Text String Detection From Natural Scenes by Structure-Based Partition and Grouping

Chucai Yi and YingLi Tian, *Senior Member, IEEE*

***Abstract*—Text information in natural scene images serves as important clues for many image-based applications such as scene understanding, content-based image retrieval, assistive navigation, and automatic geocoding. However, locating text from a complex background with multiple colors is a challenging task. In this paper, we explore a new framework to detect text strings with arbitrary orientations in complex natural scene images. Our proposed framework of text string detection consists of two steps: 1) image partition to find text character candidates based on local gradient features and color uniformity of character components and 2) character candidate grouping to detect text strings based on joint structural features of text characters in each text string such as character size differences, distances between neighboring characters, and character alignment. By assuming that a text string has at least three characters, we propose two algorithms of text string detection: 1) adjacent character grouping method and 2) text line grouping method. The adjacent character grouping method calculates the sibling groups of each character candidate as string segments and then merges the intersecting sibling groups into text string. The text line grouping method performs Hough transform to fit text line among the centroids of text candidates. Each fitted text line describes the orientation of a potential text string. The detected text string is presented by a rectangle region covering all characters whose centroids are cascaded in its text line. To improve efficiency and accuracy, our algorithms are carried out in multi-scales. The proposed methods outperform the state-of-the-art results on the public Robust Reading Dataset, which contains text only in horizontal orientation. Furthermore, the effectiveness of our methods to detect text strings with arbitrary orientations is evaluated on the Oriented Scene Text Dataset collected by ourselves containing text strings in nonhorizontal orientations.**

***Index Terms*—Adjacent character grouping, character property, image partition, text line grouping, text string detection, text string structure.**

I. Introduction

As indicative marks in natural scene images, text information provides brief and significant clues for many image-based applications such as scene understanding, content-based image retrieval, assistive navigation and automatic geocoding. To extract text information from camera-captured document images (i.e., most part of the captured image contains well organized text with clean background), many algorithms and commercial optical character recognition (OCR) systems have been developed [2], [32]. Liang *et al.* [18] used texture flow analysis to perform geometric rectification of the planar and curved documents. Burns *et al.* [3] performed topic-based partition of document image to distinguish text, white spaces and figures. Banerjee *et al.* [1] employed the consistency of text characters in different sections to restore document images from severe degradation based on the model of a Markov random field. Lu *et al.* [20] proposed a word shape coding scheme through three topological features of characters for text recognition in document image. All of the above algorithms share the same assumption that locations of text characters are approximately predictable, and background interference does not resemble text characters.

Fig. 1. Examples of text in natural scene images.

Different from document images, in which text characters are normalized into elegant poses and proper resolutions, natural scene images embed text in arbitrary shapes, sizes, and orientations into complex background, as shown in Fig. 1. It is impossible to recognize text in natural scene images directly because the off-the-shelf OCR software cannot handle complex background interferences and nonorienting text lines. Thus, we need to detect image regions containing text strings and their corresponding orientations. This is compatible with the detection and localization procedure described in the survey of text extraction algorithms [11], [38]. With knowledge of text string orientations, we can normalize them to horizontal. Some algorithms of scene text normalization are introduced in [4], [18], and [26]. However, the algorithms described in this paper will focus on text detection.

Previous work on text detection can be roughly classified into two categories. The first category focuses on text region initialization and extension by using distinct features of text characters. To extract candidates of text regions, Kasar *et al.* [12] first assigned a bounding box to the boundary of each candidate character in the edge image and then detected text characters based

Manuscript received July 23, 2010; revised November 25, 2010 and February 03, 2011; accepted February 21, 2011. Date of publication March 14, 2011; date of current version August 19, 2011. This work was supported in part by the National Institutes of Health under Grant 1R21EY020990, the National Science Foundation under Grant IIS-0957016, and the Army Research Office under Grant W911NF-09-1-0565. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Patrick Flynn.

C. Yi is with the Graduate Center, City University of New York, New York, NY 10016 USA (e-mail: cyi@gc.cuny.edu).

Y. Tian was with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA. She is now with City College, City University of New York, New York, NY 10031 USA (e-mail: ytian@ccny.cuny.edu).

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/TIP.2011.2126586

on the boundary model (i.e., no more than two inner holes in each bounding box of alphabets and letters in English). Tran *et al.* [35] calculated ridge points in different scales to describe text skeletons at the level of higher resolution and text orientations at the level of low resolution. Liu *et al.* [19] designed a stroke filter to extract the stroke-like structures. Sobottka *et al.* [33] combined a top–bottom analysis based on color variations in each row and column with a bottom–top analysis based on region growing by color similarity. Hasan *et al.* [8] and Park *et al.* [29] designed robust morphological processing. Wolf *et al.* [37] improved Otsu's method to binarize text regions from background, followed by a sequence of morphological processing to reduce noise and correct classification errors. To group together text characters and filter out false positives, these algorithms employed similar constraints involved in character, such as the minimum and maximum size, aspect ratio, contrast between character strokes and background, and the number of inner holes. However, they usually fail to remove the background noise resulting from foliage, pane, bar, or other background objects that resemble text characters. To reduce background noise, the algorithms in the second category partition images to blocks and then groups the blocks verified by the features of text characters. Shivakumara *et al.* [31] applied different edge detectors to search for blocks containing the most apparent edges of text characters. Lefevre *et al.* [17] further designed a fusion strategy to combine detectors of color, texture, contour, and temporal invariance, respectively. Weinman *et al.* [36] used a group of filters to analyze texture features in each block and joint texture distributions between adjacent blocks by using conditional random field. One limitation of these algorithms is that they used noncontent-based image partition to divide the image spatially into blocks of equal size before grouping is performed. Noncontent-based image partition is very likely to break up text characters or text strings into fragments which fail to satisfy the texture constraints. Thus, Phan *et al.* [30] performed line-by-line scans in edge images to combine rows and columns with high density of edge pixels into text regions. Gao *et al.* [7] and Suen *et al.* [34] performed heuristic grouping and layout analysis to cluster edges of objects with similar color, position, and size into text regions. However, these algorithms are not compatible with slanted text lines. Myers *et al.* [26] rectified the text line in 3-D scene images by using horizontal and vertical features of text strings, but their work does not focus on detecting text line on complex backgrounds. Epshtein *et al.* [6] designed a content-based partition named as stroke width transform to extract text characters with stable stroke widths. In addition, the color uniformity of text characters in natural scene image is taken into account for content-based partition [4], [13], [14], [24], [25]. However the unexpected background noises might share the same colors with text characters, so texture features of characters are still required. The algorithms in our proposed framework belong to this category of partition and grouping, but our content-based partition is involved in both gradient features and color features.

Different from rule-based text detection as the above algorithms, Chen *et al.* [5] and Ho *et al.* [9] adopted Adaboost learning methods by using text features to establish the corresponding classifiers. Pan *et al.* [28] took edge segment of text character as feature of sparse representation and applied a K-SVD based learning processing for text detection. Jiang *et al.* [10] took the size and shape of the text characters denoted by connected components (CC) as features and used cascade SVM learning classifiers to detect text from scene images, while Kumar *et al.* [16] took the globally matched wavelets as features to compute candidate text regions, and used Markov random field (MRF) to refine the extracted text regions.

