# An Automatic Performance Evaluation Protocol for Video Text Detection Algorithms

Xian-Sheng Hua, Liu Wenyin, *Senior Member, IEEE*, and Hong-Jiang Zhang, *Fellow, IEEE*

***Abstract*—Text presented in videos provides important supplemental information for video indexing and retrieval. Many efforts have been made for text detection in videos. However, there is still a lack of performance evaluation protocols for video text detection. In this paper, we propose an objective and comprehensive performance evaluation protocol for video text detection algorithms. The protocol includes a positive set and a negative set of indices at the textbox level, which evaluate the detection quality in terms of both location accuracy and fragmentation of the detected textboxes. In the protocol, we assign a detection difficulty (DD) level to each ground truth textbox. The performance indices can then be normalized with respect to the textbox DD level and are therefore tolerant to different ground-truth difficulties to a certain degree. We also assign a detectability index (DI) value to each ground-truth textbox. The overall detection rate is the DI-weighted average of the detection qualities of all ground-truth textboxes, which makes the detection rate more accurate to reveal the real performance. The automatic performance evaluation scheme has been applied to performance evaluation of a text detection approach to determine the best thresholds that can yield the best detection results. The protocol has also been employed to compare the performances of several text detection systems. Hence, we believe that the proposed protocol can be used to compare the performance of different video/image text detection algorithms/systems and can even help improve, select, and design new text detection methods.**

***Index Terms*—Performance evaluation, video text detection.**

## I. Introduction

THE rapid growth of video data leads to an urgent demand for efficient and true content-based browsing and retrieving systems. In response to such needs, various video content analysis schemes using one or a combination of image, audio, and textual information in the videos have been proposed to parse, index, or abstract massive amounts of data [1], [2]. Among these information sources, text present in the video frames can provide important supplemental information for indexing and retrieval. For example, with the help of the extracted text related to the news, the news videos can be segmented and catalogued more accurately in the sense of semantics. For MTV indexing, the recognized caption of the song cannot only be used for retrieval and indexing, but also tells us important information about the start and end time of each song. Moreover, if we detect and recognize the time stamp in analog home videos, it can be used as metadata like those in digital videos, which will be greatly helpful for home video clustering and browsing [3].

A great deal of effort has been put into text detection in videos and images [2], [4], [5], [7], [8], [19]–[21]. Current text detection approaches can be classified into two categories. The first category is connected component-based methods, which are based on the observation that text is represented nearly with uniform color or intensity in the images or video frames. These methods can locate text quickly but encounter difficulties when the text is embedded in complex background or touches other graphical objects [4], [19], [20]. Unfortunately, this case often happens in videos. The second category is texture classification-based [5], [7], [8], [19], [21]. However, it is hard to find accurate boundaries of text areas and false alarms often exist in "text-like" texture areas. This case also happens often. In addition, there are also several methods that try to locate text areas in the DCT compressed domain [2], but essentially they are texture-based schemes.

Performance evaluation (PE) has been gaining acceptance in many areas of computer vision and is becoming of crucial importance in producing robust techniques [9], [10]. We view performance as a set of interesting metrics on the output data that an algorithm or a system produces. Usually, these metrics are expressed in terms of the difference between the expected output and the actual output of the algorithm or system. They may help compare, select, improve, and even design new methods to be applied in new systems designed for some specific application [11]. An overview of PE can be found in [13].

A great deal of research has been performed for PE in the field of document analysis and recognition (DAR) [9]–[12], [14]. However, few PE schemes for video/image text detection are available in the literature. Almost all of the papers that address video or image text detection also use some methods to evaluate the performance of the proposed algorithms, but most of them use a simple "precision-recall" ratio (or recall ratio only) of detected textboxes to demonstrate the efficiency of the algorithms [5]–[8]. Instead of measuring the accuracy of detected textboxes, Wu *et al.* proposed an automatic evaluation algorithm that uses the percentage of the total number of characters that have been detected as a measure of the detection accuracy, in which a character is considered to be detected if it is completely covered by a detected textbox [20]. Lienhart *et al.* measures the quality of text detection system with regard to the main objective, which is not to discard character pixels [19], [21]. In fact, each of these evaluation schemes are special or simpler cases of our PE protocol, as we will show later in this paper.

Text detection in videos/images is different from DAR due to the low resolution, complex background, and variety of the

Manuscript received November 11, 2003; revised December 9, 2003. This paper was recommended by Associate Editor Q. Tian.

X.-S. Hua and H.-J. Zhang are with Microsoft Research Asia, Beijing 100080, China (e-mail: xshua@microsoft.com; hjzhang@microsoft.com).

L. Wenyin is with the Department of Computer Science, City University of Hong Kong, Kong Kong SAR, (e-mail: csliuwy@cityu.edu.hk).

Digital Object Identifier 10.1109/TCSVT.2004.825538

font/size. In this paper, we propose an objective, comprehensive, and difficulty-tolerant performance evaluation protocol for video/image text detection algorithms based on the idea in [12], which is originally developed for the PE of text segmentation algorithms from engineering drawings. We have made remarkable modifications to make it suitable for text detection from video clips. The protocol includes a positive set and a negative set of indices at the textbox level, and it evaluates the detection quality in terms of both location accuracy and fragmentation of the detected textboxes. Currently, evaluation of character recognition is not included in it. In the protocol, we assign a detection difficulty (DD) level to each ground-truth textbox. The performance indices can then be normalized with respect to the textbox DD and are therefore tolerant to different degrees of ground-truth difficulty. We also assign a detectability index (DI) value to each ground-truth textbox. The overall detection rate is the DI-weighted average of the detection qualities of all ground-truth textboxes, which makes the detection rate more accurate to reveal the real performance. Finally, we apply the proposed automatic performance evaluation scheme on a text detection approach to determine the best parameters that can yield the best detection results. We have also compared the performances of several video text detection algorithms using the proposed protocol.

The remainder of this paper is organized as follows. Section II gives a detailed description on the data structure of the ground-truth data. The automatic evaluation scheme is presented in Section III. In the experiments of Section IV, we have used the PE protocol to determine the best parameters in one video text detection algorithm and compared the performances of several different video text detection algorithms. We summarize this paper in Section V.

## II. Ground Truthing

The main purpose of video text detection is optical character recognition (OCR) (e.g., [5], [6]) since the final target is text words, which are useful for video indexing, retrieval, and understanding. Therefore, the target to evaluate is not the segmented text images themselves, though some applications may use different image areas for image processing, like de-noising. In some applications, only knowing whether there is text is also useful. However, our main objective is that the segmented text area should be good for recognition.

To evaluate the performance of text detection algorithms, we also need the three elements that were proposed in [15]. Fist of all, we need to know the expected output (ground truth), so that it can be compared with the actual output (the detection results). Second, a matching method is needed to match each ground truth textbox with one or more objects from the detected objects set. Finally, quantitative indices that measure the interesting metrics should be defined uniformly. In this section, we will give a detailed description of the first element. The remaining two are to be discussed in Section III.

Suppose $g$ is a ground-truth textbox. The ground truth information of $g$ contains the following attributes:

1) **Textbox Location** (Left(g), Top(g), Right(g) and Bottom(g)).

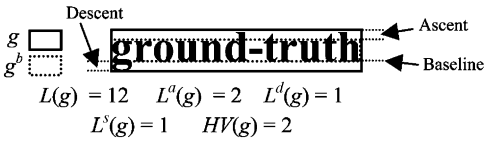


Fig. 1. Illustration of the definition of character height variance.

