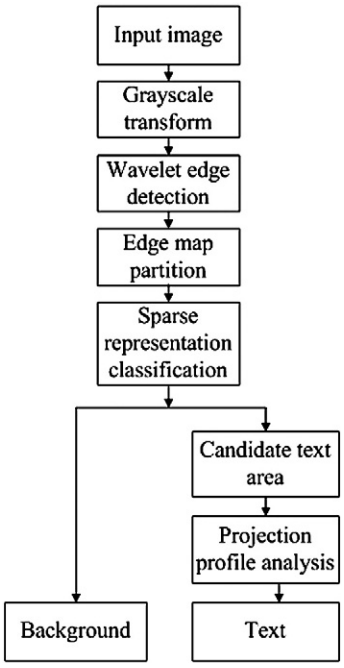

Fig. 3. Flowchart of the proposed text detection method.

a wavelet transform is equivalent to the multi-scale Canny edge detection [22].

As an example, we visualize the edges of the simpler one-dimensional signal in Fig. 4(a). Fig. 4(b) shows its discrete dyadic wavelet transform computed on three scales. Fig. 4(c) shows the corresponding derivatives and locations of the local maxima. As can be seen, each Dirac impulse in Fig. 4(c) indicates the position and amplitude of a local maximum in Fig. 4(b), which in turn indicates the edge position in Fig. 4.

Considering that most of the texts are in the horizontal or vertical directions, the orthogonal wavelet basis will be used in the following. However, this may not perform well on skewed texts. If it is necessary to detect skewed texts, we should introduce skew text line detection firstly. There are many skewed text line detection methods up to now [32,33]. We can detect the skewed angle by utilizing the skewed text line detection method, and then convert the skewed text to horizontal text. Afterwards, we can implement the proposed text detection method. The wavelet basis functions for the horizontal and vertical directions are:

$$\psi_H(x,y) = -xe^{-\frac{x^2+y^2}{2}}$$
$$\psi_V(x,y) = -ye^{-\frac{x^2+y^2}{2}} \quad (5)$$

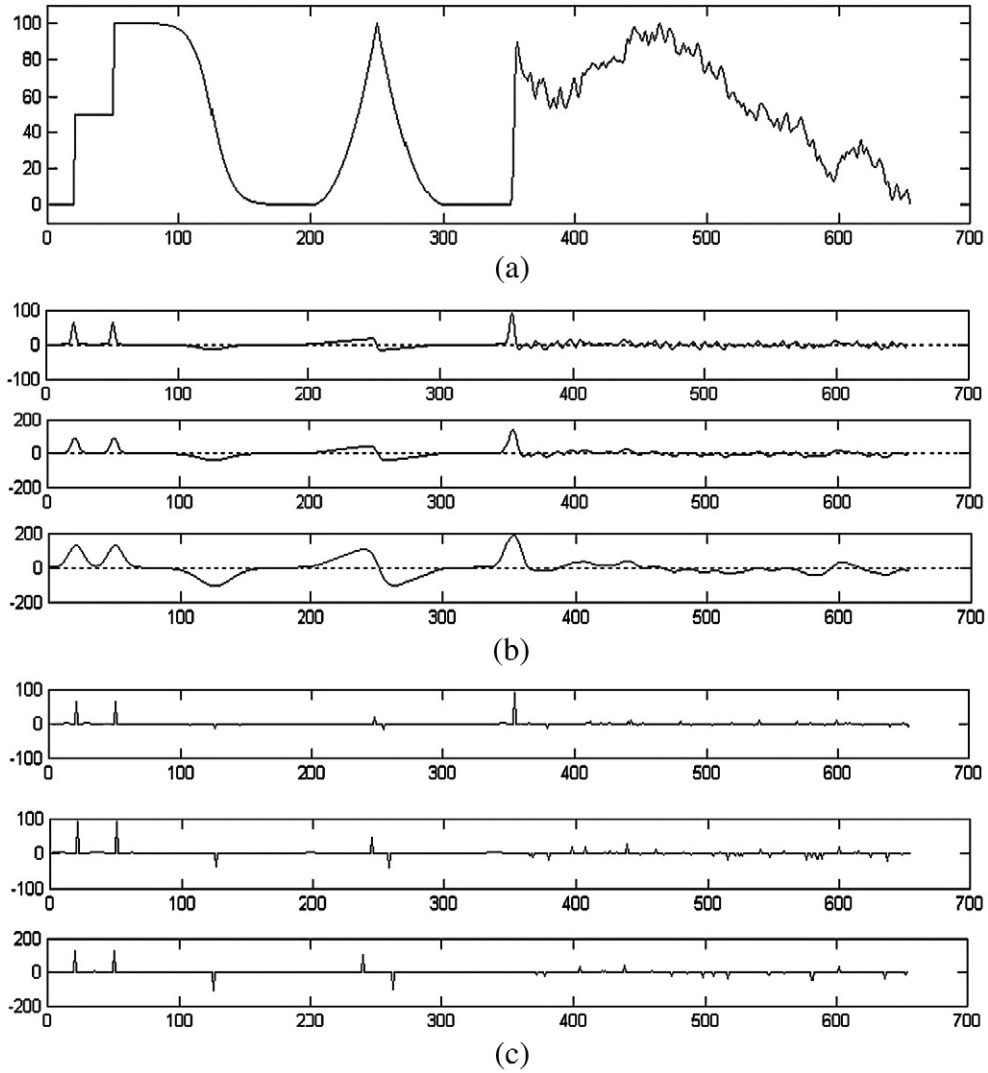

Fig. 4. Wavelet transform edge detection: (a) original signal; (b) three scales of the wavelet transform of (a); (c) derivatives of (b).