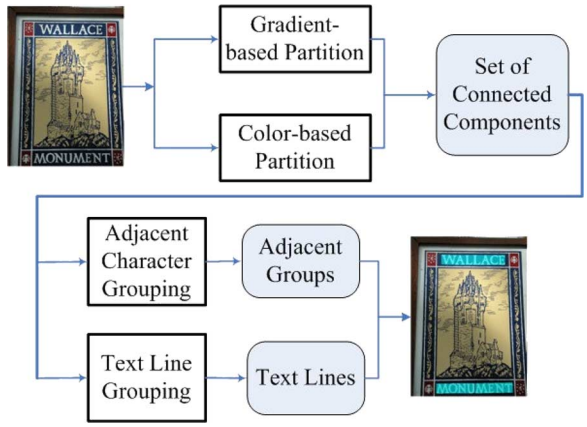


Fig. 2. Flowchart of the proposed framework of text string detection.

Although many research efforts have been made to detect text regions from natural scene images, more robust and effective methods are expected to handle variations of scale, orientation, and clutter background.

The remainder of this paper is organized as follows. Section II briefly overviews the proposed framework. Section III describes our proposed algorithms of image partition for extracting text character candidates. Section IV introduces two grouping methods to extract text strings. Section V presents the system implementation in multiple scales. Experiments and result analysis are described in Section VI. We conclude the paper and future research directions in Section VII.

II. Overview of Our Framework

In this paper, we propose a new framework to extract text strings with multiple sizes and colors, and arbitrary orientations from scene images with a complex and cluttered background. Fig. 2 depicts the flowchart of our framework. The proposed framework consists of two main steps, given here.

Step 1) Image partition to find text character candidates based on gradient feature and color uniformity. In this step, we propose two methods to partition scene images into binary maps of nonoverlapped connected components: *gradient-based method* and *color-based method*. A postprocessing is then performed to remove the connected components which are not text characters by size, aspect ratio, and the number of inner holes.

Step 2) Character candidate grouping to detect text strings based on joint structural features of text characters in each text string such as character sizes, distances between two neighboring characters, and character alignment. In this step, we propose two methods of structural analysis of text strings: *adjacent character grouping method* and *text line grouping method*.

The proposed framework is able to effectively detect text strings in arbitrary locations, sizes, orientations, colors and slight variations of illumination or shape of attachment surface. Compared with the existing methods which focus on independent analysis of single character, the text string structure is more robust to distinguish background interferences from text information. Experiments demonstrate that our framework outperforms the state-of the-arts on Robust Reading Dataset and effective to detect text strings with arbitrary orientations on our new collected Oriented Scene Text Dataset.

Overall, the work introduced in this paper offers the following main contributions to robust detection of text strings with variations of scale, color, orientation, and clutter background from natural scene images.

- Most existing work of text detection from natural scene images focuses on detecting text in horizontal orientation or independent analysis of single character. We propose a new framework to robustly detect text strings with variations of orientation and scale from complex natural scene images with clutter background by integrating different types of features of text strings.
- We formally draw a clear distinction between text character and text string by an incremental processing including image partition to extract candidate character components and connected component grouping to extract text strings.
- We model text character by features of local gradient and stroke structure. Under this model, we develop a gradient-based partition algorithm to compute connected components of candidate characters. It is more robust and achieves better results than directly using morphological processing operators.
- We model text string as a text line from cascading of connected component centroids based on Hough transform. Then, we extend the set of text features from single character component to text line structure, which is used to detect text strings in arbitrary orientations.
- We collect an oriented scene text dataset (OSTD) with text strings in arbitrary orientations, which is more challenging than the existing datasets for text detection. Text string regions are manually labeled in XML file. The OSTD dataset contains 89 images of colorful logos, indoor scenes, and street views. The resolutions of most images are from 600 × 450 to 1280 × 960. Each image contains two text strings on average. The OSTD dataset will be released to the public in our research website.

III. Image Partition

To extract text information from a complex background, image partition is first performed to group together pixels that belong to the same text character, obtaining a binary map of candidate character components. Based on local gradient features and uniform colors of text characters, we design a gradient-based partition algorithm and a color-based partition algorithm, respectively.

A. Gradient-Based Partition by Connecting Paths of Pixel Couples

Although text characters and strings vary in font, size, color, and orientation, they are composed of strokes which are rectangle connected components with closed-width boundaries and uniform torso intensities. In [6], each pixel is mapped to the width of the stroke in which it is located, and then the consistency of the stroke width is used to extract a candidate character component. In our method, each pixel is mapped to the connecting path of a pixel couple, defined by two edge pixels p and q on an edge map with approximately equal gradient magnitudes and opposite directions, as shown in Fig. 3(a). Each pixel couple is connected by a path. Then the distribution of gradient magnitudes at pixels of the connecting path is computed to extract candidate character component.

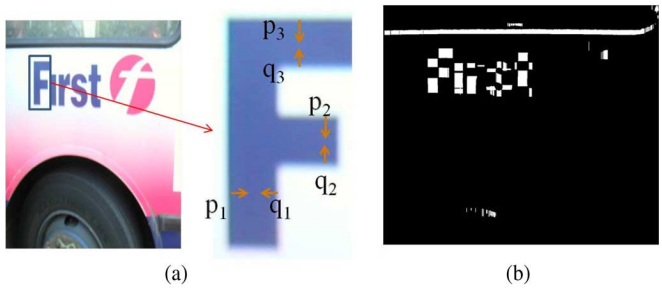


Fig. 3. (a) Examples of pixel couples. (b) Connecting paths of all pixel couples are marked as white foreground while other pixels are marked as black background.

Fig. 3(a) depicts that a character boundary consists of a number of pixel couples. We model the character by distribution of gradient magnitudes and stroke size including width, height, and aspect ratio. The partitioned components are calculated from connecting path of pixel couple across the pixels with small gradient magnitudes.