2) **Textbox Height** $(H(g))$: Equal to $(\mathrm{Bottom(g)} - \mathrm{Top(g)} + 1)$.
3) **Textbox Width** $(W(g))$: Equal to $(\mathrm{Right(g)} - \mathrm{Left(g)} + 1)$.
4) **Text String** $(STR(g))$: The ASCII string of the text in $g$ (it is not used in our protocol).
5) **Text Length** $(L(g))$: The number of the characters (i.e., the length of the text string) in $g$.
6) **Character Height Variation** $(HV(g))$: $HV$ is employed to measure the degree of the height variation of the characters in the textbox. Typically, the larger the HV is, the more difficult the textbox is correctly and accurately detected. Before we present the exact definition of $HV(g)$, we have to explain some terms. Denote by $g^b$ the base-box of $g$ is a bounding box, whose left and right sides are the same as those of $g$. The bottom side of $g^b$ is the baseline of the characters in $g$, and $H(g^b) = \arg\max_h\{N(h)\}$, where $N(h)$ denotes the number of characters in $g^b$ whose ascents are equal to $h$. Let $L^a(g)$, $L^d(g)$, and $L^s(g)$ denote the number of characters, whose ascents are not less than 4/3 of $H(g^b)$ (e.g., 'd' and 'h' in Fig. 1), whose descents are not less than 1/3 of $H(g^b)$ (e.g., 'g'), and whose heights are not more than 2/3 of $H(g^b)$ (e.g., '-'), respectively.

   The character height variance $(HV(g))$ is then defined as follows: $HV(g)$ is initiated with zero. If $L^a(g)$ is nonzero and not more than 1/3 of $L(g)$, $HV(g)$ is increased by 1. If $L^d(g)$ is nonzero and not more than 1/3 of $L(g)$, $HV(g)$ is increased by 1. If $L^s(g)$ is nonzero and not less than 1/3 of $L(g)$, then $HV(g)$ is increased by 1.
7) **Skew Angle** $(SA(g))$: The skew angle of the text font in $g$. Typically, the greater the skew angle is, the more difficult the textboxes that can be detected.
8) **Color and Texture** $(CT(g))$: If the text string in $g$ is nonhomochromous or textured, $CT(g)$ is set to 1; otherwise, it is set to zero. Generally, homochromous text strings are easier to correctly detect.
9) **Background Complexity** $(BC(g))$: We extend $g$ in each direction for ten pixels and thus form an extended textbox denoted by $g^e$. The background complexity of $g$ is described by

$$BC(g) = \frac{\left(\frac{E(g^e \cap g^c)}{A(g^e \cap g^c)}\right)}{\left(\frac{E(g)}{A(g)}\right)} \tag{1}$$

   where $g^c$ is the complement of $g$, and $A(x)$ and $E(x)$ denote the area of $x$ and the number of Sobel edge points in $x$, respectively. The more complex the background is, the more difficult the text that can be detected.

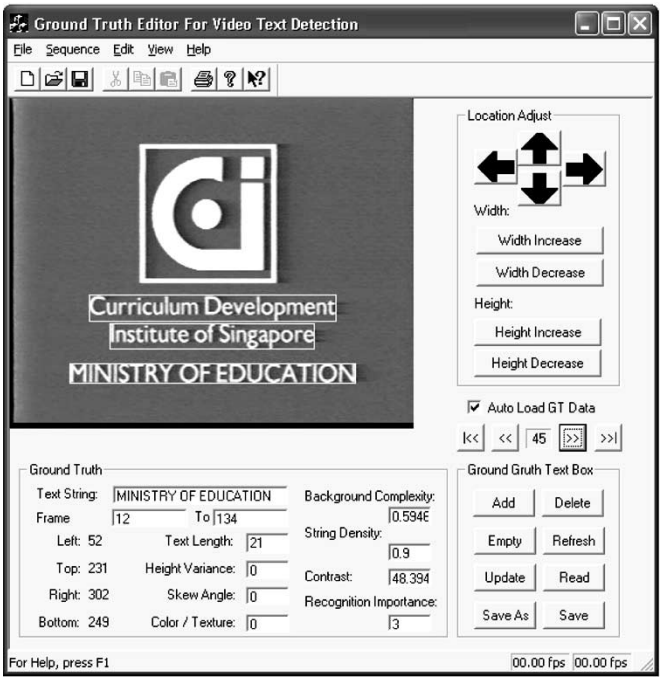


Fig. 2. Ground truthing tool (ground-truth generator).

10) **String Density** $(SD(g))$: Denote by $CW(g)$ the total width of all characters in $g$, $SD(g)$ is defined as $CW(g)/W(g)$. The denser the string is, the easier it is to be detected.
11) **Contrast** $(C(g))$: The contrast of $g$ is described by the intensity variance of all pixels in $g^e$. Typically, the less the contrast is, the more difficult text can still be detected.
12) **Recognizability Index** $(RI(g))$: $RI(g)$ is assigned to one of the four values from 0 to 3. For most cases, $RI(g)$ is set to 2. If more than one third of the characters in $g$ can hardly be recognized by human beings, $RI(g)$ is set to 0. If two thirds or more but not all of them can be recognized, $RI(g)$ is set to 1. If it is very clear, $RI(g)$ is set to 3. As aforementioned, if the objective of the evaluation does not concern whether the text could be recognized or not, we may turn off this attribute by setting it always equal to the highest value, i.e., 3.

The ground truth data used in our experiments are 45 video clips (6750 frames) excerpted from the MPEG-7 Video Content Set. Twnety-nine of them are from V3, seven are from V4, and nine are from V14. The owner of V3 and V4 is Spanish TV RTVE, and V14 comes from Ministry of Education of Singapore. There are in total 158 textboxes in the 45 clips and 128 of them are human-recognizable. The maximum number of textboxes in one clip is 21. There are also three clips that contain no text at all. We use a tool named Ground Truth Generator (shown in Fig. 2) developed by us to semi-automatically collect the ground-truth textboxes and their attributes. In the ground truthing process, we first open a video clip and then draw a bounding box for each text line in the clip by dragging the mouse over the text area. Next, we fill in the attribute values of "Text String", "Height Variance", "Skew Angle", "Color/Texture", "String Density", and "Recognizability Index" manually. It should be mentioned here that although some methods, e.g., [19], use temporal information in consecutive video frames to locate textboxes, our performance evaluation protocol treats the textboxes in each individual frame separately for the sake of reducing computing complexity. However, our ground truthing tool, which is an embedded Microsoft DirectShow-based video player that supports frame-by-frame playing, can assist users to label the start/end frame number of the textbox manually also, if the textbox is fixed on the screen across multiple frames. If the textbox is moving, it is labeled frame-by-frame separately. Once clicking the button "Add", the tool will automatically fill in the attribute values of "Location", "Text Length", "Background Complexity", and "Contrast" of the text line and add all attribute values of the current text line to the ground-truth file. After labeling all text lines in the clip, the ground truths are saved to a file. There are also some functions in the tool for editing the ground truth data. The user can continue the ground truthing process by clicking the "≫" button to go to the next clip.

Since the textboxes are considered separately for each individual frame, if the outputs of the to-be-evaluated algorithms are also represented as the start/end frame number of the textboxes, they are converted into frame-by-frame representation in the same way before evaluation. If we regard the same textbox across multiple frames as separate textboxes, we in total have 19 853 textboxes, and 16 035 of them are human-recognizable. These values are regarded as the total textbox numbers in the following sections.