Therefore, the wavelet transforms of a signal at the original basis and their derivatives are:

$$W_{2^j}^H f(x,y) = f * \psi_{2^j}^H(x,y),$$
$$W_{2^j}^V f(x,y) = f * \psi_{2^j}^V(x,y), \quad (6)$$

Here, f represents the input image, '*' indicates the convolution, and j denotes the scale of the wavelet transform. Note that many extant text detection methods use the Canny edge detection operator. However, we use wavelets instead because the wavelet transform is multi-scale. In particular, the scale j above controls the threshold for which edges are to be detected. When the scale is large, the wavelet transform tends to remove small signal fluctuations such as complex background edges. However, unobvious text edges may also disappear when the scale is too large. In the experiments, we use $j=3$. A comparison of the performance of wavelet edge detection and classical edge detection methods is shown in Fig. 5. As can be seen, wavelet edge detection gives more precise text edges. Moreover, edges of the background can be more easily discriminated from the text edges. More edge detection results on images and video frames are shown in Fig. 6.

4.2. Edge classification using a sparse representation

In this step, we find all the possible text edges by performing classification with the use of a sparse representation. As in dictionary learning, a window of the same size is slided over the transformed edge image and the edge image is segmented into many non-overlapping patches (which are represented as column vectors in Fig. 7(b)). Each patch is then classified by comparing the two reconstruction errors R_i^*, $i=1, 2$, resulting from the text and background dictionaries:

$$R_i^*(\mathbf{x},D_i) \equiv ||\mathbf{x}-\mathbf{D}_i\alpha^*(\mathbf{x},D_i)||_2^2, \quad (7)$$

$$\text{where}\,\alpha^*(\mathbf{x},\mathbf{D}) = \min_{\alpha\in\mathbb{R}^k} ||\mathbf{x}-\mathbf{D}\alpha||_2^2, s.t. ||\alpha||0 \le L \quad (8)$$

Here $\mathbf{D}_i$, $i=1, 2$, indicates the trained text dictionary and the background dictionary, respectively. If R_1^* is smaller than R_2^*, the text dictionary is better than the background dictionary in representing the patch. Therefore, the patch should be classified as text. Conversely, if $R_1^*$is larger than R_2^*, the patch should belong to the background. Then, the background patches are set to zero, and text patches are reserved. Afterwards, we reconstruct the edge image from those patches for the following refinement process.

4.3. Text area refinement

Sometimes, image blocks with several closely parallel edges may be falsely detected as text. This can be especially problematic for textures such as square objects, leaves, etc. In the following, we will use the horizontal and vertical projection profile analysis with adaptive run-length smoothing algorithm (ARSLA) [24] to separate the true texts from the candidate ones.

The classical run-length smoothing algorithm (RLSA) [30] is applicable to binary images, which takes advantage of the white runs existing in the horizontal and vertical directions. It is a low-complexity technique and can segment candidate text edges into rectangular blocks and then classify them into either text or background. In each direction, RLSA eliminates white runs whose lengths are smaller than a threshold smoothing value. Recall that the input patches are of size 16×16 pixels, the smoothing values in both the horizontal and vertical directions are thus also set to 16 pixels. However, RLSA may sometimes group inhomogeneous connected components or different slanted lines together. The ARLSA is an extension which overcomes these drawbacks. Empirically, ARLSA can also handle images containing characters with variable font sizes.

In our algorithm, ARLSA is first performed in the horizontal direction (Fig. 8(b)). Afterwards, a morphological open operation is used to remove isolated edges (Fig. 8(c)). The structuring element is a 3×3 matrix of all '1's. Then, the projection profile analysis is applied in the horizontal direction. The text line is extracted from the horizontal run histogram as a couple of parallel rectilinear lines. From the input image, the horizontal projections are computed. The number of runs in the ith pixel row is stored. Let H be the horizontal projection. The text line can be associated with the most significant part of a peak in H. The maximum (minimum) of the first derivative of the projection on the left (right) of the peak approximately represents the upper-line (baseline) of the text line. The longer and more rectilinear the text line is, the better the body evaluation is. The valleys of the histogram represent the separation between two consecutive text lines. The horizontal projection profile analysis extracts text lines from the candidate text area. Next, we perform the vertical ARLSA and vertical projection profile analysis in those text lines which have been extracted from the horizontal projection analysis. Sample results from the vertical ARLSA and projection analysis are shown in Fig. 9 (a)–(e).

