International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 8 (2016) pp. 033-041

An Integrated Approach for Multilingual Scene Text Detection

Wen-Hung Liao[1], Yi-Chieh Wu

[1] Department of Computer Science, National Chengchi University,
No.64, Sec.2, ZhiNan Rd., Wenshan District, Taipei City 11605, Taiwan
whliao@gmail.com

***Abstract*: Text messages in an image usually contain useful information related to the scene, such as location, name, direction or warning. As such, robust and efficient scene text detection has gained increasing attention in the area of computer vision recently. However, most existing scene text detection methods are devised to process Latin-based languages. For the few researches that reported the investigation of Chinese text, the detection rate was inferior to the result for English. In this research, we propose a multilingual scene text detection algorithm for both Chinese and English. The method comprises of four stages: 1. Preprocessing by bilateral filter to make the text region more stable. 2. Extracting candidate text edge and region using Canny edge detector and Maximally Stable Extremal Region (MSER) respectively. Then combine these two features to achieve more robust results. 3. Linking candidate characters: considering both horizontal and vertical direction, character candidates are clustered into text candidates using geometrical constraints. 4. Classifying candidate texts using support vector machine (SVM), to separate text and non-text areas. Experimental results show that the proposed method detects both Chinese and English texts, and achieve satisfactory performance compared to those approaches designed only for English detection.**

***Keywords*:** Multilingual scene text detection, Maximally stable extremal region (MSER), Support vector machine (SVM)**.**

I. Introduction

Unlike computer-generated text files in which the characters follow consistent writing direction, format and presentation, texts in real world are fragments of information that are free-style and usually consist of multiple languages. Therefore, it is more challenging to detect and recognize scene text than printed documents. Fig. 1 illustrates the difference by comparing computer-generated text with texts that appear in a scene.

Various approaches have been proposed to detect texts in a captured image recently. The majority of existing methods focused on text and non-text classification for a single language (mainly English). The International Conference on Document Analysis and Recognition (ICDAR) conducts a series of robust reading competition every two years [1]. One big challenge of these competitions is text detection and localization from the scene. Fig. 2 illustrates some potential applications using scene text detection and recognition technology, including image search, navigation assistance and assistance to visually-impaired people.

(a) (b)

Figure 1. Example of different text image sets: (a) text from a book (b) text in a scene

Figure 2. Scene text detection and recognition applications

However, scene text challenges of ICDAR competition are English only until 2015 [1]. Even in the latest dataset, languages other than English in the scene constitute less than 6% of the sample. There exist fewer researches about Chinese scene text detection and recognition in scene images in the past compared with English. Multilingual text detection poses more challenges as the overall structure, individual stroke and character direction may differ. Mixture of letters of distinct characteristics calls for more efforts in designing rules to extract regions containing text. Successful extraction of text areas contributes positively to the subsequent processing stage: character recognition. If the captured information is not complete, the recognition performance will be limited. Therefore, we wish to improve the scene text detection results for multilingual content to boost the accuracy of text recognition in this research.

We propose an integrated scene text detection system to

detect textual regions from scene images without restriction to a single language. In particular, we wish to retrieve more complete text information in complex scenes containing both Chinese and English, as shown in Fig. 3. Toward this objective, we have developed a framework that effectively combines textual image features at different stages of processing to level up the overall accuracy. The basic idea is to start with simple rules to eliminate least possible regions without removing potential text area prematurely. Textual features for different languages will then be considered and carefully combined to further reduce the number of candidate regions. Finally, a classifier is employed to verify whether the candidate region actually contains text information.

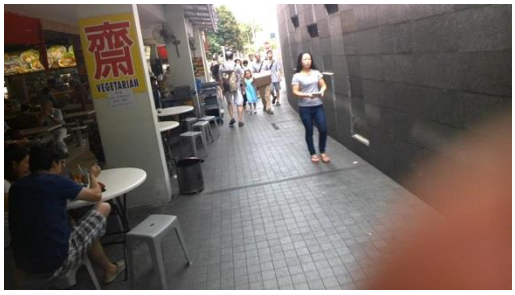


Figure 3. A multilingual scene text example (from ICDAR 2015)

The remainder of this paper is organized as follows. In Section 2 we outline a few related work and systems in scene text detection. Section 3 presents a multilingual scene text detection algorithm for both Chinese and English. It includes four stages: preprocessing, extracting character candidates, constructing text candidates and support vector machine (SVM) classification of text and non-text areas. In Section 4, we conduct comparative analysis and demonstrate experimental results of detecting both Chinese and English texts. Section 5 concludes this paper with a brief conclusion and possible strategies for future improvements.