Fig. 3 shows some example frames in the ground truth set. The attribute values of the first two (counting from top to bottom) ground-truth textboxes in Fig. 3(a)–(c) are listed in Table I.

## III. Performance Evaluation Scheme

In this section, we mainly define a set of performance indices for text detection algorithms. The matching process for the PE scheme is also explained in this section.

### A. Detection Difficulty

Segmentation of text from video frames is more complex than that of conventional OCR. Difficulties are due to a variety of factors, including text/image mixture, low image quality, character connectivity, complex background, and variation in text location, size, and font.

The detection difficulty (DD) is an index that expresses the difficulty of detecting a ground truth textbox. We define DD level $(\mathrm{L_{DD}})$ for a ground truth textbox depending on the following factors:

1) **Initial Level**: The initial DD level for each ground truth textbox is set to 1.
2) **Textbox Height**: Suppose $H_N$ denotes the normal height (i.e., the average font height of common video text) of the text string. If $H(g)$ is larger than $3H_N$ or smaller than 1/2 of $H_N$, $L_{DD}(g)$ is increased by 1.
3) **Textbox Width**: Suppose that $W_N$ denotes the normal width of one character (i.e., the average character width

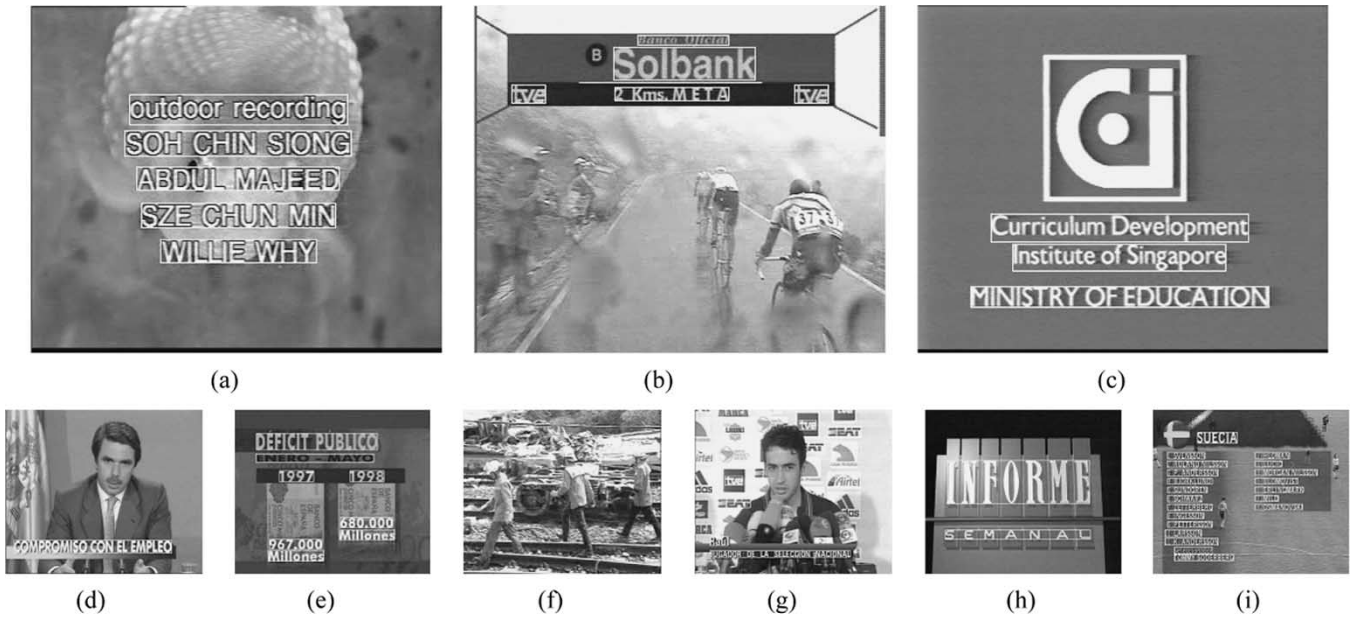


Fig. 3. Some examples of ground-truth data.

TABLE I
ATTRIBUTE VALUES OF SOME GROUND-TRUTH TEXTBOXES

| Fig. | No. | *STR* | *Left* | *Top* | *Right* | *Bottom* | *H* | *W* | *L* | *HV* | *SA* | *CT* | *BC* | *SD* | *C* | *RI* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 (a) | a1 | outdoor recording | 89 | 73 | 272 | 99 | 27 | 184 | 16 | 2 | 0 | 0 | 0.06 | 0.9 | 47.5 | 2 |
| | a2 | SOH CHIN SIONG | 85 | 103 | 275 | 125 | 23 | 91 | 12 | 0 | 0 | 0 | 0.28 | 0.8 | 52.3 | 2 |
| | a3 | ABDUL MA JEED | 95 | 133 | 265 | 153 | 21 | 151 | 11 | 0 | 0 | 0 | 0.54 | 0.8 | 58.1 | 3 |
| | a4 | SZE CHUN MIN | 99 | 163 | 262 | 184 | 22 | 164 | 10 | 0 | 0 | 0 | 0.24 | 0.8 | 57.7 | 3 |
| | a5 | WILLIE WHY | 115 | 193 | 246 | 213 | 21 | 132 | 9 | 0 | 0 | 0 | 0.21 | 0.9 | 55.9 | 3 |
| 3 (b) | b1 | Banco O??c?a? | 142 | 24 | 219 | 33 | 10 | 78 | 12 | 0 | 0 | 0 | 0.22 | 0.9 | 69.8 | 1 |
| | b2 | SolBank | 122 | 34 | 241 | 62 | 29 | 120 | 7 | 1 | 0 | 0 | 0.00 | 0.9 | 52.1 | 3 |
| | b3 | tve (*left*) | 37 | 65 | 66 | 81 | 17 | 30 | 3 | 0 | 0 | 1 | 1.33 | 1 | 62.0 | 2 |
| | b4 | 2Kms. META | 123 | 67 | 219 | 79 | 13 | 97 | 9 | 1 | 0 | 0 | 0.46 | 0.5 | 67.3 | 2 |
| | b5 | tve (*right*) | 273 | 65 | 302 | 81 | 17 | 30 | 3 | 0 | 0 | 1 | 0.87 | 1 | 83.3 | 2 |
| 3 (c) | c1 | Curriculum Development | 69 | 171 | 285 | 195 | 25 | 217 | 21 | 2 | 0 | 0 | 0.09 | 0.9 | 41.5 | 3 |
| | c2 | Institute of Singapore | 87 | 196 | 266 | 219 | 24 | 180 | 20 | 2 | 0 | 0 | 0.12 | 0.9 | 42.6 | 3 |
| | c3 | MINISTRY OF EDUCATION | 52 | 231 | 302 | 249 | 19 | 148 | 19 | 0 | 0 | 0 | 0.60 | 0.9 | 48.4 | 3 |

**Note**: "?" represents a character that can hardly be recognized by a human being.

of common video text). If $W(g)$ is smaller than $4W_N$, $L_{DD}(g)$ is increased by 1.