After performing these in the horizontal and vertical directions, isolated edges are sorted out as false text alarms and discarded. A sample result of this refinement process is shown in Fig. 9(f). Finally, the left-most, right-most, top and bottom coordinates are used to create a bounding rectangle.

5. Experimental results

In this section, experiments are performed on three test sets. The first one is the Microsoft common test set for video text detection [25] with 45 video frames. The second one is the ICDAR 2003 text locating test set [31], which includes 258 real scene pictures. The last one is collected from the web, and includes 102 images/video frames collected from movies, news clips, sports videos and music videos. These cover a variety of texts in different fonts, sizes, styles and languages, light texts on complex backgrounds, texts of poor qualities, etc. Moreover, for fair comparison, these images are not used in the training of the discriminative dictionaries.

Fig. 5. Edge detection results for the letter "R" in a complex background: (a) original image; (b) edges obtained by the Canny detector; (c) edges obtained by the Sobel detector; (d) edges obtained by the wavelet detector.

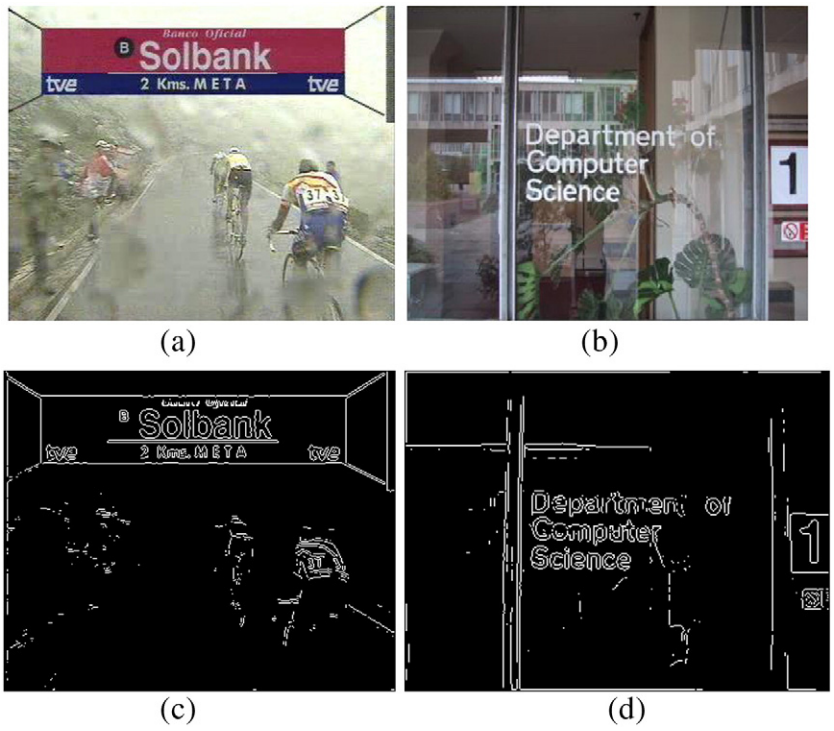

Fig. 6. Wavelet edge detection for real scene text images: (a) and (b): original images; (c) and (d): the corresponding edge maps obtained by the wavelet detector.

Two metrics that have been commonly employed in information retrieval, namely,

$$\text{Precision} = \frac{\text{Number of correctly detected text}}{\text{Number of detected text}} \tag{9}$$

$$\text{Recall} = \frac{\text{Number of correctly detected text}}{\text{Number of text}} \tag{10}$$

will be used for performance evaluation. Here, the text is considered as having been detected correctly if the overlapped area between the detection result and ground truth is larger than 90% of the ground truth. The algorithm is implemented in Matlab 7.0, and experiments are performed on a PC with an AMD Sempron 1.83 GHz CPU with 1 GB RAM running Windows XP.

5.1. Effect of the sparsity factor

The proposed method has three main parameters: size of the patch (n), size of the dictionaries (K) and the sparsity factor (L). For a reasonable tradeoff between complexity and performance, we set