On the gradient map, $G_{\mathrm{mag}}(p)$ and $d_p(-\pi < d_p \leq \pi)$ are used, respectively, to represent the gradient magnitude and direction at pixel p. We take an edge pixel p from edge map as starting point and probe its partner along a path in gradient direction. If another edge pixel q is reached where gradient magnitudes satisfy $|G_{\mathrm{mag}}(p) - G_{\mathrm{mag}}(q)| < 20$ and directions satisfy $|d_q - (d_p - (d_p/|d_p|) * \pi)| < \pi/6$, we obtain a pixel couple and its connecting path from p to q. This algorithm is applied to calculate connecting paths of all pixel couples. Fig. 3(b) marks all of the connecting paths shorter than 30 as white foreground. To perform the gradient-based partition, we employ gradient magnitude at each pixel on the connecting path and length of connecting path l describing the size of connected component to be partitioned. The partition process is divided into two rounds. In the first round, the length range of connecting path is set as $0 < l \leq 30$ to describe stroke width. For each pixel couple whose connecting path falls on this length range, we establish an exponential distribution of gradient magnitudes of the pixels on its connecting path, denoted by

$$g(G_{\mathrm{mag};\lambda}) = \lambda \exp(-\lambda G_{\mathrm{mag}}) \tag{1}$$

where the decay λ rate is estimated by $\hat{\lambda} = 1/\sum G_{\mathrm{mag}}$. A larger decay rate leads to faster falloff of gradient magnitudes on a connecting path. This means that the connecting path crosses a number of pixels with small gradient magnitudes on gradient map. This feature is consistent with the intensity

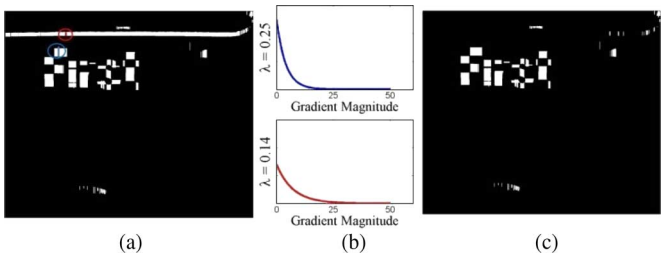


Fig. 4. (a) Two connecting paths of pixel couples marked by blue and red circles, respectively. (b) Corresponding exponential distribution of gradient magnitudes on the connecting paths. (c) Partitioned components obtained from the first round.

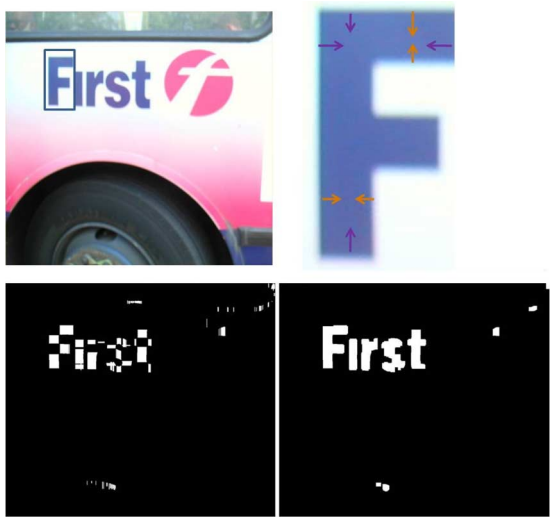


Fig. 5. Connecting path of a pixel couple along the larger side of a rectangle stroke is analyzed in the second round partition. The top row shows pixel couples in purple across the larger side of rectangle strokes. The bottom row presents the partitioned components obtained from the first round and the second round, respectively.

uniformity inside the character strokes. Thus, the connecting path with greater decay rate is marked as white foreground representing candidate character component, as shown in Fig. 4. To extract the complete stroke in rectangle shape, we start the second round to analyze the connecting paths along the stroke height (larger side). Since the aspect ratio of the rectangle stroke is no more than 6:1, we extend the length range of the connecting path to $0 < l \leq 180$. Then, we repeat the same analysis of gradient magnitudes for the connecting path not only falling on this length range but also passing through the regions of candidate character components obtained from the first round. At last, we perform morphological close and open as postprocessing to refine the extracted connected components, as shown in Fig. 5. The refined connected components are taken as candidate character components.

The gradient-based partition generates a binary map of candidate character components on black background. By the model of local gradient features of character stroke, we can filter out background outliers while preserving the structure of text characters. Fig. 6 demonstrates that the gradient-based partition performs better on character component extraction than morphological processing.

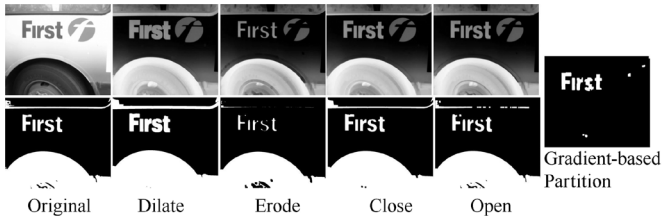


Fig. 6. Connected components obtained from direct morphological processing on gray images and corresponding binary images. We compare results of four morphological operators with result of our gradient-based partition.

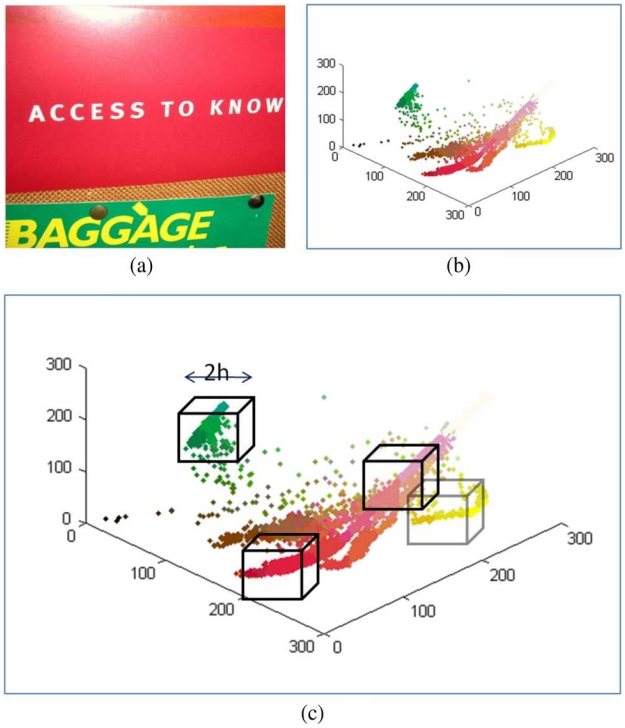


Fig. 7. (a) Scene image with multiple colors. (b) Color distribution in RGB space. (c) Four of the initial cube color clusters with radius h.

B. Color-Based Partition by Color Reduction

In most scene images, text strings are usually composed of characters with similar colors. Thus, we can locate text information by extracting pixels with similar colors. To label a region of connected pixels with similar colors as a connected component, we develop color-based partition method. Inspired by [27], we perform color reduction by using color histogram and weighted K-means clustering through the following steps.

First, a Canny edge detector is employed to obtain edge image. Second, we calculate color histograms of the original input image. To capture the dominant colors and avoid drastic color variations around edge pixels, only nonedge pixels are sampled for color histogram calculation to obtain a set of sampled pixels P. Third, after mapping all of the pixels from spatial domain to RGB color space, as shown in Fig. 7(b), weighted K-means clustering is performed to group together the pixels with similar colors. By using the initial mean point p_i which is randomly selected from the sampled pixels and an initial radius h, color clusters in RGB color space is

Fig. 8. Some examples of color-based partition, where the left column contains original images and other columns contain the corresponding dominant color layers.

established [cf. Fig. 7(c)], covering any pixel q whose color is close to p_i

$$\begin{aligned} \text{Cover}(q|p_i) &= \begin{cases} 1, & IF\ K1 \text{ is satisfied} \\ 0, & \text{otherwise} \end{cases} \\ \text{Cluster}(p_i) &= \{q|\text{Cover}(q|p_i) = 1\}. \end{aligned} \tag{2}$$

Repeat the process of cluster establishment until all of the pixels have been selected by at least one color cluster, as described in

$$P = \bigcup_i \text{Cluster}(p_i) \tag{3}$$

where $K1$ represents

$$\{|R_q - R_{p_i}| < 2h, |G_q - G_{p_i}| < 2h, |B_q - B_{p_i}| < 2h\}$$

and R, G, B denote the three color components, respectively, in RGB color model.

