Short Papers

Multi-Orientation Scene Text Detection with Adaptive Clustering

Xu-Cheng Yin, *Member, IEEE*, Wei-Yi Pei, Jun Zhang, and Hong-Wei Hao

Abstract—Text detection in natural scene images is an important prerequisite for many content-based image analysis tasks, while most current research efforts only focus on horizontal or near horizontal scene text. In this paper, first we present a unified distance metric learning framework for adaptive hierarchical clustering, which can simultaneously learn similarity weights (to adaptively combine different feature similarities) and the clustering threshold (to automatically determine the number of clusters). Then, we propose an effective multi-orientation scene text detection system, which constructs text candidates by grouping characters based on this adaptive clustering. Our text candidates construction method consists of several sequential coarse-to-fine grouping steps: morphology-based grouping via single-link clustering, orientation-based grouping via divisive hierarchical clustering, and projection-based grouping also via divisive clustering. The effectiveness of our proposed system is evaluated on several public scene text databases, e.g., ICDAR Robust Reading Competition data sets (2011 and 2013), MSRA-TD500 and NEOCR. Specifically, on the multi-orientation text data set MSRA-TD500, the f measure of our system is 71 percent, much better than the state-of-the-art performance. We also construct and release a practical challenging multi-orientation scene text data set (USTB-SV1K), which is available at http://prir.ustb.edu.cn/TexStar/MOMV-text-detection/.

Index Terms—Scene text detection, multi-orientation, adaptive hierarchical clustering, coarse-to-fine grouping

1 Introduction

Effective scene text detection is an important prerequisite for many content-based multimedia understanding applications [1], [2], [3], [4], [5], [6]. In the real world, this task is often challenging due to issues such as complex background and variation of text font, size, and color. Furthermore, text captured in real scenes is always with both multiple orientations and perspective distortions [3] (see scene image examples in Figs. 3 and 4). Complicated by these practical issues, most previous work has focused on (near) horizontal scene text detection [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] and there are only a very few methods proposed for multi-orientation text detection. Yao et al. proposed a multi-orientation scene text detection system by bottom-up grouping and top-bottom pruning but with numerous empirical rules and parameters [17], and then they extended this work to an end-to-end multi-orientation scene text recognition system [18]. Tan's group [19], [20], [21], [22] proposed a number of non-horizontal text detection approaches but specifically designed and experimented on text detection in videos not in natural scenes. Moreover, Yao's text detection approaches [17], [18] have rather limited performance, i.e., the f-measure on the MSRA-TD500 data set is only about 60 percent.

In this paper, we propose a robust multi-orientation scene text detection system with several novel and effective techniques. First, we present an adaptive hierarchical clustering algorithm with a unified distance metric learning framework. This framework has the advantages that both similarity weights (for adaptively combining different feature similarities) and the clustering threshold (for automatically determining the number of clusters) can be simultaneously optimized. Consequently, this approach can automatically select and learn many parameters used in our text detection system.

Second, we construct text candidates by grouping and linking character candidates using the above adaptive clustering algorithm. Specifically, the text candidates construction includes several sequential character grouping steps: Morphology clustering first coarsely groups character candidates with similar appearance; then orientation clustering groups character pairs with consistent orientation; finally, projection clustering finely separates text lines in the same orientation. In such a coarse-to-fine grouping strategy, we can effectively use the key feature, text line alignment in different orientations, to precisely group and construct text candidates with multiple orientations.

Third, based on the above novel techniques, we build an effective multi-orientation scene text detection system. Our proposed system is verified on a variety of public scene text databases, e.g., ICDAR Robust Reading Competition data sets (2011 and 2013), MSRA-TD500 [17], and NEOCR [23]. Specifically, evaluation on the MSRA-TD500 data set shows that our approach with f measure 71 percent, significantly outperforms recent approaches [17], [18] of f measure about 60 percent.

Yet another contribution of this work is that we collect and release a large practical challenging multi-orientation scene text data set (USTB-SV1K), which includes 1,000 street view images of six typical USA cities directly crawled from Google Street View, and each scene image averagely includes 2.96 text regions.

The rest of this paper is organized as follows. Related work is described in Section 2. In Section 3, we present adaptive hierarchical clustering with distance metric learning. Section 4 describes the proposed multi-orientation scene text detection system. Comparative experiments are demonstrated in Section 5. Final remarks are presented in Section 6.