4) **Character Height Variation**:$L_{DD}(g)$ is increased by $HV(g)$;
5) **Skew Angle**: If $SA(g)$ is larger than $\pi/6$, $L_{DD}(g)$ is increased by 1.
6) **Color and Texture**: If $CT(g)$ is nonzero, $L_{DD}(g)$ is increased by 1.
7) **Background Complexity**: If $BC(g)$ is larger than 1/2, $L_{DD}(g)$ is increased by 1.
8) **String Density**: If $SD(g)$ is less than 1/2, $L_{DD}(g)$ is increased by 1.
9) **Contrast**: If $C(g)$ is less than $V_N$, $L_{DD}(g)$ is increased by 1, where $V_N$ is a normal contrast value.

In the above descriptions, $H_N$, $W_N$, and $V_N$ can be determined by statistic analysis. However, for the sake of simplicity, we directly assign $H_N$, $W_N$, and $V_N$ to 15, 15, and 45, respectively. In addition, the increment steps can also be determined by experiments. Again they are all set to 1 for simplicity.

### *B. Detectability Index*

Some textboxes may hardly be recognized by OCR software or even by human beings due to the low resolution and complex background. If the detection algorithm missed such textboxes, it is reasonable that we do not take them into consideration when we count the detection rate. On the contrary, if they are detected, we do not regard them as false alarms. Hence, in order to get more reasonable evaluation results, we should define a detectability index (DI) for each ground truth textbox, which indicates its importance degree to which it should be detected correctly.

Generally, the more characters in the textbox and the more recognizable the characters are, the more important for us to correctly detect it. Accordingly, we define DI of a ground truth textbox $g$ as

$$DI(g) = L(g) \cdot RI(g). \tag{2}$$

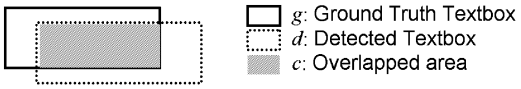


Fig. 4. Matching of a ground truth textbox and a detected textbox.


Fig. 5. Illustration of using $E(x)$ versus $A(x)$. The inner bounding box is the ground truth textbox.

The overall detection rate will be the DI-weighted average of the detection qualities of all ground truth textboxes, which will be presented in Section III-D.

### C. Textbox Detection Qualities

The matching of the ground truth and the detected results is based on their overlap area. Suppose that $g$ and $d$ are a ground truth textbox and a detected textbox, respectively. They are considered matched if their areas overlap at least partially. The overlapped area is denoted by $c$ (as shown in Fig. 4).

The textbox detection quality of the overlapped area $c$ is defined as

$$Q_o(c) = \frac{1 - E(d-c)}{E(d)} \tag{3}$$

where $(d-c)$ means $d \cap c^c$ ($c^c$ denotes the complement of $c$). The DD-tolerance textbox detection quality $(Q_{DD}(c))$ is a function of the text detection quality of the overlapped area $c(Q_o(c))$ and detection difficulty $(L_{DD}(g))$. The greater $Q_o(c)$ and $L_{DD}(g)$ are, the greater $Q_{DD}(c)$ is. In this implementation, it is defined as (the same as a similar definition in [12])

$$Q_{DD}(c) = Q_o(c)^{\frac{1}{\sqrt{L_{DD}(g)}}}. \tag{4}$$

Since a single ground truth textbox may be segmented as several textboxes, its detection quality is defined in terms of two elements. The first element, defined as

$$Q_b(g) = \frac{\sum_{k \in D(g)} (Q_{DD}(k \cap g) E(k \cap g))}{\max\left(E(g), \sum_{k \in D(g)} E(k \cap g)\right)} \tag{5}$$

is the textbox's basic quality $(Q_b(g))$, which reveals the ratio of the texts that are covered by detected textboxes. This element is very close to the measure in [19] and [21], which measures the ratio of detected character pixels. Also, if we consider the number of characters that are covered by detected textboxes, it is the same as the measure in [20], while if we consider whether the whole ground truth textbox is covered (or almost covered) by a detected textbox, it is degraded to the most commonly used measure, or "recall rate," of the groundtruth textboxes.

The second element, defined as

$$Q_{fr}(g) = \frac{\sqrt{\sum_{k \in D(g)} E(k \cap g)^2}}{\sum_{k \in D(g)} E(k \cap g)} \tag{6}$$

is the fragmentation quality (FQ) $(Q_{fr}(g))$, which reveals the extent to which the ground truth textboxes are split by the detected textboxes. This element could be roughly estimated by the number of elements in $D(g)$, which denotes the set of detected textboxes that fully or partially overlap the ground truth textbox $g$. The more elements in $D(g)$, the lower the FQ is. In this implementation, we use a more concise measure [(6)] which can also reveal how the groundtruth textbox is fragmented. The more detected textboxes are fully or partially overlapped with the groundtruth textbox, and the more uniformly the groundtruth textbox is segmented (i.e., the distribution of $\{E(k \cap g), k \in D(g)\}$), the lower the FQ is.

The total textbox detection quality of $g$ is defined as

$$Q(g) = Q_b(g) Q_{fr}(g). \tag{7}$$

It should be mentioned here that we use $E(x)$ in (3), (5) and (6) instead of $A(x)$, because it can obtain more reasonable evaluation results by using $E(x)$. If the detected textbox is larger than the ground truth textbox but has no or little effect for later recognition (see Fig. 5), we should assign a high quality value to it. For example, the detection quality of the detected textbox in Fig. 5 (the outer box) is about 0.5 if we use $A(x)$, but when we use $E(x)$, $Q(g) \approx 1$.

### D. Overall Detection Rate

Let $T_g$ denotes the set of all ground truth textboxes in the testing data. The text detection rate ($D$) of the entire testing set is the DI-weighted average of the detection qualities of all ground truth textboxes as follows:

$$D = \frac{\sum_{g \in T_g} Q(g) DI(g)}{\sum_{g \in T_g} DI(g)}. \tag{8}$$

### E. False Alarm Rate

Similarly, denoting by $G(d)$ the set of ground truth textboxes that fully or partially overlap the detected textbox $d$, the basic quality of $d$ is defined as

$$Q_b(d) = \frac{\sum_{k \in G(d)} (Q_{LD}(k \cap d) E(k \cap d))}{\max\left(E(d), \sum_{k \in G(d)} E(k \cap d)\right)}. \tag{9}$$

The FQ of $d$ is defined as

$$Q_{fr}(d) = \frac{\sqrt{\sum_{k \in G(d)} E(k \cap d)^2}}{\sum_{k \in G(d)} E(k \cap d)}. \tag{10}$$