Thus, the K-value has been fixed by the number of color clusters. Taking the color histogram as a weight table, weighted average is calculated iteratively to update the values of cluster mean points until the distance change is smaller than a predefined threshold. Fourth, the clusters whose mean points are sufficiently close are merged together to produce a final cluster, which corresponds to a color layer. The number of color layers depends on the number of dominant hues in original image and the initial radius h. The larger the cluster radius is, the more pixels will be covered by each color cluster, so the total number of color clusters is reduced, which results in less color layers. Experiments are performed to compare the detection rate among different radius h, and the results are presented in Section VI-C.

Some examples of the color-based image partition method are displayed in Fig. 8. Each input image is partitioned to several color layers. A color layer that consists of only one foreground color on white background is a binary map of candidate character components. Then, connected component analysis is performed to label foreground regions of connected pixels.

IV. Connected Components Grouping

The image partition creates a set of connected components S from an input image, including both text characters and unwanted noises. Observing that text information appears as one or more text strings in most natural scene images, we perform heuristic grouping and structural analysis of text strings to distinguish connected components representing text characters from those representing noises. Assuming that a text string has at least three characters in alignment, we develop two methods to locate regions containing text strings: adjacent character grouping and text line grouping, respectively. In both algorithms, a connected component C is described by four metrics: $height(.)$, $width(.)$, $centroid(.)$, and $area(.)$. In addition, we use $D(.)$ to represent the distance between the centroids of two neighboring characters.

A. Adjacent Character Grouping

Text strings in natural scene images usually appear in alignment, namely, each text character in a text string must possess character siblings at adjacent positions. The structure features among sibling characters can be used to determine whether the connected components belong to text characters or unexpected noises. Here, five constraints are defined to decide whether two connected components are siblings of each other.

1) Considering the capital and lowercase characters, the height ratio falls between and $1/T_1$ and T_1.
2) Two adjacent characters should not be too far from each other despite the variations of width, so the distance between two connected components should not be greater than T_2 times the width of the wider one.
3) For text strings aligned approximately horizontally, the difference between y-coordinates of the connected component centroids should not be greater than T_3 times the height of the higher one.
4) Two adjacent characters usually appear in the same font size, thus their area ratio should be greater than $1/T_4$ and less than T_4.
5) If the connected components are obtained from gradient-based partition as described in Section III-A, the color difference between them should be lower than a predefined threshold T_5 because the characters in the same string have similar colors.

In our system, we set $T_1 = T_4 = 2$, $T_2 = 3$, $T_3 = 0.5$, and $T_5 = 40$. According to the five constraints, a left/right sibling set F_L/F_R is defined for each connected component C as the set of sibling components located on the left/right of C.

To extract regions containing text strings based on adjacent character grouping, we first remove the small connect components $(\text{area} < T_s)$ from the set of connected components S. In our system, we set $T_s = 20$. Then, a left-sibling set F_L and a right-sibling set F_R for each connected component C are initialized to record its sibling connected components on the left and right, respectively. For two connected components C and C', they can be grouped together as sibling components if the above five constraints are satisfied. When C and C' are grouped together, their sibling sets will be updated according to their relative locations, that is, when C is located on the left of C', C' will be added into the right-sibling set of C, which is simultaneously added into the left-sibling set of C'. The reverse opera-

Fig. 9. (a) Sibling group of the connected component "r" where "B" comes from the left sibling set and "o" comes from the right sibling set. (b) Merge the sibling groups into an adjacent character group corresponding to the text string "Brolly?" (c) Two detected adjacent character groups marked in red and green, respectively.

tion will be applied when C is located on the right of C'. When a connected component corresponds to a text character, the five constraints ensure that its sibling set contains sibling characters rather than the foliage, pane or irregular grain.

For a connected component C, if both sibling sets are not empty and their size difference does not exceed 3, a sibling group $SG(C)$ is defined as the union of the two sibling sets and the connected component itself. At this point, each sibling group can be considered as a fragment of a text string. To create sibling groups corresponding to complete text strings, we merge together any two sibling groups $SG(C_1)$ and $SG(C_2)$ when the intersection $SG(C_1) \cap SG(C_2)$ contains no less than two connected components. Repeat the merge process until no sibling groups can be merged together. As shown in Fig. 9, the resulting union of connected components is defined as adjacent character group denoted by AG, which is a subset of the set of connected components S. Table I summarizes our algorithm in detail.

Each text string can be mapped into an adjacent character group. However, some adjacent character groups correspond to unexpected false positives instead of real text strings. We design three filters based on structure of text strings to remove these false positive adjacent character groups. The filter is described by coefficient of variation CV, which is defined as the ratio of the standard deviation σ to the mean μ. First, inside an adjacent character group, the coefficient of variation of connected component areas (calculated by the number of pixels) should be confined by an upper bound

$$CV_{\text{area}} = \frac{\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\text{area}(C_j) - \mu_A\right)^2}}{\mu_A}. \tag{4}$$

Second, the distances between every two neighboring connected component centroids in the same text string should be relatively stable

$$CV_{\text{distance}} = \frac{\sqrt{\frac{1}{N-1}\sum_{j=1}^{N-1}\left(D(m_j, m_{j+1}) - \mu_D\right)^2}}{\mu_D} \tag{5}$$

where $m_i = \text{centroid}(C_i)$, $\mu_A = (1/N)\sum_{i=1}^{N}\text{area}(C_i)$, and $\mu_D = (1/(N-1))\sum_{i=1}^{N-1} D(m_i, m_{i+1})$. Third, the coefficient of variation of stroke width measured by the horizontal scan lines passing through the connected components of the adjacent character group corresponding to a text string should be less than a predefined threshold.

Text string in scene images can be described by corresponding adjacent character groups. To extract a region containing a text string, we calculate the minimum circumscribed rectangle covering all of the connected components in the corresponding adjacent character group.

B. Text Line Grouping

In order to locate text strings with arbitrary orientations, we develop text line grouping method. To group together the connected components which correspond to text characters in the same string which is probably nonhorizontal, we use centroid as the descriptor of each connected component. Given a set of connected component centroids, groups of collinear character centroids are computed, as shown in

$$M = \{m | C \in S \text{ and } m = \text{centroid}(C)\} \tag{6}$$

$$L = \{G | G \subseteq M, |G| \geq 3, \forall m_i, m_j, m_k \in G, \text{ they are character centroids and they are colinear.}\} \tag{7}$$

where M denotes the set of centroids of all of the connected components obtained from image partition, and L denotes the set of text lines which are composed of text character centroids in alignment.