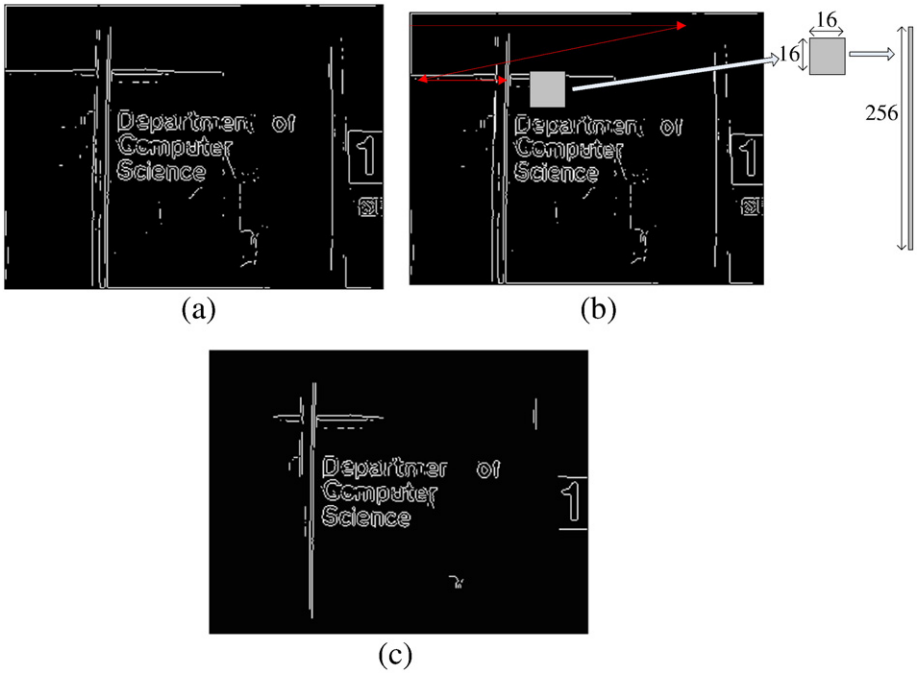

Fig. 7. Candidate text edges extracted by the discriminative dictionaries: (a) edge map of the input image; (b) edge map partition and vectorization; (c) candidate text region.

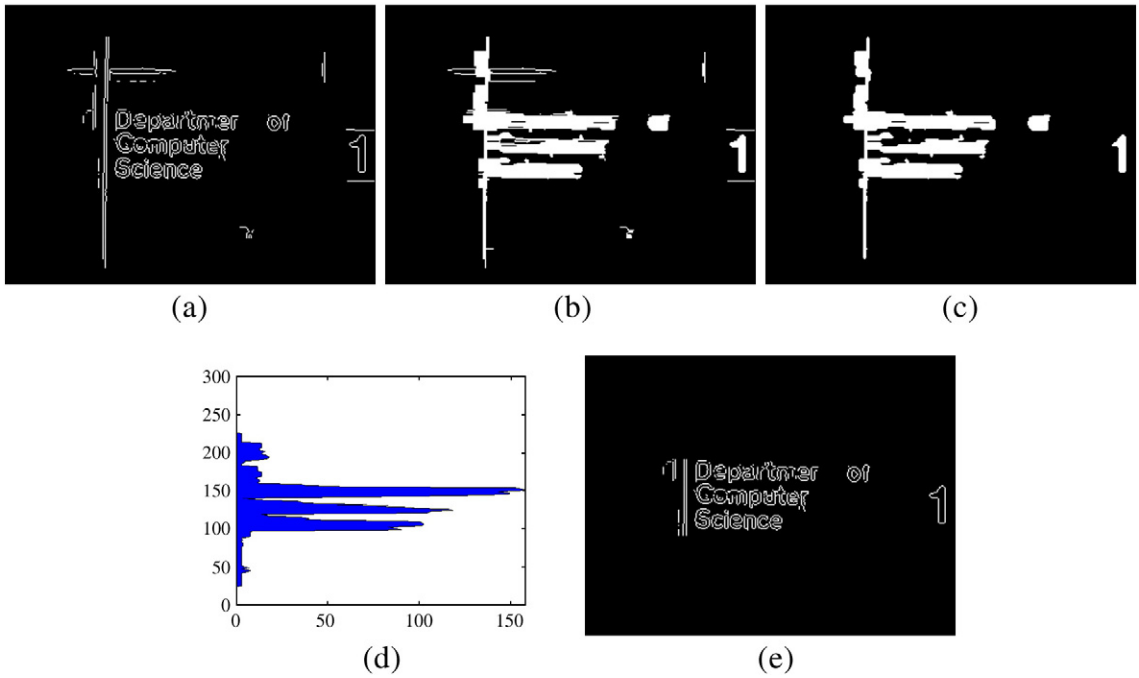

Fig. 8. Adaptive run-length smoothing algorithm and projection analysis in the horizontal direction: (a) candidate text region; (b) horizontal adaptive run-length smoothing algorithm in the horizontal direction; (c) morphological open operation; (d) projection profile analysis in the horizontal direction; (e) extracted text lines.

$n=256$ and $K=512$. Here, we first study the effect of the sparsity factor. Experiments are performed on the Microsoft common test set, with L {6, 7, 8, 9, 10}. Fig. 10 shows the effect on the precision and recall. As can be seen, the best choice is $L=8$, which leads to a precision of 98.8% and a recall of 94.2%. Therefore, in the following experiments, the sparsity factor (L) is always set to 8.