The total detection quality of $d$ is then defined as

$$Q(d) = Q_b(d) Q_{fr}(d). \tag{11}$$

Since the detection quality of $d$ reflects the degree of $d$ being a correct detection, $1 - Q(d)$ therefore reflects the degree of $d$ being a false alarm. Hence, the false alarm rate of $d$ is defined as

$$F(d) = 1 - Q(d). \tag{12}$$

The false alarm rate of the entire testing data is the $E(d)$-weighted average of false alarm rates of all detected textboxes from the testing data, denoted by $T_d$, as follows:

$$F = \frac{\sum_{d \in T_d} F(d) E(d)}{\sum_{d \in T_d} E(d)}. \tag{13}$$

### F. Combined Text Detection Index

The combined text detection index (TDI) is

$$\text{TDI} = \beta D + (1 - \beta)(1 - F) \tag{14}$$

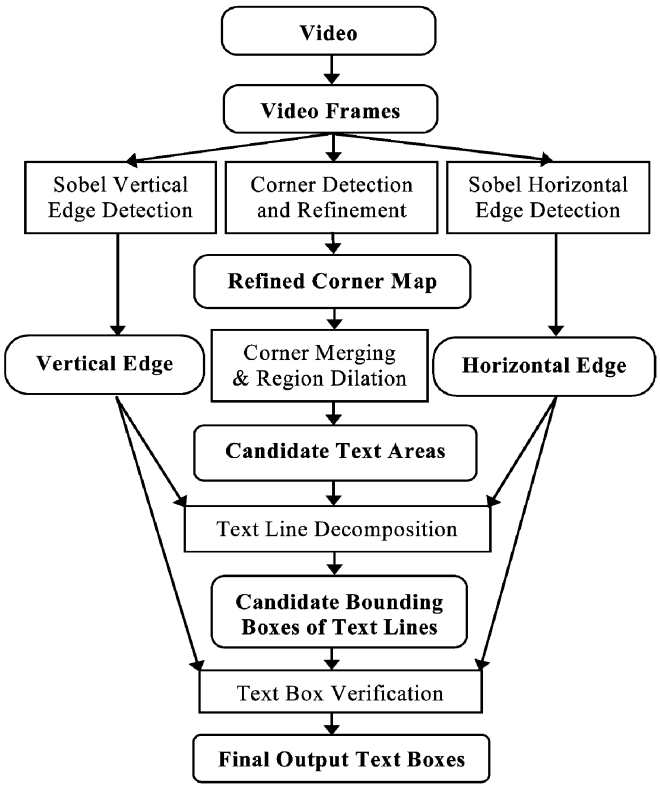


Fig. 6. Flowchart of the text detection algorithm.

where $\beta$ is the relative importance of detection and false alarm. With appropriate values of $\beta$, TDI gives a quantitative performance evaluation of the tested algorithm. The higher the TDI, the better the algorithm.

It is interesting that, if nothing is detected, $F$ is 0 and TDI is 0.5, and, sometimes, some not so good detection is less than 0.5. This is not compatible or normalized with human vision. So our new proposal is using a geometric average, as follows:

$$\mathrm{TDI}_G = D^{\beta}(1-F)^{(1-\beta)}. \tag{15}$$

## IV. Experiments

### A. Video Text Detection Algorithm Used for Evaluation

To evaluate our proposed performance evaluation protocol, we apply it to a video text detection algorithm [16] to determine the best thresholds in the algorithm. Here we will give a brief introduction to this approach and mainly describe the thresholds we will automatically determine.

The basic idea of the testing algorithm we used here is based on the observation that text regions typically are rich of corners and edges and corners and edge points are nearly uniformly distributed in text areas. There are four features we used in this approach: corner density, edge density, the ratio of vertical edge density and horizontal edge density, and center offset ratio of edges [16].

Fig. 6 shows the flow chart of the above text detection algorithm, and Fig. 7 shows the intermediate result of each step for the example video frame. In this approach, first we generate corner map of the video frame by a SUSAN corner detector [17] and obtain edge maps (vertical, horizontal and overall) with a Sobel edge detector. Then, candidate text regions are formed by corner refinement, merging, and dilation. The regions are then decomposed into single text lines using vertical and horizontal edge maps of the video frames. Next, a textbox verification step based on a set of features derived from edge maps is taken to significantly reduce false alarms. Finally, the text tracking method proposed in [18] is adopted to determine the time coverage of the textboxes.

This scheme efficiently utilizes the corner and edge distribution differences between text areas and nontext areas. In comparison with other video text detection approaches, this algorithm can obtain a more accurate text bounding box with a high recall rate and a rather low false alarm rate. The reason for these advantages is that we sufficiently utilize the distribution features of the corners and edge points. However, there are a few thresholds need to be determined to get the best results. As we all know, traditionally, we usually tune the thresholds of an algorithm manually by hand on several test images and evaluate its performance only by human vision such that it can produce the best performance, such as detection accuracy. However, it is tedious work and it is hard to test all possibilities. Now we will automate this process/task by developing the abovementioned PE protocol for video text detection.

In this section, first we will introduce the thresholds in region decomposition procedure of the testing algorithm, and then we will use our PE protocol to compare the detection results under different sets of thresholds and find the best one for the algorithms.

*1) Thresholds for Region Decomposition:* Region decomposition is performed after we obtain the candidate text areas by corner refinement, merging, and dilation [16]. To extract single text lines from candidate text areas, a vertical and a horizontal decomposition procedure using Sobel edge maps are performed. The Sobel vertical edge detector responses are scaled to [0, 255] and the edge threshold is $T_{\mathrm{edge}}$ (i.e., those responses greater than $T_{\mathrm{edge}}$ are regarded as edge points. The value is about 65 in our experiments). We can get one or more line segments, which are denoted as $l_i$, $i = 1, 2, \ldots$, when one horizontal scan line crosses a candidate text area, as shown in Fig. 8(a).

Denote the number of vertical edge points in the top and bottom $r$ lines of $l_i$ (including $l_i$) as $ETop_v(l_i, r)$ and $EBtm_v(l_i, r)$, respectively, as shown in Fig. 8(b). The length of $l_i$ is denoted as $|l_i|$. We define the number $(EN_v(l_i, r))$ and the density $(ED_v(l_i, r))$ of the edge points in the $r \times l_i$ scan area as follows:

$$EN_v(l_i, r) = \max\left(ETop_v(l_i, r), EBtm_v(l_i, r)\right) \tag{16}$$

$$ED_v(l_i, r) = \frac{EN_v(l_i, r)}{(|l_i| \times r)}. \tag{17}$$

If the line segment $l_i$ does not satisfy one or more of the following constraints, it will be deleted from the candidate text area:

$$EN_v(l_i, r_v) \geq min\_vedge\_number \tag{18}$$

$$ED_v(l_i, r_v) \geq min\_vedge\_density \tag{19}$$

where $r_v$ is about 3, $min\_vedge\_number$ is about 15, and $min\_vedge\_density$ is about 0.1.

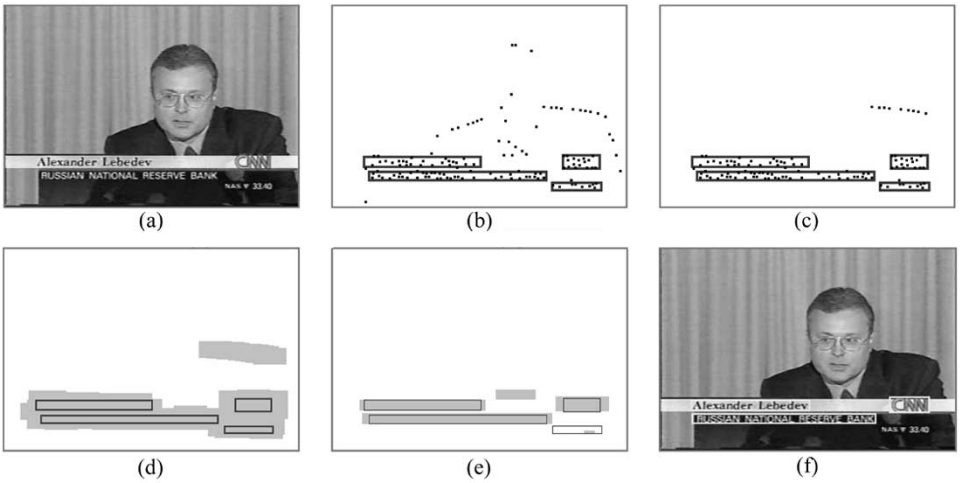


Fig. 7. Intermediate result of each step for the example video frame. (a) Example frame excerpted from CNN news. (b) Corner map. The blue boxes are the real textboxes. (c) Refined corner map. (d) Merged and dilated text area, i.e., candidate text areas. (e) After region decomposition. (f) The final result.