A solution is to search for satisfied centroid groups in the power set of M, but the complexity of this algorithm will be $O(2^{|M|})$, where $|M|$ represents the number of centroids in the set M. We design an efficient algorithm to extract regions containing text strings. At first, we remove the centroids from the set M if areas of their corresponding connected components are smaller than the predefined threshold T_s. Then, three points m_i, m_j, and m_k are randomly selected from the set M to form two line segments. We calculate the length difference Δd and incline angle difference $\Delta\theta$ between line segments $m_i m_j$ and $m_j m_k$, as shown in

$$\Delta d = \frac{D(m_i, m_j)}{D(m_j, m_k)} \tag{8}$$

$$\Delta\theta = \begin{cases} |\theta_{ij} - \theta_{jk}|, & \text{if } |\theta_{ij} - \theta_{jk}| \leq \frac{\pi}{2} \\ |\theta_{ij} - \pi - \theta_{jk}|, & \text{if } |\theta_{ij} - \theta_{jk}| > \frac{\pi}{2} \text{ and } \theta_{ij} > \theta_{jk} \end{cases} \tag{9}$$

where d is length of line segment and $d > 0$, θ is the angle of incline, and $0 \leq \theta < \pi$. The three centroids are approximately collinear if $1/T_6 \leq \Delta d \leq T_6$ and $\Delta\theta \leq T_7$. In our system, we set $T_6 = 2$ and $T_7 = \pi/12$. Thus, they compose a preliminary fitted line $l_u = \{m_i, m_j, m_k\}$, where u is the index of the fitted line. After finding out all of the preliminary fitted lines, we apply

TABLE I
PROCEDURE OF ADJACENT CHARACTER GROUPING

Locating Text Strings by adjacent character groups

$S := S - \{C | C \in S, area(C) < T_s\}$;
for every connected component $C \in S$
  Initialize the sibling sets F_L and F_R;
endfor

for two connected components C and C' with sibling sets $F_L \cup F_R$ and ${F_L}' \cup {F_R}'$ respectively
  if $1/T_1 < height(C)/\text{height}(C') < T_1$
    & $D(centroid(C).x, centroid(C').x) \le T_2 * \max\{width(C), width(C')\}$
    & $D(centroid(C).y, centroid(C').y) \le T_3 * \max\{height(C), height(C')\}$
    & $1/T_4 < Area(C)/Area(C') < T_4$
    & difference of mean RGB color value is less than T_5
      if $centroid(C).x \le centroid(C').x$,
        $F_R := F_R \cup \{C'\}$; ${F_L}' := {F_L}' \cup \{C\}$;
      else
        $F_L := F_L \cup \{C'\}$; ${F_R}' := {F_R}' \cup \{C\}$;
      endif
  endif
endfor

for every connected component C
  if $F_L \neq \emptyset$ & $F_R \neq \emptyset$ & $\big||F_R| - |F_L|\big| \le 3$
    $SG(C) := F_L \cup F_R \cup \{C\}$;
  endif
endfor

Repeat the following until no merge is performed and the rest sibling groups will be upgraded to adjacent character groups
  for two sibling groups $SG(C_1)$ *and* $SG(C_2)$
    if $|SG(C_1) \cap SG(C_2)| \ge 2$
      $SG(C_1) \coloneqq SG(C_1) \cup SG(C_2)$; $SG(C_2) \coloneqq \emptyset$;
    endif
  endfor

Filter out false positives by the three filters decided by the area, distance and stroke width respectively.

Calculate extracted regions based on the adjacent character groups.

Hough transform to describe the fitted line l_u by $\langle r_u, \theta_u \rangle$, resulting in $l_u = \{m | h(r_u, \theta_u, m) = 0\}$, where $h(r_u, \theta_u, m) = 0$ is the equation of the fitted line in the Hough space. Thus, other collinear centroids along l_u can be added into the end positions to form a complete text string increasingly. Table II summarizes our algorithms in detail.

The centroids from noise components can also be aligned as a line. To remove these false positive fitted lines, two constraints are further used to distinguish the fitted lines corresponding to text strings from those generated by unexpected noises based on the structure features of text strings. For a text string, the coefficient of variation CV of areas of the connected components corresponding to the centroids located in the fitted line should be smaller than a predefined threshold, and the distances between every two neighboring centroids should not have large coefficient of variations. Fig. 10 illustrates the processing of fitted line refinement.

For now, each text string is described by a fitted line. The location and size of the region containing a text string is defined by the connected components whose centroids are cascaded in the corresponding fitted line. The orientation of the text string is denoted by the incline angle θ of the fitted line. To cover these connected components properly, we calculate the minimum circumscribed rectangle as the extracted text region.

TABLE II
PROCEDURE OF TEXT LINE GROUPING

Locating Text Strings by Grouping Centroids of Connected Components into Fitted Text Line

$M = M - \{m | m = centroid(C), area(C) < T_s\}$;
for every three points $m_i, m_j, m_k \in M$,
  calculate Δd and $\Delta\theta$
  if $0.5 \le \Delta d \le 2$ *and* $\Delta\theta \le \frac{\pi}{12}$
    $l_u := \{m_i, m_j, m_k\}$;
  endif
endfor

for every preliminary fitted line
  for every $m_t \in M$ *and* $m_t \notin l_u$
    $\langle r_u, \theta_u \rangle := Hough(l_u)$ where $l_u = \{m | h(r_u, \theta_u, m) = 0\}$
      if $h(r_u, \theta_u, m_t) < \varepsilon$
        & fitted line $l_u \cup \{m_t\}$ meets the two constraints
          $l_u := l_u \cup \{m_t\}$
          $\langle r_u, \theta_u \rangle := Hough(l_u)$
      endif
    $L := L \cup \{l_u\}$
  endfor
endfor

Filter out false positive fitted lines in L by the coefficient of variations of areas and distances, and calculate extracted regions based on the positive fitted lines.

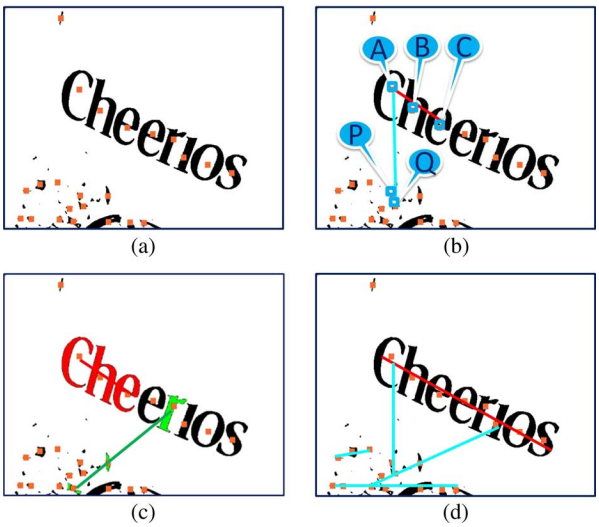


Fig. 10. (a) Centroids of connected components in a color layer shown in Fig. 8. (b) $D(m_A, m_B)$ approximately equals to $D(m_B, m_C)$ in text region while $D(m_A, m_P)$ is much larger than $D(m_P, m_Q)$ in background, where $D(.)$ represents Euclidean distance. (c) Three neighboring connected components in red share similar areas while those in green have very different areas. (d) Resulting fitted lines from centroids cascading. Red line corresponds to text region while cyan lines are false positives to be removed.

