Multioriented Video Scene Text Detection Through Bayesian Classification and Boundary Growing

Palaiahnakote Shivakumara, Rushi Padhuman Sreedhar, Trung Quy Phan, Shijian Lu, and Chew Lim Tan, *Senior Member, IEEE*

***Abstract*—Multioriented text detection in video frames is not as easy as detection of captions or graphics or overlaid texts, which usually appears in the horizontal direction and has high contrast compared to its background. Multioriented text generally refers to scene text that makes text detection more challenging and interesting due to unfavorable characteristics of scene text. Therefore, conventional text detection methods may not give good results for multioriented scene text detection. Hence, in this paper, we present a new enhancement method that includes the product of Laplacian and Sobel operations to enhance text pixels in videos. To classify true text pixels, we propose a Bayesian classifier without assuming *a priori* probability about the input frame but estimating it based on three probable matrices. Three different ways of clustering are performed on the output of the enhancement method to obtain the three probable matrices. Text candidates are obtained by intersecting the output of the Bayesian classifier with the Canny edge map of the input frame. A boundary growing method is introduced to traverse the multioriented scene text lines using text candidates. The boundary growing method works based on the concept of nearest neighbors. The robustness of the method has been tested on a variety of datasets that include our own created data (nonhorizontal and horizontal text data) and two publicly available data, namely, video frames of Hua and complex scene text data of ICDAR 2003 competition (camera images). Experimental results show that the performance of the proposed method is encouraging compared with results of existing methods in terms of recall, precision, F-measures, and computational times.**

***Index Terms*—Bayesian classifier, boundary growing, Laplacian–Sobel product (LSP), maximum gradient difference, multioriented video scene text detection, text candidate detection.**

I. Introduction

In the field of information retrieval, text detection and extraction from video has become an emerging area to solve the fundamental problem of content-based image retrieval (CBIR) to fill in the semantic gap between low level and high level features. Text detection and extraction enables the understanding of video contents with the help of text recognition using optical character recognition techniques, to give a partial solution to bridge the semantic gap between the low level and high level features. Traditional CBIR for labeling large datasets generally requires either human intervention or tedious prior study and annotation of the image content. The performance of CBIR thus degrades with increasing database size [1]–[2]. Therefore, text detection and extraction plays an important role in the field of information retrieval. In addition, due to the decrease in prices of devices and the advancement of technology, the use of video and camera devices increases drastically. This leads to huge databases containing a great variety of videos. Hence, efficient and relevant video retrieval from such databases becomes a major research issue in the field of computer vision and pattern recognition.

Text detection and extraction is quite a familiar topic in document image analysis on camera-based images with a proliferation of research papers published in the past decade [2]–[9]. Pan *et al.* [8], [9] proposed a hybrid approach to detect and localize texts in natural scene images based on histogram of oriented gradients and conditional random fields. This method uses geometrical properties of connected component analysis to group the text components. Since this method is developed for scene texts in camera images, the performance of the method on video texts degrades due to low resolution and complex background of videos. The stroke width of the character component was used as the basis for text detection in natural scenes by Epshtein *et al.* [10]. However, features based on stroke widths may not be good for texts in video frames as video texts may contain several discontinuities due to low resolution. It is observed from recent works in [2]–[10] on scene text extraction from natural scenes and camera images that these methods assume big fonts such as in caption texts in video and high contrast texts with clear character shapes for text detection. These assumptions are valid for camera-based images, but not necessarily valid for video-based images due to unfavorable properties of videos such as variation in contrast, complex background, different fonts and sizes, different orientations, perspective deformation, color bleeding, text movements, background movements, and so on [1]–[5].

Multioriented text extraction from camera images can also be seen in [11], but this method works well if the text with a clear character shape is present in the images. Thus, document-analysis-based methods for text extraction from camera images and scene text extraction from natural scene photographs may

Manuscript received December 29, 2010; revised June 8, 2011 and September 30, 2011; accepted January 18, 2012. Date of publication May 7, 2012; date of current version July 31, 2012. This work was supported in part by A*STAR, under Grant 092 101 0051 (WBS R252-000-402-305). This paper was recommended by Associate Editor Q. Tian

P. Shivakumara, R. P. Sreedhar, T. Q. Phan, and C. L. Tan are with the School of Computing, National University of Singapore, Singapore 117417 (e-mail: shiva@comp.nus.edu.sg; rushi@comp.nus.edu.sg; phanquyt@comp.nus.edu.sg; tancl@comp.nus.edu.sg).

S. Lu is with the Institute for Infocomm Research, Agency for Science, Technology, and Research, Singapore 138632 (e-mail: slu@i2r.a-star.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2012.2198129

not be suitable for scene text detection or extraction from video images [3], [12], [13].

Text in digital videos can be divided into two classes, scene text and graphics text. Scene text appears within the scene and is captured by the camera. Examples of scene text include street signs, billboards, and text on trucks and writing on shirts. Graphics text, on the other hand, refers to text that is manually added to video frames to supplement the visual and audio content. Since it is purposefully added it is often more structured and closely related to the subject than scene text. In some domains such as sports, however, scene text can be used to uniquely identify objects. Although scene text is often difficult to detect and extract due to its virtually unlimited range of poses, sizes, shapes, and colors, it is important in applications such as navigation, surveillance, video classification, and analysis of sports events [1]–[5], [12], [13].

Several methods for text detection and extraction in video frames have been proposed in the literature. These methods can be broadly classified into three categories, namely, connected component-based, texture-based, and edge and gradient-based methods. Jain and Yu [14] proposed a classical text detection algorithm based on connected component analysis. In this method, connected components based on colors are selected, which will be regarded as text if they satisfy some geometrical features. Jung *et al.* [15] also used connected component analysis to locate text in complex color images. Although the method is a classical one, it fails when multiple color characters are present in a text line. In summary, connected-component-based methods are good for caption text with plain background images, but not for images with cluttered background and particularly for multioriented scene text.