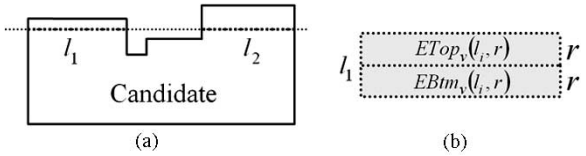


Fig. 8. Candidate text area decomposition.

For horizontal decomposition, it is the same as vertical decomposition, except that we use a horizontal edge map and vertical scan lines. That is, the vertical line segment $l_i$ will be deleted if it does not satisfy one or more of the following conditions:

$$EN_h(l_i, r_h) \geq min\_hedge\_number \tag{20}$$

$$ED_h(l_i, r_h) \geq min\_hedge\_density \tag{21}$$

where $r_h$ is about 8, $min\_hedge\_number$ is about 5, and $min\_hedge\_density$ is about 0.05 in our experiments.

After several iterations of decomposition vertically and horizontally, each area is then expanded to its bounding boxes. These bounding boxes are verified by a set of features derived from edge maps to reduce false alarm and thus we get the final results of video text detection which can be sent to the OCR engine for recognition after binarization.

*2) Threshold Determination by the Proposed PE Protocol:* In the previous section, we mentioned seven thresholds in the region decomposition procedure, which are summarized in Table II. The experiential values are obtained by experiments and the best thresholds need to be tuned all together. The last column is the tuning step in our scheme.

### B. Experiments

*1) Best Parameter Determination Using the Proposed PE Protocol:* We use the proposed PE scheme to determine seven parameters in the decomposition procedure of the above algorithm. The interface of the PE program developed by us is shown in Fig. 9. Each parameter has three value choices. Thus, there are $3^7 = 2187$ different cases of the threshold set. The testing data are the 45 video clips excerpted from MPEG-7 Content Set,

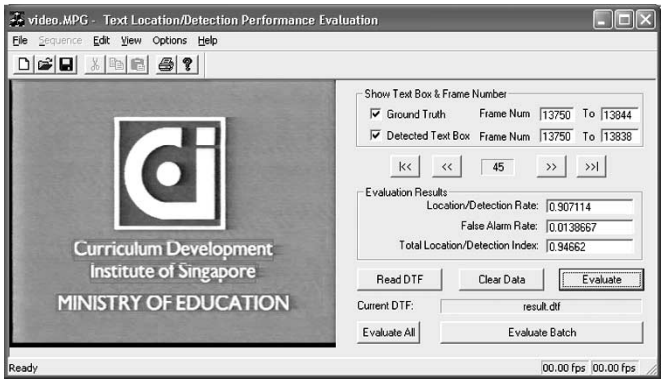


Fig. 9. Interface of the PE program.

which are mentioned in Sections II and III. Some evaluation results (using $\beta = 0.5$) including the one that can yield the best detection results (case 4) are listed in Table III. The corresponding parameter settings are listed in Table IV.

Using the parameters that can get the best detection results, we get the detection results of Fig. 3(a)–(c) and show them in Fig. 10. Parts of the performance indices on these test data are listed in Table V. From Table III and Table V, we also can see that, for the acceptable detection results (e.g., the detection rate is greater than 0.5), using (14) or (15) does not differ much.

*2) Impact of Using Different Edge Thresholds to the Detection Results:* Fig. 11 shows the PE curves of changing the edge threshold from 25 to 125 while using the best values for other thresholds. Fig. 12 shows the impact of using different edge thresholds to the detection results of an example frame.

*3) Stability of the Algorithm:* In addition, we test the text detection algorithm using the best parameters determined by our proposed PE scheme on 90 video clips from CNN news videos. The ground truths are also semi-automatically collected by the Ground Truth Generator mentioned in Section II. Of the 36 615 textboxes, 34 251 are detected correctly, and 1032 nontext areas are misdetected as textboxes. The detection rate, false alarm rate, and TDI are 0.857, 0.081, and 0.888, respectively, which are also the best results among all six threshold sets. The detection and performance evaluation results are also

TABLE II
THRESHOLDS TO BE DETERMINED BY THE PE PROTOCOL

| Threshold | Description | Experiential Value | Tuning Step |
|---|---|---|---|
| $T_{edge}$ | Edge threshold | 65 | ± 5 |
| $r_v$ | The radius of the vertical neighborhood (VN) | 3 | ± 1 |
| min_vedge_number | Minimum # of vertical edge points in VN | 15 | ± 1 |
| min_vedge_density | Minimum vertical edge points density in VN | 0.1 | ± 0.01 |
| $r_h$ | The radius of the horizontal neighborhood (HN) | 8 | ± 1 |
| min_hedge_number | Minimum # of horizontal edge points in HN | 5 | ± 1 |
| min_hedge_density | Minimum horizontal edge points density in HN | 0.05 | ± 0.01 |


(a) (b) (c)

Fig. 10. Detection results of Fig. 3(a)–(c).

TABLE III
SOME EVALUATION RESULTS FOR DIFFERENT CASES OF PARAMETER SETS ON THE GROUND TRUTH DATA

| Thresholds | Detected | | False Alarms | | Detection Rate | | False Alarm Rate | | *TDI* | | $TDI_G$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MPEG7 | CNN | MPEG7 | CNN | MPEG7 | CNN | MPEG7 | CNN | MPEG7 | CNN | MPEG7 | CNN |
| Case 1 | 11899 | 31898 | **275** | **618** | 0.712 | 0.761 | **0.047** | **0.079** | 0.833 | 0.841 | 0.824 | 0.837 |
| Case 2 | 14263 | 31213 | 546 | 1562 | 0.824 | 0.769 | 0.092 | 0.110 | 0.866 | 0.830 | 0.865 | 0.827 |
| Case 3 | 14574 | 33750 | *1432* | *3034* | 0.861 | **0.861** | *0.101* | *0.126* | 0.880 | 0.868 | 0.880 | 0.867 |
| Case 4 | **14698** | **34251** | 503 | 1032 | **0.872** | 0.857 | 0.072 | 0.081 | **0.900** | **0.888** | **0.900** | **0.887** |
| Case 5 | 12137 | *28567* | 516 | 1575 | *0.695* | *0.762* | 0.092 | 0.105 | *0.802* | *0.829* | 0.794 | 0.826 |
| Case 6 | *11566* | 30364 | 551 | 2561 | 0.721 | 0.773 | 0.074 | 0.083 | 0.824 | 0.845 | 0.817 | 0.842 |

**Note**: The numbers of true textboxes of MPEG-7 video and CNN video are 16 035 and 36615, respectively; $\beta = 0.5$.

TABLE IV
THRESHOLDS OF THE SIX CASES LISTED IN TABLE III

| Threshold | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 | Case 6 |
|---|---|---|---|---|---|---|
| $T_{edge}$ | 60 | 60 | 65 | 65 | 70 | 70 |
| $r_v$ | 2 | 2 | 3 | 3 | 4 | 4 |
| min_vedge_number | 14 | 15 | 16 | 14 | 15 | 16 |
| min_vedge_density | 0.09 | 0.09 | 0.10 | 0.10 | 0.11 | 0.11 |
| $r_h$ | 7 | 8 | 9 | 7 | 8 | 9 |
| min_hedge_number | 4 | 5 | 6 | 4 | 5 | 6 |
| min_hedge_density | 0.04 | 0.04 | 0.05 | 0.05 | 0.06 | 0.06 |

listed in Table III, from which we can also see that the evaluation values are rather close and have similar trends on the two testing sets (the average differences of detection rates, false alarm rates, TDI, and $TDI_G$ between the two testing sets are 0.040, 0.018, 0.020 and 0.022, respectively), which indicates that the PE method is much stable. That is to say, based on the proposed PE protocol, a given algorithm can maintain similar TDI and $TDI_G$ values for various sets of test data (the average $L_{DD}$ of CNN and MPEG7 test data set are 3.33 and 2.00, respectively).