II. Related Work

More and more applications provide services to retrieve rich information in scene text in recent years. Google Translate is a multiple language translation App on both iOS and Android platform. It is capable of translating texts in real-time from live video streams. In the latest version, its performance is further enhanced using deep convolutional neural networks [2]. However, this service relies on cloud database and network, thus users cannot operate the recognition function offline currently. Fig. 4 shows the result of Google Translate using Fig. 3 as input image. We can spot some erroneous detection in non-text area (at the top-right corner) when translating Chinese to English.

In OpenCV 3.0 library, there are also modules released specifically for scene text detection and recognition [3]. According to the documentation, the scene text detection algorithm implemented in this module has been initially proposed by Neumann and Matas [4]. Two OCR approaches are available for recognition: tesseract-ocr and HMM-ocr. However, this module handles English only.

Existing methods for scene text detection can be divided into three broad categories: sliding-window-based, connected-component-based, and hybrid approach.

(a) (b)

Figure 4. Test results using Google Translate, (a) Translating English to Chinese (b) Translating Chinese to English

A. Sliding window based method

Zhou *et al.* [6] summed up two common properties according to the writing system using Chinese, English and Arabic as detection example of multi-languages:

- Writing direction is often horizontally oriented.
- Characters are composed of strokes.

Based on the above principles, three texture descriptors were selected to represent multilingual features:

1. Histogram of oriented gradient (HoG)
2. Mean of gradient (MG)
3. Local binary patterns (LBP)

The authors then adopted a tandem of AdaBoost classifiers to combine different features to train and classify text and non-text of originating from multiple languages. However, this method can only handle horizontal texts. It cannot be applied to extract Chinese due to various combinations of writing directions. Moreover, since different features must be trained for different languages respectively, the computation is rather complex.

B. Connected component based method

Epshtein *et al.* proposed the stroke width transform (SWT) algorithm in [7]. This method used Canny [8] edge detection and computed the stroke width of each pixel and combined a connected component with similar stroke width to form character candidates. These character candidates are linked together as a text string according to certain heuristics. However, the performance of this method depends heavily on the quality of edge detection. Yao *et al.* [9] improved this method by making it possible to detect text string in any direction using machine learning approach.

Chen *et al.* [10] used Canny edges to complement Maximally Stable Extremal Region (MSER) [11] to distinguish character candidates and background by the gradient direction of edge pixels. Moreover, it improved the stroke width transform by using the central skeleton as a reference to compute text stroke width. Since it utilizes MSER to detect character candidates, errors will occur when input images are blurred or of low resolution as MSER results are less reliable under these circumstances.

C. Hybrid method

Pan *et al.* [12] created text confidence map on a series of different scaling of grayscale images (image pyramid) to represent the possibility of text of an area using the Waldboost classifier trained by histogram of gradient (HoG) features. They adopted Niblack's algorithm [13] to convert grayscale image into binary image, and use the text

confidence as a feature with conditional random field (CRF) to determine whether the candidate area contains text or not. In the combined text string stage, they applied minimum spanning tree (MST) to connect the same line of text. Because this method uses image pyramid and CRF, the computation is quite demanding and its accuracy is still questionable.

III. The Proposed Methodology

In the section, we will introduce the proposed scene text detection system. It consists of four stages as shown in Fig. 5:

A) Preprocessing: To retrieve the complete character information in images, we adopt these filters in this step:

a) Bilateral filter: To smooth images.

b) High-pass filter: To sharpen images.

c) Median filter: To filter out noise from images.

B) Extracting character candidates: In order to refine the character candidate information in images, we use Canny detector and MSER to retrieve the edges and local features of characters, followed by a repairing process.

C) Constructing text candidates: We choose the character candidates according to structural features such as stroke width and character aspect ratio. We then link character candidates to text candidates according to shape and geometric constraints.

D) SVM classifier: We determine if the text candidates actually contain a text string using a SVM classifier.