To overcome the problem of complex background, texture-based methods consider the appearance of text as a special texture to discriminate text from nontext [4], [16]–[18]. Li *et al.* [17] used the mean, second, and third-order central moments in the wavelet domain as texture features and a neural network classifier as the classifier for text block detection. Zhong *et al.* [18] detected text in the JPEG/MPEG compressed domain using texture features from DCT coefficients. Its robustness in complex background may not be satisfactory due to the limitation of spatial domain features. Kim *et al.* [4] proposed a texture-based method using support vector machines (SVM). To classify text and nontext pixels, the method employs an adaptive mean shift algorithm along with SVM. Although the SVM-based learning approach makes the algorithm fully automatic, it is difficult to discriminate text from nontext using pure texture features in a complex background because the features are insufficient to discriminate text using general textures.

Edge and texture features without classifier were proposed by Liu and Dai in [19] for text detection, but the method uses a large number of features to discriminate text and nontext pixels. Edge-texture features with a classifier are also used for video text detection [20], [21]. These methods depend too much on the classifier used. Jung *et al.* [22] proposed a method based on a stroke filter and a classifier for text detection in video. The method is sensitive to complex background and small fonts. Wavelet transform and a set of texture features without a classifier were proposed by us in [23] for accurate text detection in videos. Although the method works well for a variety of frames, it requires more time to process due to the large number of features and wavelet transforms. Ye *et al.* [24] used a combination of edge features and texture features in the wavelet transform domain with a classifier to detect text in videos. To enhance the performance of text detection, we [25] proposed another method to classify low and high contrast images before employing any text detection methods. Recently, we have also explored a combination of Fourier and statistical features in color space to handle the complexity of video text detection [26]. Since it uses a large number of features in frequency and color domain, it is computationally expensive and is limited to horizontal text detection.

In the same way, an edge-feature-based method by Wu *et al.* [27] used nine second-order Gaussian derivatives to extract vertical strokes in horizontally aligned text regions. From the strokes the chip is constructed. The chip in which algorithms are embedded will be further checked for structural properties like values of height and width. Leinhart and Wernicke [28] located text in images and video frames using image gradient features and a neural network classifier. Chen *et al.* [29] used canny edge features and a morphological close operation to detect candidate text blocks. The combination of edge and gradient features was used for efficient text detection with low false positives by Wong and Chen in [30], where this method assumed that text is in the horizontal direction. The above methods often have a high recall rate, but produce many false positives since the background may also have strong edge (gradient) just as text does [31], [32]. Edge and color features were also used with the aim of low contrast text detection by Cai and Lyu in [33], where the mask operation was performed for controlling contrast by setting ad hoc thresholds. Color-feature-based clustering for detecting caption texts in videos was proposed by Mariano and Kasturi in [34]. This method works if text lines contain uniform color characters and words. The method proposed in [35] considered transient color as a basis for overlay text detection and extraction from complex video scenes. Generated transition maps using color transient may work well for overlay texts, but not for scene texts embedded in complex background.

Based on the above discussion, we can conclude that there are methods to detect graphics text efficiently, but little work was done on both graphic and scene text detection. Furthermore, a few methods addressed the issue of multioriented scene text detection, but the performance of these methods is low because scene text poses many challenges compared to graphics texts. Multioriented text has only been partially addressed in [3] and [36], where the algorithm is limited to caption text and a few selected directions. Recently, we have addressed this multioriented issue in [37] based on Laplacian and skeletonization methods. However, this method still has room for improvement in the following respects.

1) Recall, precision, and F-measure are low for multi-oriented text (nonhorizontal text) detection, because the segmentation method fails to split a single text component into text lines properly.

2) The method gives high false positive rate and misdetection rate (MDR) for the multioriented text because of heuristics involved in the segmentation method.
3) The method is computationally expensive since it involves connected component labeling to classify simple and complex components and Fourier transform to extract the text components. The present method aims at overcoming the above problems.

Hence, in this paper, we propose a new method for enhancing text information by taking advantage of Laplacian and Sobel operations on the input image. The new method makes use of a Bayesian classifier without assuming *a priori* probability of the original image to classify enhanced text pixels from nontext pixels followed by a boundary growing method for traversing multioriented text in video. Since Laplacian operation is a second-order derivative, it gives more noisy pixels apart from enhancing text pixels, while Sobel operation is a first-order derivative that gives only high contrast pixels by suppressing low contrast pixels. Therefore, we propose the product of Laplacian and Sobel operations on the input image to obtain actual text pixels without noise as they were shown in [38] to perform sharpening of edge pixels. It is observed that the probability of a pixel being classified as text in probable text matrices obtained by different clustering will be high compared to the probability of a nontext pixel. The advantage of contrast enhancement and the use of Bayesian classifier at the pixel level is that the method works well for different fonts, font size, scripts, orientation of text, and so on in normal cases. However, for exceptional cases such as gigantic fonts and tiny fonts with very short text lines, the method's performance may degrade. With this motivation, we propose a Bayesian classifier to classify text pixels accurately. Further, a boundary growing method is introduced to traverse multioriented text, which works based on the fact that characters in the text line will be aligned in one particular direction with close spatial proximity.

The remainder of this paper is as follows. Different steps for text detection are discussed in Section II. Section III provides experimental results on different datasets to validate the proposed method performance. Conclusions and future work are given in Section IV.

II. Proposed Method

The proposed methodology is described in four subsections. In Section II-A, we calculate the product of Laplacian and Sobel operations on the input image to enhance the text details and it is called the Laplacian–Sobel product (LSP) process. The Bayesian classifier is used for classifying true text pixels based on three probable matrices, as described in Section II-B. The three probable matrices are obtained on the basis of LSP such that high contrast pixels in LSP are classified as text pixels (HLSP), K-means with $k = 2$ of maximum gradient difference of HLSP (K-MGD-HLSP), and K-means of LSP (KLSP). Here, MGD is the difference between maximum and minimum values of a sliding window over HLSP. Text candidates are obtained by intersecting the output of the Bayesian classifier with Canny operation of the input image. The boundary growing method based on a nearest-neighbor concept is described in Section II-C. Section II-D provides geometrical properties of text blocks to eliminate false positives. The steps of the method for multioriented text detection are shown in Fig. 1.