5.2. Effect of the font size of the text

As mentioned in the previous section, the size of the sliding window is set to 16×16 so as to have a reasonable tradeoff between training time and performance. Because of this fixed size, the sliding window may not be able to capture enough information when the

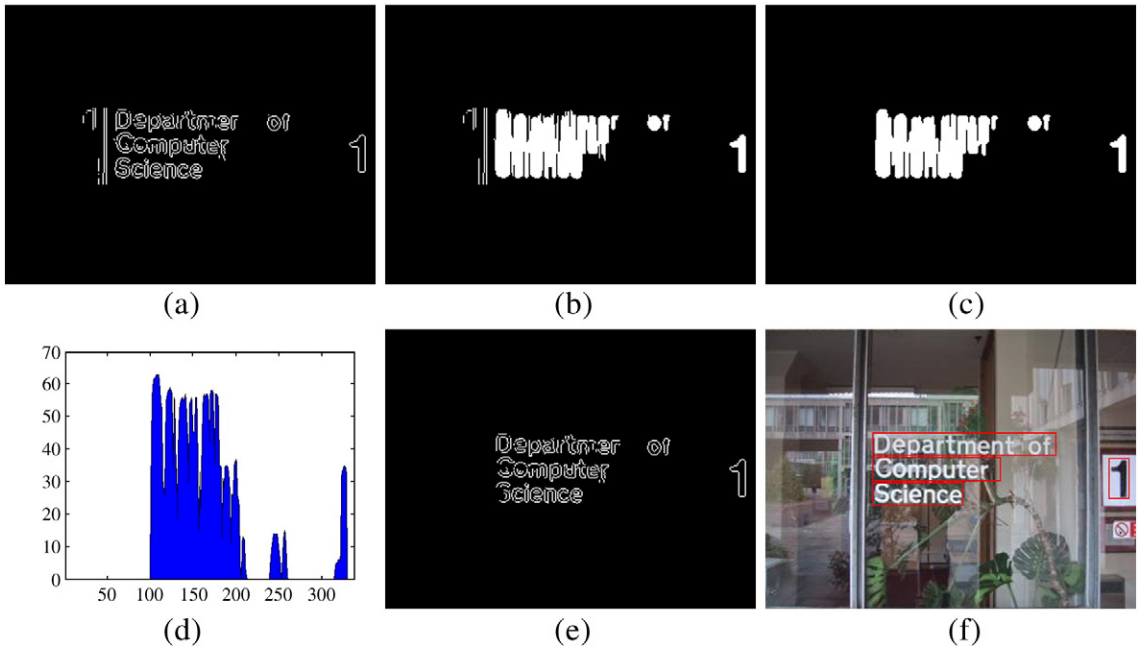

Fig. 9. Adaptive run-length smoothing algorithm and projection analysis in the vertical direction: (a) extracted text lines; (b) horizontal adaptive run-length smoothing algorithm in the horizontal direction; (c) morphological open operation; (d) projection profile analysis in the vertical direction; (f) result of vertical projection analysis; (e) final text detection result.

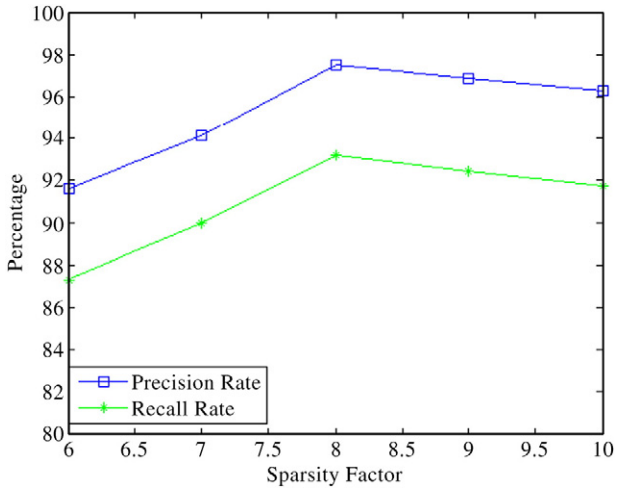

Fig. 10. Effect of different sparsity factors on precision and recall.

text's font size is too large, leading to a subsequent degradation in performance. This is demonstrated in the following experiment. First, we choose ten images that do not contain any text. Texts of twelve different sizes, from 8-point to 164-point per character, are then added to these non-text images, leading to a total of $10\times12=120$ images. The resultant detection results are shown in Fig. 11. As can be seen, both precision and recall fall sharply when the text size is roughly larger than 130 points. Hence, in order to handle large texts (with a font size larger than 130 points), a downsampling operation can be used to first reduce the size of text into one-quarter of its original size. Afterwards the proposed method can be implemented in the different size images.

5.3. Comparison with various text detection methods

In this section, the following methods will be compared:

1) The proposed method, which will be denoted as "wavelet + discriminative dictionary";
2) A variant of the proposed method, which uses the Canny edge detector instead of the wavelet detector ("Canny + discriminative dictionary");
3) Pan et al.'s method [17] ("Canny + K-SVD");
4) The use of a support vector machine (SVM) classifier operating on the wavelet features directly ("wavelet + SVM"). In this experiment, the RBF Kernel function with gamma of 0.5 is used for the SVM. The training set used by the SVM is as the same as the discriminative dictionary.