*4) Performance Comparison of Different Video Detection Schemes Using the Proposed PE Protocol:* We also compare the algorithm (HUA) described in Section IV with the other three text detection schemes using our PE protocol on the abovementioned testing data. The first scheme (denoted by QI) for comparison is from [1], the second one (XI) is from [18], and the third one (XI-2) is an improved version of the second one, in which detection results in consecutive frames are used to enhance the final performance. The evaluation results of the four algorithms are listed in Table VI. It can be seen from the table that the algorithm HUA produce better detection results than the other three algorithms.

## V. SUMMARY

In this paper, we propose an objective, comprehensive, and difficulty-tolerant performance evaluation protocol for video text detection algorithms. The protocol is tolerant of the ground truth difficulty. The performance evaluation scheme has been applied to a video text detection approach to determine the best thresholds those can yield the best detection results, and

TABLE V
SOME PERFORMANCE INDICES FOR THE DETECTION RESULTS IN FIG. 10

| Fig. | No. | *STR* | $L_{DD}$ | *DI* | $Q(g)$ | *D* | *F* | *TDI* | $TDI_G$ |
|---|---|---|---|---|---|---|---|---|---|
| | a1 | outdoor recording | 4 | 32 | 0.9187 | | | | |
| | a2 | SOH CHIN SIONG | 2 | 38 | 0.9344 | | | | |
| 10 (a) | a3 | ABDUL MA JEED | 3 | 33 | 1.0000 | 0.9429 | 0.0826 | 0.9302 | 0.9301 |
| | a4 | SZE CHUN MIN | 2 | 30 | 0.9423 | | | | |
| | a5 | WILLIE WHY | 2 | 27 | 0.9534 | | | | |
| | b1 | Banco O??c?a? | 2 | 0 | 0.0000 | | | | |
| | b2 | SolBank | 3 | 21 | 0.8935 | | | | |
| 10 (b) | b3 | tve (*left*) | 7 | 6 | 0.9444 | 0.9573 | 0.0296 | 0.9639 | 0.9638 |
| | b4 | 2Kms. META | 4 | 18 | 0.9953 | | | | |
| | b5 | tve (*right*) | 6 | 6 | 0.9126 | | | | |
| | c1 | Curriculum Development | 4 | 63 | 0.9694 | | | | |
| 10 (c) | c2 | Institute of Singapore | 4 | 60 | 0.9336 | 0.9587 | 0.0256 | 0.9666 | 0.9665 |
| | c3 | MINISTRY OF EDUCATION | 3 | 57 | 0.9731 | | | | |

**Note**: $D$, $F$, TDI, and $TDI_G$ here are the PE results of all textboxes in the corresponding video frames.

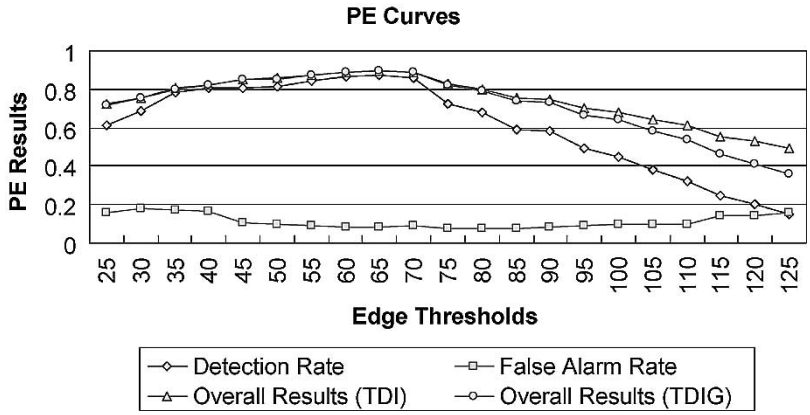


Fig. 11. PE curves of changing the edge threshold.

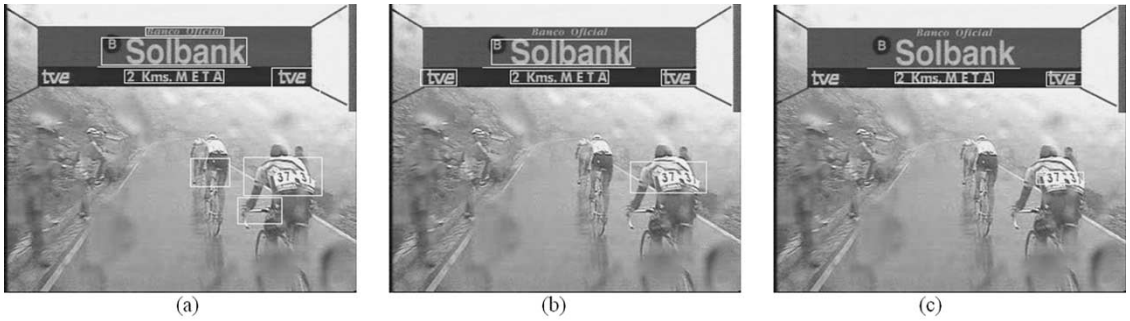


Fig. 12. Examples: impact of different edge thresholds to detection. Edge thresholds for (a)–(c) are 25, 45, and 85, respectively. The detection result when the edge threshold is 65 has been shown in Fig. 10(b).

TABLE VI
EVALUATION RESULTS OF FOUR TEXT DETECTION APPROACHES

| Algorithm | HUA | QI | XI | XI-2 |
|---|---|---|---|---|
| Total Clips | 90 | 90 | 90 | 90 |
| Total Textboxes | 36615 | 36615 | 36615 | 36615 |
| Total Missed Textboxes | 2364 | 6644 | 2650 | **1955** |
| Total False Alarms | **1032** | 1324 | 3063 | 1611 |
| Detection Rate | **0.857** | 0.599 | 0.671 | 0.831 |
| False Alarm Rate | **0.081** | 0.153 | 0.205 | 0.142 |
| Overall PE Results (*TDI*) | **0.888** | 0.723 | 0.733 | 0.845 |
| Overall PE Results ($TDI_G$) | **0.887** | 0.712 | 0.730 | 0.844 |

to compare the performance of several video text detection systems. The proposed protocol can not only be used to compare different video/image text detection algorithms/systems, but also can help improve, select, and even design new text detection methods (e.g., to determine whether or not a threshold is sensitive to noises and other environmental variables).

## REFERENCES

[1] W. Qi *et al.*, "Integrating visual, audio and text analysis for news video," in *Proc. Int. Conf. Image Processing (ICIP 2000)*, Vancouver, BC, Canada.