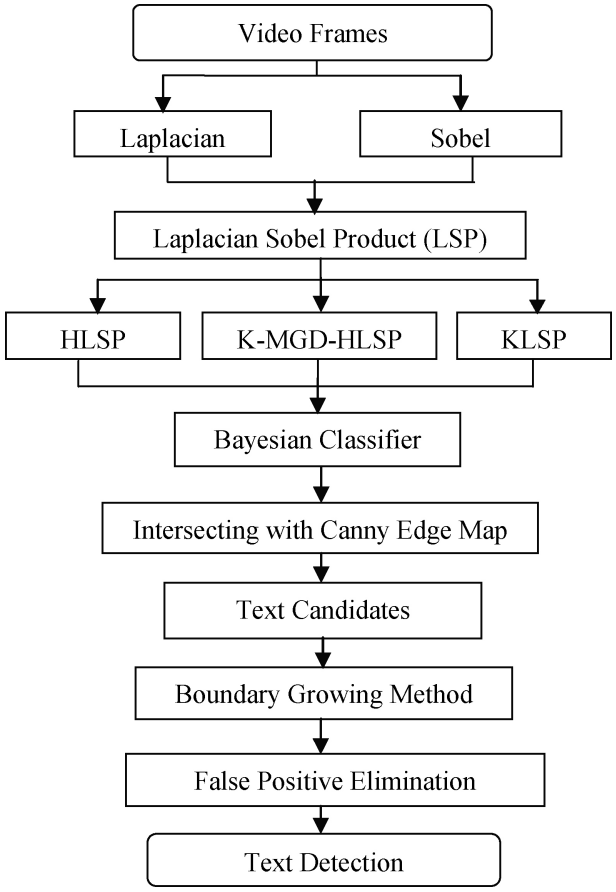


Fig. 1. Flow chart of the proposed method.

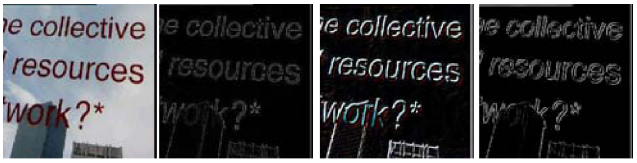


Fig. 2. LSP process. (a) Input. (b) Laplacian. (c) Sobel. (d) LSP.

A. Text Enhancement

We have noticed that text regions typically have a large number of discontinuities from text to background regions and background to text. Thus, this property gives a strong indication of the presence of text. In this section, we try to exploit this property by using Laplacian and Sobel masks and combining the results as follows. We use a 3×3 mask to obtain fine enhanced details of text pixels. The Laplacian operation is a second-order derivative and is used to detect discontinuities in four directions: horizontal, vertical, up-left, and up-right for the image in Fig. 2(a). As a result, it enhances details of both low and high contrast pixels in the image as

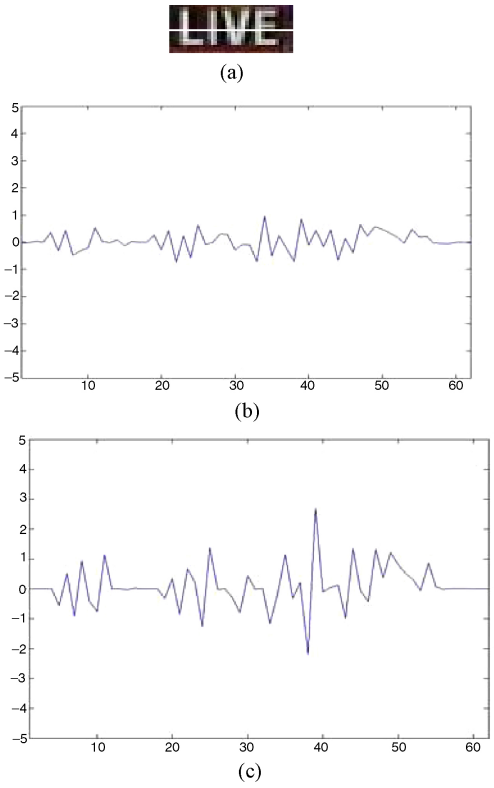


Fig. 3. High positive and negative values. (a) Text sample. (b) Laplacian for (a). (c) LSP for (a).

shown in Fig. 2(b). However, this operation produces more noisy pixels than the Sobel operation [38]. This noise may be the cause for poor performance of the text detection. On the other hand, it is known that Sobel mask operation is a first-order derivative and hence it produces fine details at discontinuities in horizontal and vertical directions [38]. This results in an enhancement at high contrast text pixels, but no enhancement at low contrast text pixels as shown in Fig. 2(c). Thus, we propose a new operation called LSP to preserve details at both high and low contrast text pixels while reducing noise in relatively flat areas as shown in Fig. 2(d). LSP is the product of the results of Laplacian and Sobel operations. This process takes advantage of both Sobel and Laplacian operations to obtain fine details at text pixel areas in the image. This is clearly illustrated in Fig. 3, where (a) is a sample text image, (b) is a graph of Laplacian values versus the row along the middle scan line in (a), and (c) is a graph of LSP values versus the row along the same middle scan line in (a). We can notice from Fig. 3(b) and (c) the high positive and negative peaks for LSP compared to Laplacian alone. This is evident that LSP enhances details at text pixels.

Further, in order to show that LSP is better than Laplacian alone in terms of quantative measures, we conduct experiments using only Laplacian and LSP for text detection. The experimental results will be presented in the experimental section.

B. Bayesian Classifier for True Text Pixel Classification

Classification of exact text and nontext pixel is an important step in text detection as it affects the performance of the text detection methods. Therefore, we consider this as a two-class classification problem and hence we propose a Bayesian classifier method for classifying text and nontext pixels accurately. To estimate the posterior probability of the Bayesian classifier, we need to have *a priori* and conditional probabilities. We estimate conditional probabilities based on three probable text matrices that are: 1) HLSP (P_1); 2) K-MGD-HLSP (P_2); and 3) KLSP (P_3) for which the LSP is the basis as described in the previous section. We choose 100 sample images randomly from our database to produce the three probable text matrices.

Fig. 4. Three probable matrices. (a) HLSP (P_1). (b) K-MGD-HLSP (P_2). (c) KLSP (P_3).