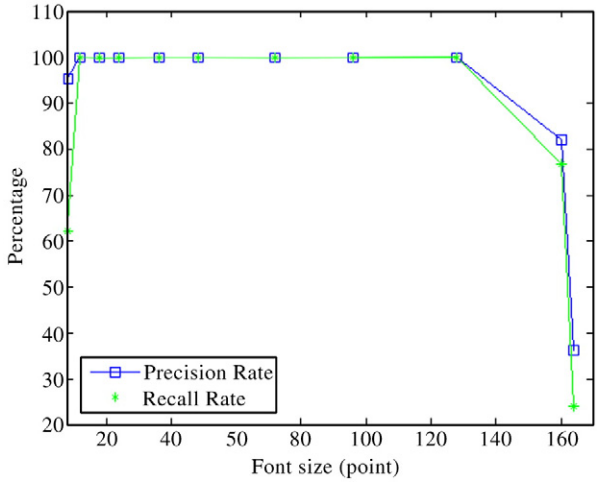

Fig. 11. Effect of the text's font size on precision and recall.

Experiments are performed on a total of 405 images from all the three databases.

Table 1 shows the results. As can be seen, the precision and recall of the wavelet-based method are 2.3% and 2.2% higher than that of the Canny-based method. Hence, wavelet edge detection is more accurate than the Canny filter when the background is complex. The disadvantage of the wavelet-based method is that it is more computationally expensive during text candidate detection. However, only a small additional computation cost is needed in the subsequent processes. Also, we can see from Table 1 that the recall and precision of the two discriminative dictionary-based methods are higher than those of Pan et al.'s method [17]. It thus verifies our original hypothesis that using multiple discriminative dictionaries is better than using a single reconstructive dictionary in text candidate detection. Moreover, note that the proposed method is also better than that of using a support vector machine (SVM) classifier directly on the wavelet features. This is probably because the edge map extracted by the wavelet transform is sparse and thus a sparse representation is useful. Table 1 also reports the average processing time for a 640×480 image. As can be seen, the speed of the proposed method is comparable with those of the others.

Fig. 12 shows some representative results of the proposed text detection method. As can be seen, these include a wide range of texts. In particular,

- Fig. 12(a), (d), (e), (f), (i) and (j) contain texts in real scene pictures, while Fig. 12(b), (c), (g) and (l) contain texts that are added on the image or video frames;
- Fig. 12(b), (c) and (l) contain non-English/Chinese texts of different sizes;
- Fig. 12(d), (e) and (j) contain texts embedded in complicated backgrounds;
- Fig. 12(f) contains texts with two different languages;
- Fig. 12(f) and (k) contain texts with different colors;
- Fig. 12(g) contains texts of different fonts and sizes;
- Fig. 12(j) contains light text on a complicated background;
- Fig. 12(h) does not contain any text, and it demonstrates that the proposed method can successfully avoid false alarms in a non-text image.

Moreover, a number of methods have reported results on the Microsoft common test set. These include

(1) Ye et al. [8]: a feature selection algorithm which uses a SVM classifier to verify multi-scale wavelet features in the text/non-text classification tasks;
(2) Mancas-Thillou et al. [26]: it uses two metrics to merge similar colors together for text-driven segmentation in the RGB color space;
(3) Lienhart et al. [27]: it detects texts by using a complex-valued multilayer feed-forward network trained to detect text at a fixed scale and position;

Table 1
Performance of various text detection approaches.

Method	Precision	Recall	Average processing time for a 640×480 image (in sec)
Wavelet + discriminative dictionary	78.73%	76.33%	25.9
Canny + discriminative dictionary	76.43%	74.10%	25.9
Canny + K-SVD [17]	70.62%	72.32%	20.8
Wavelet + SVM	76.21%	74.63%	19.7