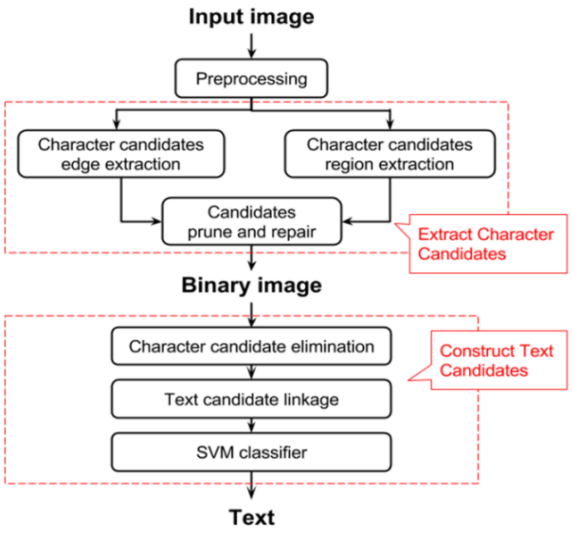


Figure 5. Multilingual scene text detection system

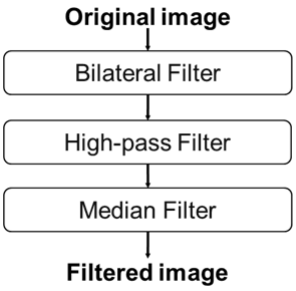


Figure 6. Preprocessing procedure

A. *Preprocessing*

In this stage, we use three different filters (smoothing, sharpening, and denoising) to preprocess images in order to obtain a picture with more stable text regions. The procedure is shown in Fig. 6.

Firstly, we use bilateral filter to reduce noise and retain the edge features in an original image. With this operation, we can get more complete region without losing the edge details of a character.

Secondly, we use high-pass filter with 3x3 sharpening mask to strengthen the high frequency component, increase the contrast of the background and text, and enhance the edge of an image.

At last, in order not to emphasize the noise after sharpening, we adopt median filter to reduce noise in an image, and retain the shape and position of the edge. An example of the preprocessing stage is depicted in Fig. 7.

B. *Extracting character candidates*

In order to retrieve image regions that contain characters, we compile the following list of properties:

- The main feature of a character is a collection of strokes.
- A stroke has significant edge features.
- These edges form multiple cycles.
- The area of a stroke contains stable gradient.
- Characters in the same text are usually of the same color.

According to the above heuristics, we adopt Canny edge detector and Maximally Stable Extremal Region (MSER) detection to extract character candidates in images. The overall procedure is illustrated in Fig. 8.

Figure 7. An example of flow of preprocessing. (a) Original image. (b) Image processed by bilateral filter. (c) Image sharpened after (b). (d) Image smoothed after (c).

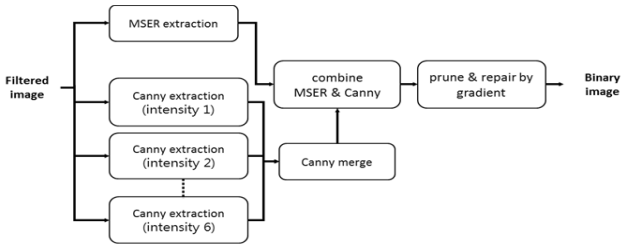


Figure 8. Extracting character candidates

(a)

(b)

Figure 9. Canny edge extraction. (a) Six different intensity simulations of an image with corresponding Canny edge detection. The number denotes the parameter used for intensity modification. (b) The merged result.

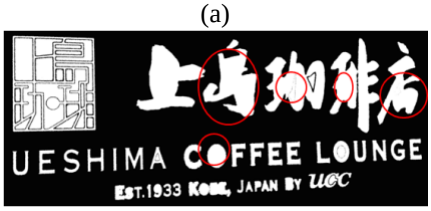
Because the edge maps obtained from Canny operator are affected by the intensity of the input images, we will simulate different lighting conditions to generate several images from the same source. We will then apply Canny operator to these images and obtain the corresponding edge maps. Later, we will merge these edge detection results in order to retrieve more complete edges information of a character. Fig. 9(a) illustrates the six simulations of different intensities of an input image and individual Canny detection result. Fig. 9(b) shows the merged Canny edge map which contain most complete edge information.

The main concept of MSER is defined as the extremal value of intensity in a region. In most images, the local binarization is stable in certain regions over a large range of threshold values, and the property of MSER is suitable for detecting text area, because characters in the same text are usually of the same color. However, we have noticed that the performance of MSER detection is usually inferior for low resolution and blurred images. This issue is demonstrated in Fig. 10. We found low performance of MSER will occur when the environment is too light or dark, or image is blurred or of low resolution. It causes the detection of the edge area of character candidates to be unstable. For example, the detected stroke areas can be incomplete, or background region between strokes can be wrongly identified as the character area. These factors lead to data loss and cause character candidates to be classified as non-text at later stages.

We then merge these edge detection results to yield a more stable outcome. To be exact, we have adopted merged Canny maps to gain better accuracy on edge area, as described above. Since we already have the contrast between text area and background in MSER, we can determine the gradient direction of the edge of character, and discard the background recognized as character according to the gradient direction. An example is presented in Fig. 11.