When we look at pixel values in LSP there are text pixels that have high positive and negative values, as is seen in Fig. 3(c). We derive three probable matrices by keeping these values in our mind. HLSP performs classification of text pixels whose magnitudes are greater than 0.5 in the normalized LSP, as shown in Fig. 4(a). HLSP gives high contrast pixels in LSP, but it may lose low contrast text pixels that have negative values.

K-MGD-LSP is the classification of high contrast text pixels by applying K-means with $K = 2$ on MGD of HLSP, where MGD is the difference between the maximum and minimum gradient values in a sliding window over HLSP as shown in Fig. 4(b). K-MGD-HLSP gives high contrast and low contrast pixels since it involves MGD and helps in selecting pixels that have high negative and positive values in LSP. In this case, there are less chances of losing text pixels.

KLSP is the classification of probable text pixels by K-means on LSP, as shown in Fig. 4(c). This may include both text and nontext pixels as it dose not care about negative and positive values of pixels in LSP. In this way, we compute three probable text matrices to estimate posterior probability. The result may improve if we increase the number of probable text matrices. For this paper, we limit it to three that have been shown to work well for our purpose. Note that the three probable text matrices P_1, P_2, and P_3 each represent a binary decision on every text pixel. Collectively, we use the average of the binary decisions to estimate the conditional text probability matrix (TPM), as shown in Fig. 5(a). By the same token, we find three probable nontext matrices N_1, N_2, and N_3 based on the same processes above using the same sample images, representing each a binary decision on every pixel's likelihood of being a nontext. Similarly, we use their collective decisions to estimate the conditional nontext probability matrix (NTPM) as shown in Fig. 5(b).

More specifically, the conditional TPM is defined as follows:

$$P(f(x, y)|T) = \frac{P_1(f(x, y)) + P_2(f(x, y)) + P_3(f(x, y))}{3} \quad (1)$$

Fig. 5. Posterior probability estimation and text candidates. (a) TPM. (b) NTPM. (c) Bayesian result. (d) Text candidates.

where T represents the text class. Similarly, the conditional NTPM is defined as follows:

$$P(f(x, y)|NT) = \frac{N_1(f(x, y)) + N_2(f(x, y)) + N_3(f(x, y))}{3} \tag{2}$$

where NT represents the nontext class. Now, the above conditional probabilities are used to estimate a posterior probability matrix. The Bayesian classifier formula for each pixel $f(x, y)$ is given by

$$P(T|f(x, y)) = \frac{P(f(x, y)|T)P(T)}{P(f(x, y)|T)\,P(T) + P(f(x, y)|NT)\,P(NT)} \tag{3}$$

where $P(T|f(x, y))$ is the probability that a pixel $f(x, y)$ in the resultant output frame of the Bayesian classifier in Fig. 5(c) is a text pixel. $P(T)$ and $P(NT)$ are the *a priori* probabilities calculated based on text and nontext pixels in LSP. We set the following decision:

$$P(T|f(x, y)) \geq 0.5 \rightarrow BC(x, y) = 1$$

where BC is the resultant matrix produced by the Bayesian classifier algorithm as shown in Fig. 5(c). Then, text candidates are obtained by intersecting the Bayesian result with the Canny operation result of the original image as shown in Fig. 5(d).

C. Boundary Growing Method (BGM) for Traversing Multioriented Text

The main problem of multioriented text detection is traversing the text pixels detected by the Bayesian classifier along the text direction to fix a closed bounding box. This is because the complex background in video makes traversing only text pixels more challenging and interesting. Unlike conventional projection profiles used by other text detection methods for horizontal texts, we introduce a new idea of boundary growing method based on the nearest-neighbor concept. The basis for this method is that text lines in the image always appear with characters and words in regular spacing in one direction. The method scans a text candidate image [Fig. 5(d)] from the top left pixel to the bottom right pixel. When the method finds a component during scanning, it fixes a bounding box for that component and allows the bounding box to grow until it reaches a pixel of the component where an adjacent bounding box is formed. This process will continue till the end of the text line. The end of the text line is determined empirically based on the spacing between characters, words, and lines. Fig. 6 illustrates the boundary growing procedure for the first text lines shown in Fig. 2(a), where (a)–(f) show the process of growing bounding boxes along the text direction. This process

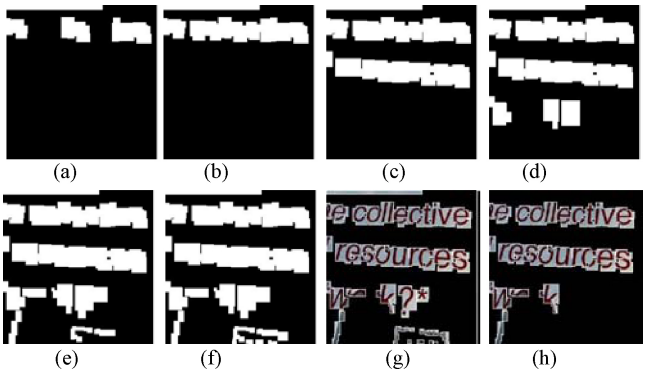


Fig. 6. Boundary growing method. (a) BGM for components. (b) BGM for first line. (c) BGM for second line. (d) BGM for third line. (e) BGM for third line and false positives. (f) BGM for false positives. (g) False positives shown. (h) False positive elimination.

may give false positives as shown in Fig. 6(g), because this process sometimes covers background information. We use a few heuristics such as aspect ratio of the bounding box and edge ratio to eliminate false positives. Some false positives, nevertheless, may still remain as shown in Fig. 6(h) after the elimination process.

D. False Positive Elimination

Since the canny edge map is used for obtaining text candidates, the proposed method may produce a larger number of false positives. It is noted that false positive elimination is challenging and difficult [36]. In this paper, we use geometrical properties of text blocks for the purpose of false positive elimination as these properties are quite common in the literature to use for false positive elimination. Let W, H, A, AR, and EA be the width, height, area, aspect ratio, and edge area of text block B, respectively

$$AR = \frac{W}{H} \tag{4}$$

where $A = W * H$ and

$$EA = \sum_{(i.j)\in B} BC(i, j). \tag{5}$$

If $AR < T_1$ and $EA/A < T_2$, the text block is considered as a false positive; otherwise, it is accepted as a text block. The first rule checks whether the aspect ratio is below a certain threshold. The second rule assumes that a text block has a high edge density due to the transitions between text and background. Here, T_1 and T_2 are determined based on the experimental study given in our earlier work [32], and the same dataset (the 100 sample images that were used for Bayesian classifier training) is used for both T_1 and T_2.