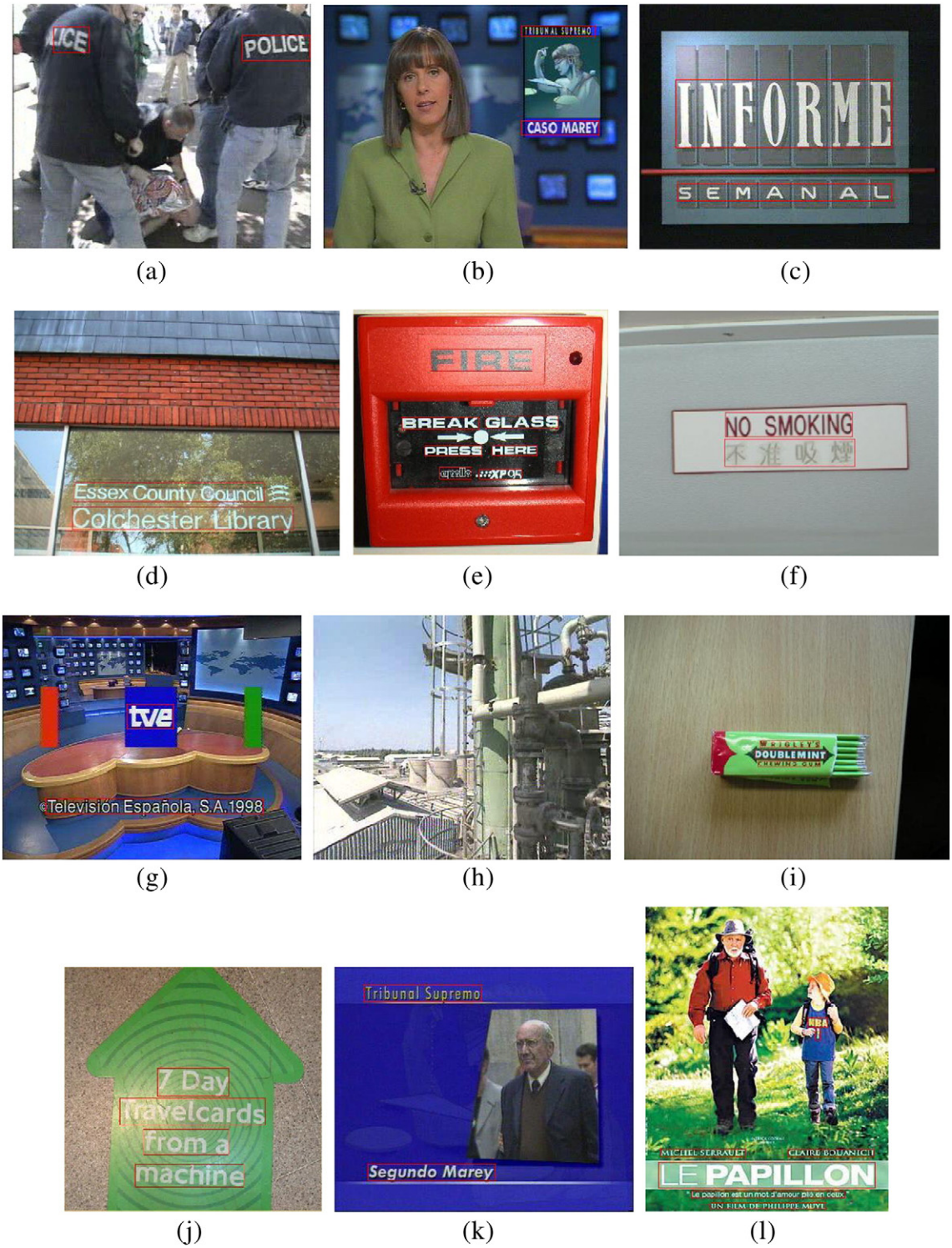

Fig. 12. Sample text detection results using the proposed method.

(4) Shivakumara et al.'s method [27]: it is based on candidate text block selection and segmentation of text portion of the image;
(5) Gllavata et al.'s method [29]: it is based on wavelet coefficients classification.

Hence, in Table 2, we also compare the performance of the proposed method with the above methods on this data set. As can be seen, the proposed method and Ye et al.'s method exhibit similar recall, and are far better than the other three methods, while the proposed method is more accurate than Ye et al.'s method. Mancas-Thillou et al.'s method and Lienhart et al.'s method exhibit a medium recall and precision. Shivakumara et al.'s method uses an improved edge feature extracting algorithm for text detection, which exhibits a medium recall and a high false rate. Gllavata et al.'s method shows the lowest precision as well as the lowest recall. This is mainly because of the noisy and complex background that often occurs in images. As can be seen, the proposed method can locate the text area much more precisely.

Table 2
Comparison of the various methods on the Microsoft common test set.

Method	Recall	Precision
Ye et al. [8]	94.2%	97.6%
Mancas-Thillou et al. [26]	91.0%	93.6%
Lienhart etl al. [27]	91.4%	94.4%
Shivakumara etl al. [28]	92%	90.4%
Gllavata etl al. [29]	90%	87%
Our method	94.2%	98.8%

6. Conclusion

In this paper, we proposed a sparse representation classification method based on discriminative dictionaries. We first extracted the edges of an input image by the wavelet transform. They are then classified by using sparse representations with discriminative dictionaries. Finally, the text regions are located by performing ARLSA and projection profile analysis on the edge classification results. Experimental results show that the proposed text detection method outperforms the other techniques. In particular, it allows robust text detection, without having to place assumptions on the text size, color or other textural properties. Moreover, it can detect overlay texts in images and video frames, as well as texts in the scene images.

Acknowledgements

The authors would like to thank the editor and anonymous reviewers for their detailed review, valuable comments, and constructive suggestions. This paper is supported by the National Natural Science Foundation of China (No. 60871096 and 60835004), the Ph.D. Programs Foundation of the Ministry of Education of China (No. 200805320006), the Key Project of the Chinese Ministry of Education (2009-120), and the Open Projects Program of the National Laboratory of Pattern Recognition, China.