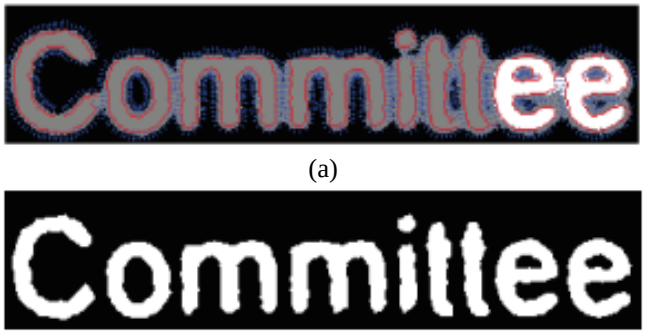
However, some area in image cannot be detected by Canny detector, but the same area can be detected successfully by MSER detection. Under this circumstance, we adopt the MSER detection result as character candidates directly. Fig. 12 illustrates the issue discussed here. For all other cases, we merge the outcome from these two detection methods to recover lost data. The integrated result is shown in Fig. 13.

(a)



(b)

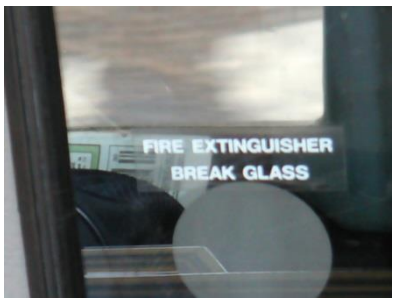
Figure 10. MSER detection. (a) Original image (b) MSER detection result. The areas circled in red show incomplete and ambiguous results.



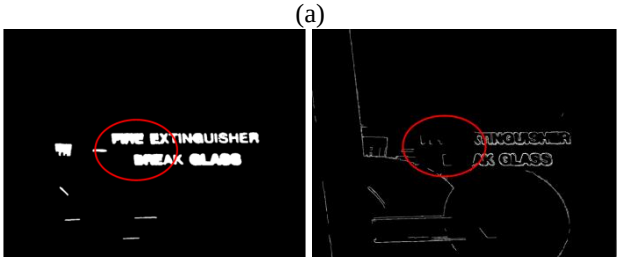
(a)

(b)

Figure 11. Canny detection. (a) The red lines depict Canny edges, and blue arrows indicate the gradient direction. (b) Character candidate areas obtained by combining edge information



(a)



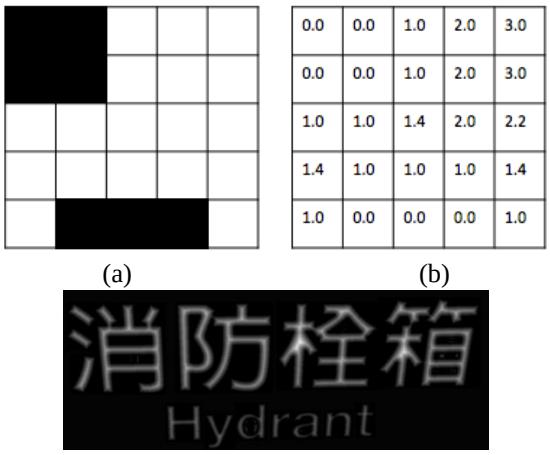
(b) (c)

Figure 12. No merging when MSER and Canny generate conflicting results. (a) Original image (b) MSER detection (c) Canny detection.

(a)

(b)

Figure 13. (a) MSER detection result (b) Merging MSER and Canny detection results.



(a) (b)

(c)

Figure 14. Distance Transform computation. (a) Binarized image, with foreground colored in white. (b) Distance transform computed for each pixel (c) Distance transform shown as an image.

C. Constructing text candidates

To construct text candidates, we should determine whether a character candidate is indeed a character or not in the first step because we do not want to include noise or background pixels to interfere our results of character candidates. Therefore, we adopt a series of decision process whose criteria become increasingly stricter, similar to a process funnel. After that, we can merge these characters into text candidates.

1) Character candidate elimination

Variance of stroke width of the same character is usually low. Geometric shape of characters can also be used as a restriction to filter out background and noise. Here we use two rules to filter out non-characters regions, namely, stroke width and aspect ratio respectively.

a) Stroke width: We compute Euclidean distance of Distance Transform (DT) between the foreground object and its boundary to obtain stroke width. Figs. 14(a)(b) indicate that the pixel value of background is set to 0, and pixel value of the adjacent of foreground with background is set to 1. The intensity of other points is set to the shortest adjacent distance. Fig. 14(c) shows the result of distance transform. The maximum in these areas are known as the skeletal points. Its value is equal to half the stroke width. Therefore, from the skeleton point along the downhill direction, we assign the value of the skeletal point to each pixel until we reach the border.