III. Experimental Results

In order to show that the proposed method is effective and robust in terms of metrics and multioriented text lines detection, we have considered a variety of datasets that include images of sports news, low contrast, different fonts and size, different orientations, and so on. As there is no standard database available for video text detection, we create our own

TABLE I
PERFORMANCE LSP VERSUS LAPLACIAN ALONE ON OUR DATA (IN %)

Methods	Recall	Precision	F-Measure	MDR
LSP	0.87	0.74	0.79	0.09
Laplacian alone	0.61	0.66	0.63	0.14

dataset for the purpose of evaluation. We have selected 220 nonhorizontal text images (that include 176 scene text images and 44 graphics text images), 800 horizontal text images (that include 160 Chinese text, 155 scene text, and 485 English text images), and a publicly available video data (Hua's data) [39] comprising of 45 images (that include 12 scene text and 33 graphics text images). We have also experimented on the ICDAR 2003 competition complex scene text dataset [40] containing 251 images to check the effectiveness of our method on camera-based images. In summary, 1020 (800 horizontal and 220 nonhorizontal images) video images, 45 Hua's data, and 251 camera images are used for experimentation and comparative study with the existing methods.

We consider five popular existing methods that are "edge-texture" [19], "gradient" [30], "edge-caption" [33], "edge-color" [34], "color-cluster" [35], and our recently published work in [37] for the purpose of comparative study. The main reason for considering these existing methods is that these methods work with fewer constraints for complex background without a classifier and training as in our proposed method. The parameters involved in these methods are set according to the information given in the respective papers. To find values of the parameters, the same dataset (the 100 sample images that were used for Bayesian classifier training) is used.

We evaluate the performance at the block level, which is a common granularity level in the literature [19], [29], [30], [34], [35]. The following categories are defined for each detected block by a text detection method.

1) *Truly detected block* (*TDB*): a detected block that contains at least one true character. Thus, a TDB may or may not fully enclose a text line.
2) *Falsely detected block* (*FDB*): a detected block that does not contain text.
3) *Text block with missing data* (*MDB*): a detected block that misses more than 20% of the characters of a text line (MDB is a subset of TDB). The percentage is chosen according to [29], in which a text block is considered correctly detected if it overlaps at least 80% with the ground-truth block.
4) *Average processing time* (APT): processing time per frame required for detecting text in the images.

The performance measures are defined as follows.

1) *Recall (R)* = TDB/ATB. Here, ATB is the actual number of text blocks.
2) *Precision (P)* = TDB/(TDB + FDB).
3) *F-measure (F)* = $2 \times P \times R/(P + R)$.
4) MDR = MDB/TDB.

There are two other performance measures commonly used in the literature, *detection rate* and *false positive rate*; however, they can also be converted to recall and precision: recall = detection rate and precision = 1 − false positive rate [30].

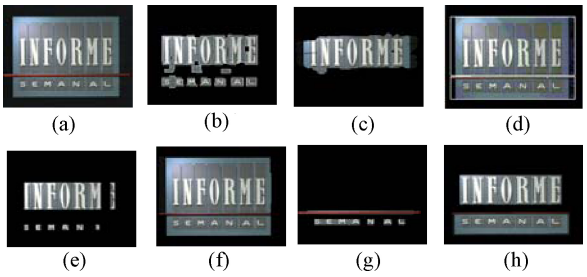


Fig. 7. Results on Hua's dataset. (a) Input. (b) Proposed. (c) Laplacian–Fourier. (d) Edge-caption. (e) Edge-texture. (f) Gradient. (g) Edge-color. (h) Color-cluster.

TABLE II
PERFORMANCE ON HUA'S DATASET (IN %)

Methods	Recall	Precision	F-Measure	MDR	APT(s)
Proposed Method	0.87	0.85	0.85	0.18	5.6
Laplacian–Fourier [37]	0.93	0.81	0.87	0.07	11.7
Edge-caption [33]	0.72	0.82	0.77	0.44	1.13
Edge-texture [19]	0.75	0.54	0.63	0.16	24.9
Gradient [30]	0.51	0.75	0.61	0.13	1.6
Edge-color [34]	0.69	0.43	0.53	0.13	9.2
Color-cluster [35]	0.47	0.44	0.45	0.44	17.2

Hence, only the above five performance measures are used for evaluation including APT. To study the misdetection, we also include MDR as a performance measure and provide discussions on partial detection in all experiments to ensure a fair comparative study.

A. Experiment to Compare LSP and Laplacian

In order to show that LSP helps in achieving better results than Laplacian alone, we conduct an experiment on 1020 video images that include 800 horizontal and 220 nonhorizontal text images. We run the proposed method with Laplacian only and LSP on this dataset and the results are reported in Table I. Table I shows that the proposed method with LSP gives better results than the method with Laplacian alone in terms of recall, precision, F-measure, and MDR because LSP operation helps in locating text correctly and eliminating false positives.

B. Experiment on Hua's Data

We will now test the proposed method on an independent dataset comprising of 45 different images obtained from [39]. While the dataset is small, it provides an objective test of the proposed method in comparison with the other six methods. Fig. 7 shows for the input image in Fig. 7(a), the proposed method, Laplacian–Fourier method, edge-texture, and color-clustering methods detect almost all text in the images, while edge-caption, edge-color, and gradient-based methods fail to detect all texts in the input image as their bounding boxes enclose a lot of background information. Table II shows that the proposed method is better than the other methods in terms of precision. However, the Laplacian–Fourier method is better than the proposed method in terms of recall, F-measure, and MDR at the cost of computations.