References

[1] J. Lim, J. Park, G.G. Medioni, Text segmentation in color images using tensor voting, Image and Vision Computing 25 (5) (2007) 671–685.
[2] M. Lyu, J. Song, M. Cai, A comprehensive method for multilingual video text detection, localization, and extraction, IEEE Transactions on Circuits and Systems for Video Technology 15 (2) (2005) 243–255.
[3] K.I. Kim, K. Jung, J.H. Kim, Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (12) (2003) 1631–1639.
[4] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, S.D. Joshi, Text extraction and document image segmentation using matched wavelets and MFR model, IEEE Transactions on Image Processing 16 (8) (2007) 2117–2128.
[5] C. Jung, Q. Liu, J. Kim, Accurate text localization in images based on SVM output scores, Image and Vision Computing 27 (2009) 1295–1301.
[6] X. Liu, H. Fu, Y. Jia, Gaussian mixture modeling and learning of neighboring characters for multilingual text extraction in images, Pattern Recognition 41 (2008) 484–493.
[7] D. Chen, O. Jean-Marc, B. Herve, Text detection and recognition in images and video frames, Pattern Recognition 37 (3) (2004) 595–608.
[8] Q.X. Ye, Q.M. Huang, W. Gao, D.B. Zhao, Fast and robust text detection in images and video frames, Image and Vision Computing 23 (6) (2005) 565–576.
[9] C. Strouthopoulos, N. Papamarkos, Text identification for document image analysis using a neural network, Image and Vision Computing 16 (12–13) (1998) 879–896.
[10] M. Anthimopoulos, B. Gatos and I. Pratikakis, A two-stage scheme for text detection in video images, Image and Vision Computing, (2010) to appear.
[11] H.Y. Shen, J. Coughlan, V. Ivanchenko, Figure-ground segmentation using factor graphs, Image and Vision Computing 27 (7) (2009) 854–863.
[12] J. Wright, A. Ganesh, A.Y. Yang, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.
[13] W.V. Aerschot, M. Jansen, A. Bultheel, Normal mesh based geometrical image compression, Image and Vision Computing 27 (4) (2009) 459–468.
[14] T.-W. Lee, M.S. Lewicki, Unsupervised image classification, segmentation, and enhancement using ICA mixture models, IEEE Transactions on Image Processing 11 (3) (2002) 270–279.
[15] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Transactions on Image Processing 15 (12) (2006) 3736–3745.
[16] S. Roth, M.J. Black, Fields of experts, a framework for learning image priors, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 860–867.
[17] W. Pan, T.D. Bui, C.Y. Suen, Text detection from scene images using sparse representation, Proceeding of the 19th International Conference on Pattern Recognition (ICPR' 08), 2008, pp. 1–5.
[18] F. Zhang, X.Q. Ye, W.Y. Liu, Image decomposition and texture segmentation via sparse representation, IEEE Journal Signal Processing Letters 15 (2008) 641–644.
[19] L. Shang, Non-negative sparse coding shrinkage for image denoising using normal inverse Gaussian density model, Image and Vision Computing 26 (8) (2008) 1137–1147.
[20] M. Aharon, M. Elad, A.M. Bruckstein, The KSVD: an algorithm for designing of overcomplete dictionaries for sparse representations, IEEE Transactions on Signal Processing 54 (11) (2006) 4311–4322.
[21] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08), 2008, pp. 1–8.
[22] J.F. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.
[23] S.G. Mallat, S. Zhong, Characterization of signals from multiscale edges, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (7) (1992) 710–732.
[24] N. Nikolaou, M. Makridis, B. Gatos, Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths, Image and Vision Computing 28 (4) (2010) 590–604.
[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 (4) (2004) 498–507.
[26] C. Mancas-Thillou, B. Gosselin, Color text extraction with selective metric-based clustering, Computer Vision and Image Understanding 107 (1–2) (2007) 97–107.
[27] R. Lienhart, A. Wernicke, Localizing and segmenting text in images and videos, IEEE Transactions on Circuits and Systems for Video Technology 12 (4) (2002) 256–268.
[28] P. Shivakumara, W.H. Huang, C.L. Tan, Efficient video text detection using edge features, Proceedings of the 19th International Conference on Pattern Recognition (ICPR' 08), 2008, pp. 1–4.
[29] J. Gllavata, R. Ewerth, B. Freisleben, Text detection in images based on unsupervised classification of high-frequency wavelet coefficients, Proceedings of the 17th International Conference on Pattern Recognition (ICPR' 04), 2004, pp. 425–428.
[30] K. Wong, R.G. Casey, M. Wahl, Document analysis system, IBM Journal of Research and Development 26 (6) (1982) 647–656.
[31] The ICDAR 2003 competitions. <http://algoval.essex.ac.uk/icdar/Competitions.html>.
[32] A. Chaudhuri, S. Chaudhuri, Robust detection of skew in document images, IEEE Transaction on Image processing 6 (2) (1997) 344–349.
[33] S.T. Li, Q.H. Shen, J. Sun, Skew detection using wavelet decomposition and projection profile analysis, Pattern Recognition Letters 28 (5) (2007) 555–562.