The output of the DT algorithm is an image. The pixel value of foreground object is half the stroke width. We determine whether a pixel belongs to the character area or background noise based on the standard deviation of the character candidates. If the standard deviation of stroke width of the character candidates is large, it is less likely to be a true character. The exclusion criterion is defined in (1):

$$\frac{std_{sw}}{mean_{sw}} < 0.5 \tag{1}$$

where std_{sw} is the standard deviation of foreground pixels in the procedure, and $mean_{sw}$ represents the mean of the extracted foreground pixels at this stage.

b) Stroke width: We observed the shapes of characters and found that most characters are formed by roughly rectangular connected components as depicted Fig. 15. But in Chinese and English, "i", "I", "l", "j" and "J" are relatively slim texts, with structures similar to railing or cable in a scene. Eq. (2) is employed to remove the background noises while retain the slim text at the same time.

$$aspect\ ratio > 15\ \&\&\ aspect\ ratio < \frac{1}{15} \tag{2}$$

Fig. 16 displays the result of character candidate extraction. We found that some parts of background or noise are still included because they have stable width and the aspect ratio is similar to that of characters. These regions will be eliminated at a later processing stage.

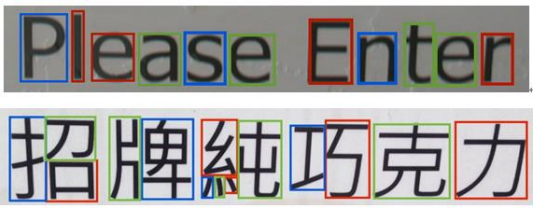
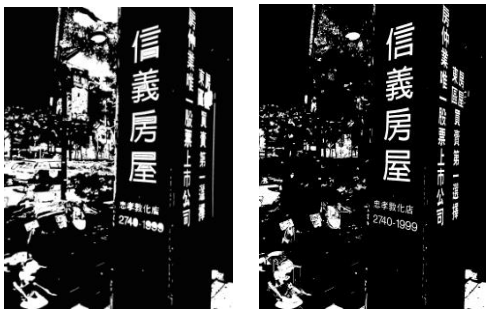


Figure 15. Characters marked by merging connected components.



(a) (b)

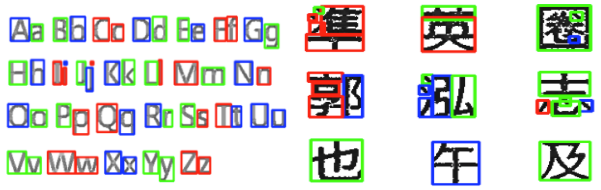
Figure 16. Character candidate elimination (a) Before processing (b) After processing

2) Text candidate linkage

Generally, English is written horizontally, and the strokes in a character are connected with each other. Only "i" and "j" are divided into two parts in the alphabet. In Chinese, writing direction can either be horizontal or vertical. Strokes in a Chinese character may be connected or not. Moreover, the composition can be classified into many different forms. Thus,

an English character includes a single component (except for i, j). But a Chinese character includes multiple components, and different characters have different combinations of structure, as shown in Fig. 17.

Since we need to handle both English and Chinese, the assumption of horizontal composition does not apply. Considering the mixed structure of horizontal and vertical arrangement in Chinese, we have identified several rules for text linkage. As illustrated in Fig. 18, assume p and q denote two character components respectively. Let x_p and y_p be the upper left corner position of rectangle p. $Height_p$ and $width_p$ are the height and width of rectangle p. Sw_p is the mean of stroke width of rectangle p. $Gray_p$ is the mean of grayscale value of character component p.



(a) (b)

Figure 17. Comparison of connected components. (a) English (b) Chinese

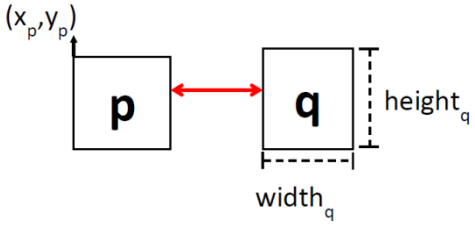


Figure 18. Diagram showing the relative position of two character components. Double arrow indicates the distance between these two components.