V. Multiscale Computing

To detect text strings in different sizes and alleviate the computational complexity of adjacent character grouping and text line fitting, the scene images are processed in scale space. Text characters of different sizes will be processed in different scales, and the connected components whose areas are smaller than T_s in the current scale will be directly removed by the grouping algorithms. Since the characters in same text string have similar font size, we can also save the computational cost of grouping together two connected components with excessive area difference. According to Lindeberg's theory of scale invariance [23], we perform scale space analysis by the convolution of original image $I(x, y)$ and Gaussian kernels $G(x, y, \sigma)$, where σ is the scale and x, y are coordinates. In scale space, increasing scale level σ results in image blur so that the connected components are gradually removed. In the process of scale-level increasing, smaller connected components will disappear before larger ones. We calculate the images $I_\sigma(x, y)$ in scale space by

$$I_\sigma(x, y) = I(x, y) * G(x, y, \sigma). \tag{10}$$

A character searches for its partners with approximate font sizes scale by scale from coarse to fine where it shows up. Larger size characters are processed in a coarser scale while smaller size characters are visited in a finer scale, as shown in Fig. 11. Once a character is integrated into an adjacent character group or a text line at some scale, it will not be involved in the processing at subsequent scale levels. The extracted regions will be scaled up but disabled (red regions in Fig. 11). Thus, sibling grouping or line fitting is prevented across the centroids of different scale levels. The multiscale computing enables our algorithm to be computationally tractable for the scene image whose size is up to 2000 × 2000.

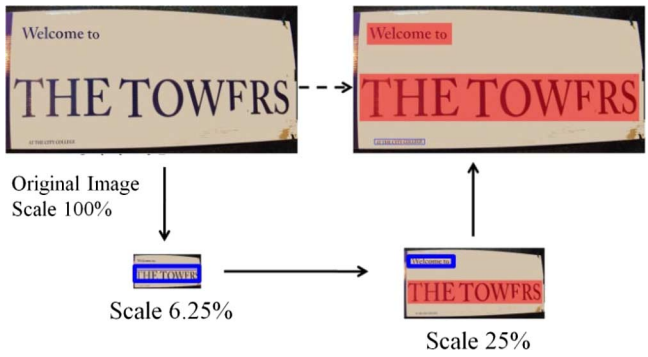


Fig. 11. Multiscale computing. Regions detected at the current scale level are shown as blue rectangles; red regions containing the characters processed at coarser scale level will be skipped.

VI. Experimental Results

A. Datasets

Two datasets are employed to evaluate the proposed algorithms. The first is the Robust Reading Dataset[1] from ICDAR 2003. In this dataset, there are 509 images in total, in which 258 images are prepared for training and 251 images for testing. The image regions containing text strings are labeled in a XML file. Each image contains about four text regions on average. All of the text strings in this dataset are in horizontal. In our testing, we selected 420 images which are compatible with the assumption that a text string contains at least three characters with relatively uniform color. Furthermore, to verify that text line grouping can detect text strings with arbitrary orientations, we collect 89 scene images with nonhorizontal text strings to construct the OSTD. The resolutions of most images are from 600 × 450 to 1280 × 960. The average number of text strings is two on each image. Text string regions are also manually labeled in the .xml file. This OSTD dataset contains colorful logos, indoor scenes, and street views.

B. Performance Evaluation

To evaluate the performance, we calculate two metrics, precision p and recall r as in [21], [22]. Here, precision p is the ratio of area of the successfully extracted text regions to area of the whole detected region, and recall is the ratio of area of the successfully extracted text regions to area of the groundtruth regions. The area of a region is the number of pixels inside it. Low precision means overestimate while low recall means underestimate. To combine p and r, a standard f measure is defined by

$$f = 1 \Big/ \left(\frac{\alpha}{p} + \frac{(1-\alpha)}{r} \right) \tag{11}$$

where α represents the relative weight between the two metrics. In our evaluation, we set $\alpha = 0.5$.

Since two algorithms are designed for partition and grouping in our framework, we evaluate four types of combinations of them on the Robust Reading dataset (RRD): 1) gradient-based partition with adjacent character grouping (GA); 2) color-based

[1][Online]. Available: http://algoval.essex.ac.uk/icdar/Datasets.html

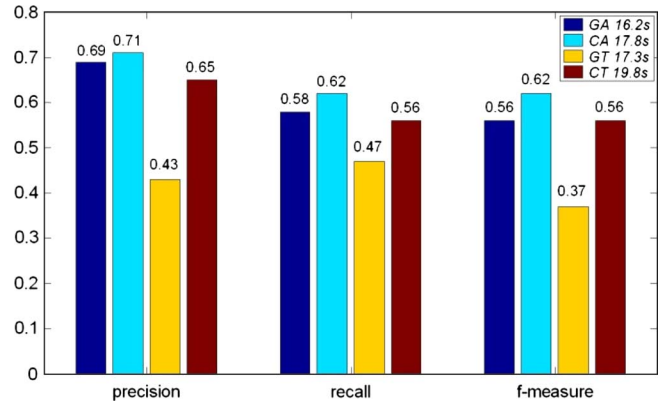


Fig. 12. Performance evaluation of the four combinations of partition and grouping on the Robust Reading Dataset, where the box presents the average time of text string detection in each scene image.

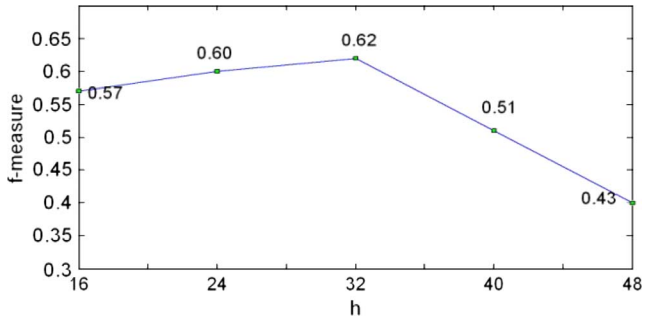


Fig. 13. Variation of f-measure obtained from color-based partition and adjacent character grouping according to the radius h defined in the color-based partition.