Since Hua's data include big font texts as shown in Fig. 7 and it is a small dataset, the Laplacian–Fourier method

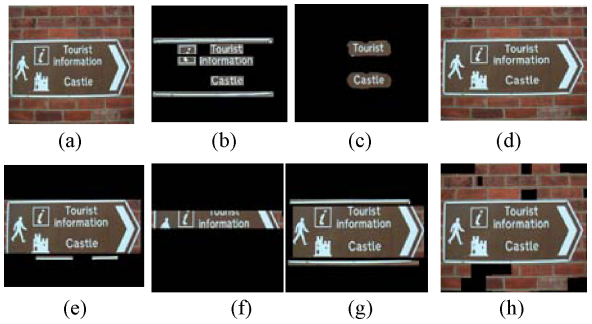


Fig. 8. Results on ICDAR-03 competition data. (a) Input. (b) Proposed. (c) Laplacian–Fourier. (d) Edge-caption. (e) Edge-texture. (f) Gradient. (g) Edge-color. (h) Color-cluster.

TABLE III
PERFORMANCE ON ICDAR 2003 DATA (IN %)

Methods	Recall	Precision	F-Measure	MDR	APT(s)
Proposed Method	0.87	0.72	0.78	0.14	7.9
Laplacian–Fourier [37]	0.86	0.76	0.81	0.13	6.8
Edge-caption [33]	0.66	0.83	0.73	0.26	1.2
Edge-texture [19]	0.53	0.61	0.57	0.24	16.1
Gradient [30]	0.52	0.83	0.64	0.08	1.0
Edge-color [34]	0.67	0.33	0.44	0.43	6.1
Color-cluster [35]	0.60	0.44	0.51	0.45	9.1

TABLE IV
PERFORMANCE ON OUR DATASET
(HORIZONTAL AND NONHORIZONTAL) (IN %)

Methods	Recall	Precision	F-Measure	MDR	APT(s)
Proposed Method	0.87	0.74	0.79	0.13	9.3
Laplacian–Fourier [37]	0.82	0.76	0.78	0.21	9.5
Edge-caption [33]	0.50	0.76	0.60	0.32	1.2
Edge-texture [19]	0.56	0.75	0.64	0.21	22.6
Gradient [30]	0.51	0.87	0.64	0.13	1.7
Edge-color [34]	0.54	0.52	0.52	0.29	6.4
Color-cluster [35]	0.48	0.58	0.52	0.29	17.5

requires more processing time for performing segmentation tasks compared to the proposed method. Due to the problem of segmentation, the Laplacian–Fourier method produces more false positives than the proposed method and hence precision is lower than the proposed method. On the other hand, the proposed method works well irrespective of font size. The gradient method is the best in terms of computational time, but worse in F-measure compared to the proposed method.

C. Experiment on ICDAR 2003 Competition Data

While the proposed method is designed to work with low contrast and low resolution video images, this experiment aims to show how the proposed method and the six other methods perform with high resolution images captured by cameras. Here, we use the benchmark database ICDAR 2003 competition scene text data [40]. The sample results are shown in Fig. 8. It can be seen that the proposed method gives good results with few false positives for the input image shown in Fig. 8(a), while the Laplacian–Fourier method also gives good results with few text line missing. The other existing methods either miss text information or fix improper bounding box for

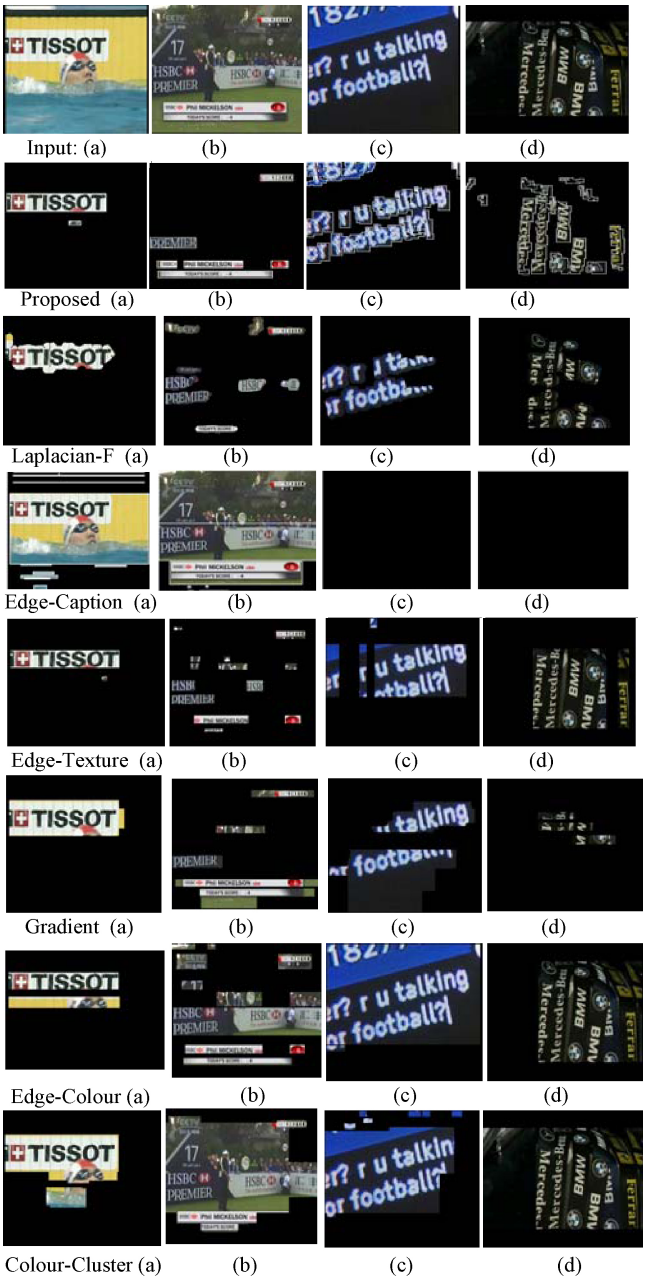


Fig. 9. Results on both horizontal and nonhorizontal text images. (a)–(d) Row 1: sample input images, row 2: results of the proposed method, row 3: results of the Laplacian–Fourier method, row 4: results of the edge-caption method, row 5: results of the edge-texture method, row 6: results of the gradient method, row 7: results of the edge-color method, row 8: results of the color-cluster method.

the text lines. The recall of the proposed method is higher than the Laplacian–Fourier method. Due to more false positives of the proposed method, its precision and F-measure are lower than the Laplacian–Fourier method according to the results in Table III. Comparatively, the other four existing methods perform poorly with the dataset.