(a) (b) (c) (d) (e)

Figure 19. Examples illustrating the criteria for text candidate linkage.(a) Source images (b) Spacing criteria (c) Difference of width and height (d) Difference of mean of grayscale (e) Difference of mean of stroke width

In order to compose them into text candidates, we compute the following properties for character components. These criteria describe the relationship of two candidate component, and if the condition is matched, we merge these components into one:

- Spacing criteria

$$abs(x_q - x_p - width_p) < 1.5 \times max(width_p, width_q) \quad (3)$$

$$abs(x_q - x_p - width_p) < 1.5 \times max(width_p, width_q) \quad (4)$$

- Difference of width and height

$$min(width_p, width_q) > 0.2 \times max(width_p, width_q) \quad (5)$$

$$min(height_p, height_q) > 0.2 \times max(height_p, height_q) \quad (6)$$

- Difference of mean of grayscale

$$abs(gray_p - gray_q) < \frac{1}{8} \times 255 \quad (7)$$

- Difference of mean of stroke width

$$abs(sw_p - sw_q) < \frac{1}{3} \times max(sw_p, sw_q) \quad (8)$$

Examples illustrating each criterion are provided in Fig. 19. Fig. 20 shows an example of composing character candidates into a text candidate using these criteria. It is observed that most English text strings can be detected using the above method. However, if the parts (points or strokes) are smaller than those of the Chinese character, they will be ignored in the step of text candidate linkage. To address this issue, we exclude the size factor and just check if there is some overlap between the two components to determine whether these character candidates belong to text or not according to the following rules:

- To check if p and q intersects

$$p \cap q \neq null \quad (9)$$

- Difference of mean of grayscale: The formula is the same as Eq. (7).
- Difference of mean of stroke width

$$abs(sw_p - sw_q) < \frac{1}{2} \times max(sw_p, sw_q) \quad (10)$$

An example illustrating these merging criteria is given in Fig. 21. Fig. 22 demonstrates the results of combining overlapping text regions.

(a)

(b)

Figure 20. Character components identified as a text candidate: (a) English (b) Chinese.

Figure 21. An example showing the case of combining overlapping character components.

(a) (b) (c)

Figure 22. Combining overlapping texts (a) Adjacent text linkage (b) Result after linking overlapping text (c) Result of overlapping text linkage for tilted text

Figure 23. Results after constructing text candidates. There are still non-text areas misclassified as text candidates.

D. SVM classifier

Since source of scene images are not strictly limited, we cannot determine what is text or not by a single criterion. There can still be noises after text candidate linkage, as depicted in Fig. 23. We adopted support vector machine (SVM) [14] to build models to distinguish text string from non-text area. We selected the following features to form the input vector to the SVM classifier:

- Standard deviation of stroke width: I_{DM} is the image of stroke width, as known as distance map. N is the pixel number of character.

$$std_{sw} = \sqrt{\sum_{\substack{\forall 0<i<height,\\ 0<j<width}} (I_{DM}(i,j) - mean_{sw})^2} \ / \ mean_{sw}$$

where

$$mean_{sw} = \sum_{\substack{\forall 0<i<height,\\ 0<j<width}} I_{DM}(i,j) \ / N, \ \forall I_{DM}(i,j) > 0 \quad (11)$$

- Image size ratio: $area_{original}$ is the size of original image, and $area_{candidate}$ is the size of character candidate.

$$image\ size\ ratio = area_{candidate} \ / \ area_{original} \quad (12)$$

- Stroke width smoothness: $\theta_{i,j}$ is the gradient direction of *pixel(i,j)*, and N_{edge} is the number of edge pixels.

$$smooth = \left(\sum_{\substack{\forall 0<j<height\\ 0<i<width}} \left(\sum_{\substack{\forall j-1<n<j+1\\ i-1<m<i+1}} \theta_{i,j} - \theta_{m,n} \right) \right) / N_{edge}$$
$$\forall (i,j) and (m,n) \in edge's\ pixel \quad (13)$$

- Mean of gradient: Gx_i and Gy_j are the differential in the horizontal and vertical directions of gradient of *pixel(i,j)* in the image of stroke width respectively. N is the pixel number of character.

$$MG = \sum_{\substack{\forall 0<j<height\\ 0<i<width}} \sqrt{Gx_i^2 + Gy_j^2} \ / \ N, \ \forall I_{DM}(x,y) > 0 \quad (14)$$

- Standard deviation of edge's gradient: N_{edge} is the pixel number of edge.

$$std_{gradient} = \sqrt{\sum_{\substack{\forall 0<j<height\\ 0<i<width}} \left(\sqrt{Gx_i^2 + Gy_j^2} - mean_{gradient} \right)^2} \ / mean_{gradient}$$

where

$$mean_{gradient} = \sum_{\substack{\forall 0<j<height\\ 0<i<width}} \sqrt{Gx_i^2 + Gy_j^2} \ / N_{edge} \quad (15)$$
$$\forall (i,j) \in edge's\ pixel$$

In the training dataset, we collect 600 images as training samples (229 are collected from ICDAR 2011 database, 200 are collected from Multilingual Scene Text image set. The remaining images contain no text, which are used as negative training samples). Fig. 24 depicts some samples from the training set.