[2] Y. Zhong, H. J. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 385–392, Apr. 2000.
[3] L. Zhao *et al.*, "Video shot grouping using best-first model merging," in *Proc. 13th SPIE Symp. Electronic Imaging—Storage and Retrieval for Image and Video Databases*, San Jose, CA, Jan. 2001.
[4] A. K. Jain and B. Yu, "Automatic text location in images and video frames," *Pattern Recognit.*, vol. 31, no. 12, pp. 2055–2076, 1998.
[5] V. Wu, R. Manmatha, and E. M. Riseman, "Finding text in images," in *Proc. 20th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Philadelphia, PA, 1997, pp. 3–12.
[6] ——, "Finding text in images," in *Proc. 2nd ACM Int. Conf. Digital Libraries (DL'97)*, July 1997.
[7] H. P. Li and D. Doermann, "Automatic text detection and tracking in digital video," *IEEE Trans. Image Processing*, vol. 9, pp. 147–156, Jan. 2000.
[8] A. Wernicke and R. Lienhart, "On the segmentation of text in videos," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME 2000)*, New York, July 2000, pp. 1511–1514.
[9] I. T. Philips *et al.*, "A performance evaluation protocol for graphics recognition systems," in *Graphics Recognition—Algorithms and Systems*, K. Tombre and A. Chhabra, Eds. Berlin, Germany: Springer, 1998, vol. 1389, Lecture Notes in Computer Science, pp. 373–389.
[10] P. J. Phillips and K. W. Bowyer, "Introduction to the special section on empirical evaluation of computer vision algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 289–290, Apr. 1999.
[11] L. Wenyin and D. Dori, "A protocol for performance evaluation of line detection algorithms," *Machine Vis. Applicat.*, vol. 9, no. 5/6, pp. 240–250, 1997.
[12] ——, "A proposed scheme for performance evaluation of graphics/text separation algorithms," in *Graphics Recognition—Algorithms and Systems*, K. Tombre and A. Chhabra, Eds. Berlin, Germany: Springer, 1998, vol. 1389, Lecture Notes in Computer Science, pp. 359–371.
[13] Benchmarking and Performance Evaluation, ECVNet. [Online]. Available: http://www-prima.inrialpes.fr/ECVNet/ benchmarking.html
[14] L. Lam and C. Y. Suen, "Evaluation of thinning algorithms from an OCR viewpoint," in *Proc. 2nd Int. Conf. Document Analysis and Recognition*, Tsukuba, Japan, 1993, pp. 287–290.
[15] L. Wenyin and D. Dori, "Principles of constructing a performance evaluation protocol for graphics recognition algorithms," in *Performance Characterization and Evaluation of Computer Vision Algorithms*, R. Klette, S. Stiehl, and M. Viergever, Eds. Boston, MA: Kluwer, 1999, pp. 97–106.
[16] X.-S. Hua, X.-R. Chen, L. Wenyin, and H.-J. Zhang, "Automatic location of text in video frames," in *Proc. ACM Multimedia 2001 Workshops: Multimedia Information Retrieval (MIR2001)*, Ottawa, ON, Canada, Oct. 5, 2001, pp. 24–27.
[17] S. M. Smith and J. M. Brady, "SUSAN—a new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, May 1997.
[18] J. Xi, X.-S. Hua, X.-R. Chen, L. Wenyin, and H.-J. Zhang, "A video text detection and recognition system," in *Proc. 2001 IEEE Int. Conf. Multimedia and Expo (ICME2001)*, Tokyo, Japan, Aug. 22–25, 2001, pp. 1080–1083.
[19] R. Lienhart and A. Wernicked, "Localizing and segmenting text in images and videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 236–268, Apr. 2002.
[20] V. Wu, R. Manmatha, and E. M. Riseman, "Textfinder: an automatic system to detect and recognize text in images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 1224–1229, Nov. 1999.
[21] R. Lienhart and W. Effelsberg, "Automatic text segmentation and text recognition for video indexing," *ACM/Springer Multimedia Syst.*, vol. 8, pp. 69–81, Jan. 2000.

**Xian-Sheng Hua** received the B.S. and Ph.D. degrees from Beijing University, Beijing, China, in 1996 and 2001, respectively, both in applied mathematics.

Since 2001, he has been with Microsoft Research Asia, Beijing, where he is currently an Associate Researcher with the media computing group. His current interests are in the areas of pattern recognition, content-based video analysis, and image analysis. He has authored more than 20 publications in these areas and has five patents or pending applications.

Dr. Hua is a member of the Association for Computing Machinery.

**Liu Wenyin** (M'99–SM'02) received the B.Eng. and M.Eng. degrees in computer science from the Tsinghua University, Beijing, China, in 1988 and 1992, respectively, and the D.Sc. degree in information management engineering from the Technion, Israel Institute of Technology, Haifa, in 1998.

He was with Tsinghua University as a Faculty Member for three years and with Microsoft Research China/Asia, Beijing, as a full-time Researcher for another three years. Currently, he is an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His research interests include graphics recognition, sketch recognition and pen-based user interface, pattern recognition and performance evaluation, user modeling and personalization, multimedia information retrieval, personalization information management, object-process methodology, and software engineering and Web services engineering. He has authored or coauthored more than 100 publications and has nine patents pending in these areas.

Dr. Wenyin is a member of the Association for Computing Machinery (ACM), the IEEE Computer Society, and the IEEE Education Society. He played a major role in developing the Machine Drawing Understanding System (MDUS), which won First Place in the Dashed Line Recognition Contest held during the First IAPR Workshop on Graphics Recognition at Pennsylvania State University in 1995. In 1997, he won a Third Prize in the ACM/IBM First International Java Programming Contest (ACM Quest for Java'97) (http://www.acm.org/jquest/webquest1.html). In 2003, he was the recipient of the ICDAR Outstanding Young Researcher Award by the International Association for Pattern Recognition (IAPR) for his significant impact in the research domain of graphics recognition, engineering drawings recognition, and performance evaluation. He co-chaired the Fourth International Contest on Arc Segmentation held during the fourth IAPR Workshop on Graphics Recognition, Kingston, ON, Canada, in September 2001, and chaired the fifth International Contest on Arc Segmentation held during the Fifth IAPR Workshop on Graphics Recognition, Barcelona, Spain, in July 2003. He is currently organizing the Sixth IAPR Workshop on Graphics Recognition, Hong Kong, in August 2005.

**Hong-Jiang Zhang** (S'90–M'91–SM'97–F'04) received the B.S. degree from Zhengzhou University, Zhengzhou, China, in 1982 and the Ph.D. degree from the Technical University of Denmark, Copenhagen, in 1991, both in electrical engineering.

From 1992 to 1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. He also worked at MIT Media Lab in 1994 as a Visiting Researcher. From 1995 to 1999, he was a Research Manager with Hewlett-Packard Labs, where he was responsible for research and technology transfers in the areas of multimedia management; intelligent image processing and Internet media. In 1999, he joined Microsoft Research Asia, Beijing, China, where he is currently a Managing Director in charge of the Advanced Technology Center.

Dr. Zhang is a member of the Association for Computing Machinery. He has authored three books, over 200 referred papers and book chapters, seven special issues of international journals on image and video processing, content-based media retrieval, and computer vision, as well as numerous patents or pending applications. He currently serves on the editorial boards of five IEEE/ACM journals and a dozen committees of international conferences.