D. Experiment on Our Data (Horizontal and Nonhorizontal)

The experiments in Sections III-B and III-C clearly show the superiority of both the present method and our earlier Laplacian–Fourier method over the five existing methods on

horizontal video text data and camera text data. Both our methods are comparable with each other in that the proposed method generally fares better in precision for video data and recall for camera data while our earlier Laplacian–Fourier method gains slightly in F-measure for both the data but at the cost of longer computational time.

To show the proposed method is effective and has the ability to handle both horizontal and nonhorizontal, we combine 800 horizontal and 220 nonhorizontal text images for the purpose of experimentation and comparative study with existing methods. In Fig. 9, the first row (a)–(d) shows a selection of input frames representing both horizontal [Fig. 9(a), (b)] and nonhorizontal text images [Fig. 9(c), (d)], while the remaining rows show the results of the proposed method, Laplacian–Fourier method, edge-caption, edge-texture, gradient, edge-color, and color-cluster methods, respectively. Fig. 9 shows that the Laplacian–Fourier method detects almost all the text lines for horizontal text images, but it misses a few text lines for nonhorizontal text images. It is observed from Fig. 9 that the edge-caption, edge-texture, gradient, edge-color, and color-cluster-based methods detect almost all texts for horizontal text images with fewer false positives but for nonhorizontal images, the methods fix either one bounding box to enclose all the text lines or miss the text lines. This shows that the existing methods are good for horizontal text detection but not good for nonhorizontal text detection. On the other hand, the proposed method detects all the text lines in the input images with few false positives. Therefore, the recall of the proposed method is higher than all other methods, but precision is lower than the Laplacian–Fourier method as shown in Table IV. In addition, the proposed method requires less computational time compared to the Laplacian–Fourier method that requires expensive Fourier transform and connected component labeling to convert complex components into simple components. However, Table IV shows that the gradient method is the best in misdetection compared to the other methods, but worse in F-measure compared to the proposed method. The main reason for poor performance of the five existing methods is that the methods suffer from the fixed constant thresholds for classification of text pixel and the limitation that uniform color characters should be present in text lines.

IV. Conclusion and Future Work

In this paper, we proposed a new video scene text detection method that made use of a new enhancement method using Laplacian and Sobel operations of input images to enhance low contrast text pixels. A Bayesian classifier was used to classify true text pixels from the enhanced text matrix without *a priori* knowledge of the input image. Three probable text matrices and three probable nontext matrices were derived based on clustering and the result of enhancement method. To traverse the multioriented text, we proposed a boundary growing method based on the nearest neighbor concept. Experimentation and comparative study showed that the proposed method outperformed the existing methods in terms of measures, especially on complex nonhorizontal data. However, there are few problems in handling false positives. We planned to extend this method to detection of curve-shaped text lines with good recall, precision, F-measures, and low computational times. Notwithstanding the current limitations that we will deal with in our future research, the contribution of this paper lies in our continued effort in detecting multioriented text lines in videos, which hitherto has not been well explored by others.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive suggestions to improve the quality of this paper.

References

[1] J. Zang and R. Kasturi, "Extraction of text objects in video documents: Recent progress," in *Proc. DAS*, 2008, pp. 5–17.
[2] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: A survey," *Pattern Recognition*, vol. 37, no. 5, pp. 977–997, 2004.
[3] D. Crandall and R. Kasturi, "Robust detection of stylized text events in digital video," in *Proc. ICDAR*, 2001, pp. 865–869.
[4] K. L. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuous adaptive mean shift algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1631–1639, Dec. 2003.
[5] F. Wang, C. W. Ngo, and T. C. Pong, "Structuring low quality videotaped lectures for cross-reference browsing by video text analysis," *Pattern Recognit.*, vol. 41, no. 10, pp. 3257–3269, Oct. 2008.
[6] U. Bhattacharya, S. K. Parui, and S. Mondal, "Devanagari and Bangla text extraction from natural scene images," in *Proc. ICDAR*, 2009, pp. 171–175.
[7] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic detection and recognition of signs from natural scenes," *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 87–99, Jan. 2004.
[8] Y. F. Pan, X. Hou, and C. L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 800–813, Mar. 2011.
[9] Y. F. Pan, X. Hou, and C. L. Liu, "Text localization in natural scene images on conditional random field," in *Proc. ICDAR*, 2009, pp. 6–10.
[10] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. CVPR*, 2010, pp. 2963–2970.
[11] P. P. Roy, U. Pal, J. Liados, and F. Kimura, "Multi-oriented English text line extraction using background and foreground information," in *Proc. DAS*, 2008, pp. 315–322.
[12] K. Jung, "Neural network-based text location in color images," *Pattern Recognit. Lett.*, vol. 22, no. 14, pp. 1503–1515, Dec. 2001.
[13] X. Tang, X. Gao, J. Liu, and H. Zhang, "A spatial-temporal approach for video caption detection and recognition," *IEEE Trans. Neural Netw.*, vol. 13, no. 4, pp. 961–971, Jul. 2002.
[14] A. K. Jain and B. Yu, "Automatic text location in images and video frames," *Pattern Recognit.*, vol. 31, no. 12, pp. 2055–2076, 1998.
[15] K. Jung and J. H. Han, "Hybrid approach to efficient text extraction in complex color images," *Pattern Recognit. Lett.*, vol. 25, no. 6, pp. 679–699, Apr. 2004.
[16] C. W. Lee, K. Jung, and H. J. Kim, "Automatic text detection and removal in video sequences," *Pattern Recognit. Lett.*, vol. 24, no. 15, pp. 2607–2623, Nov. 2003.
[17] H. Li, D. Doremann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 147–156, Jan. 2000.
[18] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 385–392, Apr. 2000.
[19] C. Liu, C. Wang, and R. Dai, "Text detection in images based on unsupervised classification of edge-based features," in *Proc. ICDAR*, 2005, pp. 610–614.
[20] X. H. Ma, W. W. Y. Ng, P. P. K. Chan, and D. S. Yeung, "Video text detection and localization based on localized generalization error model," in *Proc. ICMLC*, 2010, pp. 2161–2166.
[21] M. Anthimopoulos, B. Gatos, and I. Pratikakis, "A hybrid system for text detection in video frames," in *Proc. DAS*, 2008, pp. 286–292.