TABLE III
COMPARISON BETWEEN OUR ALGORITHM AND THE TEXT DETECTION ALGORITHMS PRESENTED IN [21] AND [22] ON THE ROBUST READING DATASET

	Precision	Recall	f-measure
Ours	**0.71**	**0.62**	**0.62**
H. Becker	0.62	0.67	0.62
A. Chen	0.60	0.60	0.58
Ashida	0.55	0.46	0.50
HWDavid	0.44	0.46	0.45
Q. Zhu	0.33	0.40	0.33
Wolf	0.30	0.44	0.35
J. Kim	0.22	0.28	0.22
Todoran	0.19	0.18	0.18
N. Ezaki	0.18	0.36	0.22

partition with adjacent character grouping (CA); 3) gradient-based partition with text line grouping (GT); and 4) color-based partition with text line grouping (CT). Since the text line grouping can detect text strings with arbitrary orientations. We furthermore perform evaluation of gradient-based partition with text line grouping (GT) and color-based partition with text line grouping (CT) on our OSTD.

C. Results and Discussions

The experimental results on the Robust Reading dataset are illustrated in Fig. 12, where blue bars denote results of GA, cyan bars denote results of CA, yellow bars denote results of GT, and red bars denote results of CT. The average time of text string detection is presented in the upper boxes.

The combination of color-based partition and adjacent character grouping (CA) achieves the highest precision and recall. In most of the cases, color uniformity acts as a stronger indicator to distinguish the connected components of text characters from surrounding background. However color-based partition takes more computing time than gradient-based partition. Also, color-based partition makes adjacent character grouping be performed in each of the color layers. Color-based partition still performs better when adjacent character grouping is replaced by the text line grouping. Fig. 12 also illustrates that text line grouping gives lower efficiency and precision than the adjacent character grouping for either partition. Adjacent character grouping is supported by the information of text orientations while text line grouping is performed for arbitrary text orientations, so its calculation cost is more expensive. Meanwhile, the indetermination of text orientation produces more false positive fitted lines.

Since the results of color-based partition depend on the initial radius of the color cluster in RGB space, we perform another experiment on the Robust Reading dataset to select the best radius h which achieves the highest f-measure in the text detection. Fig. 13 illustrates that the best radius is 32. When the radius turns smaller, a text character is very likely to break up into different color layers because the partition would be sensitive to the illumination variation. When the radius turns larger, more background noises would be assigned to the same color layer as the text characters. Thus, the radius 32 is a tradeoff to deal with the two negative situations.

By comparison with the algorithms presented in the text locating competition in ICDAR, the precision of our algorithm achieves the first rank while the recall and f-measure is comparable with the algorithms with the high performance, as shown in Table III. As a ruled-based algorithm, no trained classifiers can be applied directly, so it takes more time to perform the text string detection than the learning-based algorithms when not taking into account the time spent on training.

Some example results of text string detection on the Robust Reading Dataset are presented in Fig. 14. Instead of using the rectangle line to denote the borders of text regions, we calculate the minimum oriented rectangle E (marked in cyan in Fig. 14) to cover the detected text strings.

The experiment on the OSTD demonstrates that the text line grouping in our framework is able to detect the text strings with arbitrary orientations, as shown in Fig. 15. Without the orientation of the text line, the multiple-line text leads to text line grouping among the characters of different text strings. The overfitting comes out as the overlap of rectangle text regions. However, it will not influence the location of detected text strings because we take the union of the text regions from the fitted text lines as the final detected text regions.

In the experiment, color-based partition and text line grouping (CT) are performed to extract text strings with arbitrary orientations. The resulting precision, recall, and f-measure are 0.56, 0.64, and 0.55, respectively. We can see that the performance, especially the precision, is lower than the experimental results on horizontal text dataset. Based on the definition in Section VI-B, precision p is related to the slant angles of text lines, as shown in

$$precision = p(\Theta) = \frac{|l(\Phi) \cap h(\Theta)|}{|h(\Theta)|} \tag{12}$$

Fig. 14. Some example results of text string detection on the Robust Reading Dataset. The detected regions of text strings are marked in cyan.

Fig. 15. Example results of text detection on the OSTD which contain nonhorizontal text strings. The detected regions of text strings are marked in cyan.

where $h(\Theta)$ denotes the detected text lines with slant angle Θ, $l(\Phi)$ denotes the groudtruth text lines with slant angle Φ, and $|.|$ denotes the number of including text lines. Supposing that all of the positive text lines have been detected, we only consider the false positives, and then there is $h(\Theta) \supseteq h(\Phi)$, so $|h(\Theta)| \geq |h(\Phi)|$. Since $\forall \theta_i \notin \Phi$, there is $l(\theta_i) = \varnothing$. Thus, $l(\Phi) \cap h(\Theta) = l(\Phi) \cap h(\Phi)$. According to (12), we calculate

$$\Phi = \arg\max_{\Theta} \left(p(\Theta) \right). \tag{13}$$

If the orientations of text strings Φ have been known, we can get $precision = p(\Phi)$ exactly. However, there is no prior knowledge on text string orientations in the process of text line grouping. False positives related to slant angles of text strings will be introduced to decline the precision.

Some typical false positive text regions are presented in Fig. 16. They originate from the linear alignment of the noise components in similar sizes.

Most of the text strings in video have uniform color and horizontally aligned characters, which are compatible with our algorithms. We are able to extract the video text string by using color-based partition and adjacent character grouping (CA), as shown in Fig. 17.

Fig. 16. Some results containing false positive text regions which are marked in red, while the true text regions are marked in cyan color.

Fig. 17. Some results of video text string detection.

Fig. 18. Examples of images where our method fails.

Fig. 18 depicts some examples that our method cannot handle to locate the text information because of very small size, overexposure, characters with nonuniform colors or fade, strings with less than three character members, and occlusions caused by other objects such as wire mesh.

VII. Conclusion and Future Work

Due to the unpredictable text appearances and complex backgrounds, text detection in natural scene images is still an unsolved problem. To locate text regions embedded in those images, we propose a new framework based on image partition and connected components grouping. Structural analysis is performed from text characters to text strings. First, we choose the candidate text characters from connected components by gradient feature and color feature. Then, character grouping is performed to combine the candidate text characters into text strings which contain at least three character members in alignment. Experiments show that color-based partition performs better than gradient-based partition, but it takes more time to detect text strings on each color layer. The text line grouping is able to extract text strings with arbitrary orientations. The combination of color-based partition and adjacent character grouping (CA) gives the best performance, which outperforms the algorithms presented in ICDAR.

Our future work will focus on developing learning based methods for text extraction from complex backgrounds and text normalization for OCR recognition. We also attempt to improve the efficiency and transplant the algorithms into a navigation system prepared for the wayfinding of visually impaired people.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments and insightful suggestions that improved the quality of this manuscript.

References

[1] J. Banerjee, A. M. Namboodiri, and C. V. Jawahar, "Contextual restoration of severely degraded document images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 517–524.

[2] T. M. Breuel, "The OCRopus open source OCR system," in *Proc. IS&T/SPIE 20th Annu. Symp.*, 2008, pp. 1–15.

[3] T. J. Burns and J. J. Corso, "Robust unsupervised segmentation of degraded document images with topic models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1287–1294.