Figure 24. Training samples from Multilingual Scene Text image set.

IV. Result and Discussion

We employ two different datasets in order to test the efficacy of the proposed scene text detection scheme. The first dataset is ICDAR 2011 Robust Reading Competition database, which is the benchmark for all scene image text detection methods. The target language is English. The second dataset is our collection for multilingual scene text images (Chinese and English). We evaluate the performance for our approach by measuring English with a word as a unit and Chinese with a character as a unit, based on precision, recall, and F-measure. We compute the three evaluation values by the following equations, respectively.

$$Precision = True\ Positive/(True\ Positive + False\ Positive) \quad (16)$$

$$Recall = True\ Positive/(True\ Positive + False\ Negative) \quad (17)$$

$$F\text{-}measure = 2 \times Precision \times Recall/(Precision + Recall) \quad (18)$$

A. ICDAR 2011 Database

The dataset we choose is Challenge 2 of ICDAR 2011 Robust Reading Competition. There are 229 training images and 255 testing images included in the dataset. Texts in images are English and most of them are typographical, horizontal and no tilt.

We compare our approach with existing methods in Table 1. These detection methods are designed for English only. They are more suitable for horizontal writing style. Our proposed framework is more flexible, and achieves comparable performance when applied to this dataset.

The key factor affecting the computation time of our approach is the number of detected candidate characters. If the portion of text and noise in the image is large, it will take more time in the stage of text candidate linkage. The time complexity is $O(n)+O(n^2)$ where n is the number of text candidates.

Table 1. ICDAR 2011 Database Detection Methods Comparison

Method	Criteria For Assessment		
	Precision	**Recall**	**F-measure**
Neumann's [15]	85.4%	67.5%	75.4%
Shi's [16]	83.3%	63.1%	71.8%
Kim's [17]	83.0%	62.5%	71.3%
Ours	83.3%	60.0%	69.7%
TH-TextLoc System [17]	67.0%	57.7%	62.0%

Table 2. The Average Detection Time Comparison

Method	**Execution Time(seconds)**
Shi's	1.6
Ours	2.2
Neumann's	3.1

Table 2 summarizes the average detection time per image for three different methods on a computer with a 2.4GHz processor. Although the precision of our approach is lower than the Neumann's method, the execution time of our approach is faster than that of Neumann.

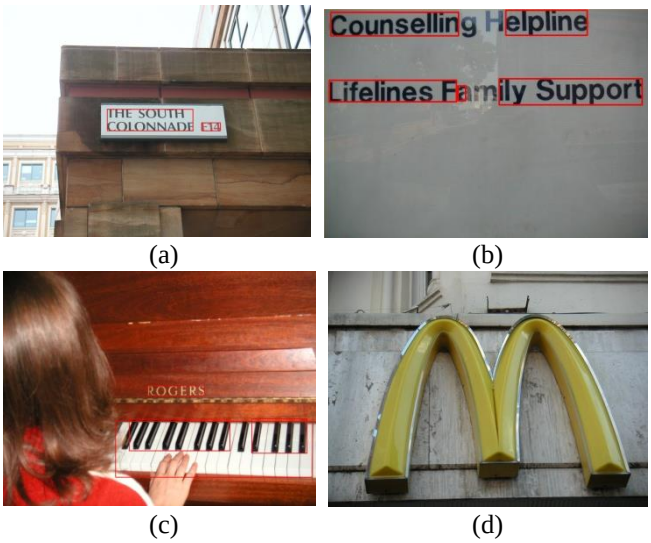
Fig. 25(a) depicts an example of successful text detection. The most important factors that interfere with text detection include blur, specular reflection, and irregular text structure Fig. 25(b) shows an incomplete detection due to specular reflection. Some detection errors arise from regular backgrounds such as bricks, leaves, or windows, because they have similar textual structures, as given in Fig. 25(c). Fig. 25(d) is another failure case as there is only one character in the scene.

B. Multilingual Scene Text image set

We collect 200 testing images and 200 training images of scene text, including Chinese and English. Texts in these scene images have both horizontal and vertical direction, with no tilt. Table 3 shows the detection result for Multilingual Scene Text image set. Precision and recall on this dataset are better since it is what the proposed algorithm is designed for. Fig. 26 depicts some detection results using images from the Multilingual Scene Text image set.