[22] C. Jung, Q. Liu, and J. Kim, "A stroke filter and its applications to text localization," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 114–122, 2009.
[23] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A robust wavelet transform based technique for video text detection," in *Proc. ICDAR*, 2009, pp. 1285–1289.
[24] Q. Ye, Q. Huang, W. Gao, and D. Zhao, "Fast and robust text detection in images and video frames," *Image Vis. Comput.*, vol. 23, no. 6, pp. 565–576, Jun. 2005.
[25] P. Shivakumara, W. Huang, C. L. Tan, and P. Q. Trung, "Accurate video text detection through classification of low and high contrast images," *Pattern Recognit.*, vol. 43, no. 6, pp. 2165–2185, Jun. 2010.
[26] P. Shivakumara, T. Q. Phan, and C. L. Tan, "New Fourier-statistical features in RGB space for video text detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1520–1532, Nov. 2010.
[27] V. Wu, R. Manmatha, and E. M. Riseman, "Textfinder: An automatic system to detect and recognize text in images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1224–1229, Nov. 1999.
[28] R. Lienhart and A. Wernickle, "Localizing and segmenting text in images and videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 4, pp. 256–268, Apr. 2002.
[29] D. Chen, J. M. Odobez, and J. P. Thiran, "A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning," *Signal Process. Image Commun.*, vol. 19, no. 3, pp. 205–217, 2004.
[30] E. K. Wong and M. Chen, "A new robust algorithm for video text extraction," *Pattern Recognit.*, vol. 36, pp. 1397–1406, Jun. 2003.
[31] M. R. Lyu, J. Song, and M. Cai, "A comprehensive method for multilingual video text detection, localization, and extraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 243–255, Feb. 2005.
[32] T. Q. Phan, P. Shivakumara, and C. L. Tan, "A Laplacian method for video text detection," in *Proc. ICDAR*, 2009, pp. 66–70.
[33] J. Zhou, L. Xu, B. Xiao, and R. Dai, "A robust system for text extraction in video," in *Proc. ICMV*, 2007, pp. 119–124.
[34] M. Cai, J. Song, and M. R. Lyu, "A new approach for video text detection," in *Proc. ICIP*, 2002, pp. 117–120.
[35] V. Y. Mariano and R. Kasturi, "Locating uniform-colored text in video frames," in *Proc. ICPR*, 2000, pp. 539–542.
[36] W. Kim and C. Kim, "A new approach for overlay text detection and extraction from complex video scene," *IEEE Trans. Image Process.*, vol. 18, no. 2, pp. 401–411, Feb. 2009.
[37] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A Laplacian approach to multi-oriented text detection in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 412–419, Feb. 2011.
[38] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Delhi, India: Pearson Education (Singapore) Pte. Ltd., 2002, pp. 128–141.
[39] X. S. Hua, L. Wenyin, and H. J. Zhang, "An automatic performance evaluation protocol for video text detection algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 498–507, Apr. 2004.
[40] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proc. ICDAR*, 2003, pp. 682–687.

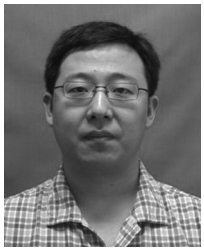
Palaiahnakote Shivakumara received the B.Sc., M.Sc., M.Sc. (technology by research), and Ph.D. degrees in computer science from the University of Mysore, Mysore, India, in 1995, 1999, 2001, and 2005, respectively, and the B.Ed. degree from Bangalore University, Bangalore, India, in 1996.

He is currently a Research Fellow with the Department of Computer Science, School of Computing, National University of Singapore, Singapore. From 1999 to 2005, he was a Project Associate with the Department of Studies in Computer Science, University of Mysore, where he conducted research on document image analysis, including document image mosaicing, character recognition, skew detection, face detection, and face recognition. From 2005 to 2007, he was a Research Fellow with the Department of Computer Science, School of Computing, National University of Singapore, where he worked on the field of image processing and multimedia. In 2007, he was a Research Consultant with Nanyang Technological University, Singapore, for six months working on image classification. He has published around 90 research papers in national, international conferences and journals. He has been a reviewer for several conferences and journals. His current research interests include image processing, pattern recognition, including text extraction from videos, document image processing, biometric applications, and automatic writer identification.

Rushi Paduman Sreedhar was an Undergraduate Student Researcher with the Department of Computer Science, School of Computing, National University of Singapore, Singapore. His current research interests include image and video analysis.

Trung Quy Phan is currently pursuing the Graduate degree with the Department of Computer Science, School of Computing, National University of Singapore, Singapore.

He is currently a Research Assistant with the School of Computing, National University of Singapore. His current research interests include image and video analysis.



Shijian Lu received the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2005.

Currently, he is a Senior Research Fellow with the Institute for Infocomm Research, Agency for Science, Technology, and Research, Singapore. He has published over 40 peer-reviewed journal and conference papers. His current research interests include document image analysis and medical image analysis.

Dr. Lu is a member of PREMIA, Singapore, which is a member of IAPR.

Chew Lim Tan (SM'02) received the B.Sc. (hons.) degree in physics from the University of Singapore, Singapore, in 1971, the M.Sc. degree in radiation studies from the University of Surrey, Surrey, U.K., in 1973, and the Ph.D. degree in computer science from the University of Virginia, Charlottesville, in 1986.

He is currently a Professor with the Department of Computer Science, School of Computing, National University of Singapore, Singapore. His current research interests include document image analysis, text and natural language processing, neural networks, and genetic programming. He has published more than 360 research publications in these areas.

Dr. Tan is an Associate Editor of *Pattern Recognition* and the *ACM Transactions on Asian Language Information Processing*, and is an Editorial Member of the *International Journal on Document Analysis and Recognition*. He is a member of the Governing Board of the International Association of Pattern Recognition.