[4] 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.

[5] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 366–373.

[6] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in nature scenes with stroke width transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2963–2970.

[7] J. Gao and J. Yang, "An adaptive algorithm for text detection from natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 2, pp. 84–89.

[8] Y. M. Y. Hasan and L. J. Karam, "Morphological text extraction from images," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1978–1983, Nov. 2000.

[9] W. T. Ho and Y. H. Tay, "On detecting spatially similar and dissimilar objects using adaboost," in *Proc. Int. Symp. Inf. Technol.*, 2008, pp. 899–903.

[10] R. Jiang, F. Qi, L. Xu, and G. Wu, "A learning-based method to detect and segment text from scene images," *J. Zhejiang Univ.*, vol. 8, pp. 568–574, Apr. 2007.

[11] K. C. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: A survey," *Pattern Recognit.*, vol. 5, pp. 977–997, May 2004.

[12] T. Kasar, J. Kumar, and A. G. Ramakrishnan, "Font and background color independent text binarization," in *Proc. 2nd Int. Workshop Camera-Based Document Anal. Recognit.*, 2007, pp. 3–9.

[13] J. S. Kim, S. H. Kim, H. J. Yang, H. J. Son, and W. P. Kim, "Text extraction for spam-mail image filtering using a text color estimation technique," in *Proc. 20th Int. Conf. Ind., Eng., Other Applic. Appl. Intell. Syst.*, 2007, pp. 105–114.

[14] P. K. Kim, "Automatic text location in complex color images using local color quantization," in *Proc. IEEE TENCON*, 1999, vol. 1, pp. 629–632.

[15] 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.

[16] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Joshi, "Text extraction and document image segmentation using matched wavelets and mrf model," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2117–2128, Aug. 2007.

[17] S. Lefevre and N. Vincent, "Caption localisation invideo sequences by fusion of multiple detectors," in *Proc. 8th Int. Conf. Document Anal. Recognit.*, 2005, pp. 106–110.

[18] J. Liang, D. DeMenthon, and D. Doermann, "Geometric rectification of camera-captured document images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 591–605, 2008.

[19] Q. Liu, C. Jung, and Y. Moon, "Text segmentation based on stroke filter," in *Proc. Int. Conf. Multimedia*, 2006, pp. 129–132.

[20] S. Lu and C. L. Tan, "Retrieval of machine-printed Latin documents through Word Shape Coding," *Pattern Recognit.*, vol. 41, no. 5, pp. 1799–1809, May 2008.

[21] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, 2003, pp. 682–687.

[22] S. M. Lucas, "ICDAR 2005 text locating competition results," in *Proc. Int. Conf. Document Anal. Recognit.*, 2005, vol. 1, pp. 80–84.

[23] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *J. Appl. Stat.*, vol. 21, no. 2, pp. 224–270, 1994.

[24] C. Mancas-Thillou and B. Gosselin, "Color text extraction with selective metric-based clustering," *Comput. Vis. Image Understanding*, vol. 107, pp. 97–107, 2007.

[25] C. Mancas-Thillou and B. Gosselin, "Spatial and color spaces combination for natural scene text extraction," in *Proc. IEEE Conf. Image Process.*, 2006, pp. 985–988.

[26] G. K. Myers, R. C. Bolles, Q. T. Luong, J. A. Herson, and H. B. Aradhye, "Rectification and recognition of text in 3-D scenes," *Int. J. Document Anal. Recognit.*, pp. 147–158, 2004.

[27] N. Nikolaou and N. Papamarkos, "Color reduction for complex document images," *Int. J. Imaging Syst. Technol.*, vol. 19, pp. 14–26, 2009.

[28] W. M. Pan, T. D. Bui, and C. Y. Suen, "Text detection from natural scene images using topographic maps and sparse representations," in *Proc. Int. Conf. Image Process.*, 2009, pp. 2021–2024.

[29] C. J. Park, K. A. Moon, W. G. Oh, and H. M. Choi, "An efficient extraction of character string positions using morphological operator," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2000, pp. 1616–1620.

[30] T. Phan, P. Shivakumara, and C. L. Tan, "A Laplacian method for video text detection," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 66–70.

[31] P. Shivakumara, W. Huang, and C. L. Tan, "An efficient edge based technique for text detection in video frames," in *The Eighth IAPR Workshop on Document Analysis Systems*, 2008.

[32] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Int. Conf. Document Anal. Recognit.*, 2007, pp. 629–633.

[33] K. Sobottka, H. Kronenberg, T. Perroud, and H. Bunke, "Text extraction from colored book and journal covers," in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 1999, no. 4, pp. 163–176.

[34] H. M. Suen and J. F. Wang, "Segmentation of uniform colored text from color graphics background," *VISP*, no. 6, pp. 317–332, Dec. 1997.

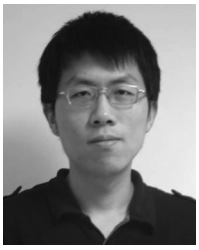
[35] H. Tran, A. Lux, H. L. Nguyen, and A. Boucher, "A novel approach for text detection in images using structural features," in *Proc. 3rd Int. Conf. Adv. Pattern Recognit.*, 2005, pp. 627–635.

[36] J. Weinman, A. Hanson, and A. McCallum, "Sign detection in natural images with conditional random fields," in *Proc. IEEE Int. Workshop Mach. Learning Signal Process.*, 2004, pp. 549–558.

[37] C. Wolf, J. M. Jolion, and F. Chassaing, "Text localization, enhancement and binarization in multimedia documents," in *Proc. Int. Conf. Pattern Recognit.*, 2002, vol. 4, pp. 1037–1040.

[38] J. Zhang and R. Kasturi, "Extraction of text objects in video documents: Recent progress," in *Proc. 8th IAPR Workshop Document Anal. Syst.*, 2008, pp. 5–17.

[39] J. Zhou, L. B. Xiao, R. Dai, and S. Si, "A robust system for text extraction in video," in *Proc. Int. Conf. Mach. Vis.*, 2007, pp. 119–124.



Chucai Yi received the B.S. and M.S. degrees from Huazhong University of Science and Technology, Wuhan, China, in 2007 and 2009, respectively. He is currently working toward the Ph.D. degree in computer science at the Graduate Center, City University of New York, New York.

His research focuses on text detection and recognition in natural scene images. His research interests include object recognition, image processing, and machine learning.

YingLi Tian (M'99–SM'01) received the B.S. and M.S. degrees from TianJin University, Tianjin, China, in 1987 and 1990, respectively, and the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 1996.

After holding a faculty position with the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, she joined Carnegie Mellon University, Pittsburgh, PA, in 1998, where she was a Postdoctoral Fellow with the Robotics Institute. She was then a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY, from 2001 to 2008. She is currently an Associate Professor with the Department of Electrical Engineering, City College of New York, New York. Her current research focuses on a wide range of computer vision problems from motion detection and analysis, to human identification, facial expression analysis, and video surveillance.