Table 3. Multilingual Scene Text Image Set Detection Result

Method	**Criteria For Assessment**		
	Precision	**Recall**	**F-measure**
Ours	87.4%	74.5%	80.4%



(a) (b) (c) (d)

Figure 25. ICDAR 2011 database detection results using our approach. (a) Successful detection (b) Incomplete detection (c) Error detection (d) Single character can not be detected.

(a) (b) (c)

Figure 26. Multilingual Scene Text image set detection results using our approach. (a) Successful detection (b) Combination error (c) Erroneous detection.

V. Conclusion

In this paper we proposed a hybrid detection method to detect scene text that can be applied to both Chinese and English languages. Experimental results indicate that the proposed mechanism can produce satisfactory detection results, especially in the mixture of English and Chinese.

For future improvements, we believe that the structure analysis of the underlying language can be further exploited to enhance the performance and precision of scene text

detection. Additionally, through extra hardware support such as multiple cameras on Amazon Fire phone, it is possible to several images of the same scene using different settings. Devices with high dynamic range (HDR) functions will enable us to obtain a series images of different intensities. In both cases, it is possible to extract more stable information and thus better detection outcome due to redundancy.

References

[1] Introduction of International Conference on Document Analysis and Recognition, http://rrc.cvc.uab.es/ (accessed December 26, 2015).

[2] Wikipedia contributors, "Google Translate", Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Google_Translate&oldid=696639548 (accessed December 26, 2015).

[3] OpenCV contributors, "text. Scene Text Detection and Recognition", http://docs.opencv.org/3.0-alpha/modules/text/doc/text.html (accessed December 26, 2015).

[4] L. Neumann and J. Matas. "Real-Time Scene Text Localization and Recognition", CVPR 2012.

[5] Liu, Cheng-Lin, et al. "Online and Offline Handwritten Chinese Character Recognition: Benchmarking on New Databases", Pattern Recognition 46.1 (2013): 155-162.

[6] Gang Zhou, Yuehu Liu, Quan Meng, and Yuanlin Zhang. "Detection Multilingual Text in Natural Scene.", IEEE-ISAS 2011.

[7] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. "Detecting Text in Natural Scenes with Stroke Width Transform", CVPR, page 2963-2970. IEEE, 2010.

[8] John Canny. "A Computational Approach to Edge Detection.", Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698, 1986.

[9] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Yu. "Detecting Texts of Arbitrary Orientations in Natural Images", CVPR, IEEE, 2012.

[10] Huizhong Chen, Sam S. Tsai, Georg Schroth, David M. Chen, Radek Grzeszczuk, and Bernd Girod. "Robust Text Detection in Natural Images with Edge-enhanced Maximally Stable Extremal Regions", IEEE Trans. on Image Processing, 2011.

[11] J. Matas, O. Chum, M. Urban, T. Pajdla. "Robust Wide Baseline Stereo From Maximally Stable Extremal Region", Proc. Of British Machine Version Conference, 2002.

[12] Yi-Feng Pan, Xinwen Hou, and Cheng-Lin Liu. "A Hybrid Approach to Detect and Localize Texts in Natural Scene Images", IEEE Trans. Image Processing, 2011.

[13] Wayne Niblack. "An Introduction to Digital Image Processing", Prentice-Hall, 1986.

[14] Boser, B. E,. Guyon, I. M, Vapnik, V. N. "A Training Algorithm for Optimal Margin Classifiers", "Proceedings of the fifth annual workshop on Computational learning theory - COLT '92". p. 144.

[15] L. Neumann and J. Matas. "On Combining Multiple Segmentations in Scene Text Recognition", in Proc. Int. Conf. on Document Analysis and Recognition, 2013.

[16] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao. "Scene Text Detection Using Graph Model Built Upon Maximally Stable Extremal Regions", Pattern Recognition Letters, vol. 34, no. 2, pp. 107–116, 2013.

[17] A. Shahab, F. Shafait, and A. Dengel. "ICDAR 2011 Robust Reading Competition Challenge 2: Reading Text in Scene Images", in ICDAR 2011, 2011, pp. 1491–1496.

Author Biographies

Wen-Hung Liao received his MS and Ph.D in 1991 and 1996, respectively, from the department of Electrical and Computer Engineering, the University of Texas at Austin. He joined National Chengchi University in Taiwan since 2000 and is currently an associate professor and chairperson of the Computer Science Department. His research interests include computer vision, pattern recognition, human computer interaction and multimedia signal processing.

Yi-Chieh Wu received the B.E. degree in Psychology from National Taiwan University, M.S degree in Computer Science from National Chengchi University, Taiwan, in 1999 and 2014. She had been working in the industry as software engineer over 16 years, and currently she is studying Ph.D. in National Chengchi University. Her research interests include user interaction analysis, computer vision, and human-computer interface.