2 Related Work

Existing methods for scene text detection can roughly be categorized into three groups: sliding window based methods [7], [8], [9], [24], connected component based methods [4], [10], [11], [12], [14], [25], and hybrid methods [13]. Sliding window based methods, also known as region-based methods, use a sliding window to search for possible text in the image and then use machine learning techniques to identify text. Connected component based methods extract character candidates from images by connected component analysis followed by grouping character candidates into text; additional checks may be performed to remove false positives. The hybrid method [13] exploits a region detector to detect text candidates and extracts connected components as character candidates by local binarization; non-characters are eliminated with a Conditional Random Fields model, and characters can finally be grouped into text. Recently, Maximally Stable Extremal Regions (MSERs) and Extremal Regions (ERs) based methods, which can be categorized as connected component based methods but using MSERs or ERs as character candidates, have become the focus of several recent works [4], [15], [26], [27], [28]. For the recent impressive scene text detection technology, readers can refer to the work by Yin's group [4], [14],

- *X.-C. Yin is with the Department of Computer Science and Technology and also with the Beijing Key Laboratory of Materials Science Knowledge Engineering, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China. E-mail: xuchengyin@ustb.edu.cn.*
- *W.-Y. Pei and J. Zhang are with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China. E-mail: {peiweiyi.ustb, zj123zyx}@gmail.com.*
- *H.-W. Hao is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: hongwei.hao@ia.ac.cn.*

Manuscript received 28 Apr. 2014; revised 28 Oct. 2014; accepted 22 Dec. 2014. Date of publication 31 Dec. 2014; date of current version 7 Aug. 2015.
Recommended for acceptance by E.G. Learned-Miller.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2014.2388210

[15]. Their technology won the first place of both "Text Localization in Real Scenes" and "Text Localization in Born-Digital Images" in ICDAR 2013 Robust Reading Competition [29].

However, most of existing methods have focused on detecting horizontal or near-horizontal text in natural scenes because of many challenging issues for detecting multi-orientation text. The fundamental difficulty is that the text line alignment (the axis-oriented assumption of a text line) feature can no longer be used to regularize the text construction, while most current clustering- or rule-based methods always rely on such information for character grouping and line construction [4], [12], [13], [16] because the bottom alignment is the key and most stable feature for text lines [4]. Another main challenge is that in arbitrary orientations, it is complicated to determine numerous empirical rules and to train character and text classifiers.

To deal with the above problems, researchers have proposed several, though a very few, non-horizontal text detection methods. Yao et al. [17] proposed a multi-orientation scene text detection system, which uses a two-level classification scheme (bottom-up grouping and top-bottom pruning) based on Stroke Width Transform (SWT) [10]. However, their method involves using numerous features and rules, and empirically selecting a lot of parameters. More recently, Kang et al. used higher order correlation clustering to partition MSERs into text line candidates, and proposed a robust multi-orientation scene text detection system [30]. In our paper, we use adaptive clustering with distance metric learning to automatically select features and determine several main parameters.

At the same time, Tan's group proposed a number of non-horizontal text detection approaches with skeleton concept with Laplacian transform [19], boundary growing with Bayesian classifier [20], [21], pixel growing with quad tree [22], all of which, however, are designed and experimented on text detection in videos. Generally, detecting text in complex natural scenes is another different, big challenging task.

Another related topic is text line segmentation in document images, especially for historical documents [31] and handwriting documents [32], [33], [34]. Conventional text-line segmentation methods can be roughly categorized into four groups [31]: projection-based methods, Hough transform-based methods, image segmentation-based methods, and bottom-up grouping methods. Specifically, bottom-up methods group small units of images (pixels, connected components, characters, and words) into text lines, which can naturally be viewed as a clustering process for aggregating image components according to proximity without the assumption of straight lines [33]. One typical method, Docstrum (Document Spectrum) [35], is based on bottom-up, k-nearest-neighbor clustering of connected components and groups characters into text lines, paragraphs and then the document page. Moreover, clustering-based text line segmentation becomes a recent tendency with state-of-the-art performance in the literature [36]. However, clustering-based text line segmentation approaches (e.g. [33], [34]) can't learn the weights of clustering similarities and the threshold for deciding the number of clusters simultaneously; moreover, many of these approaches use empirical rules and parameters specifically derived from (handwriting) document images, which is time-consuming and error-prone.

Recently, Yin et al. proposed a multi-orientation scene text detection system with two separate components [4], where they first proposed an accurate and robust method for detecting horizontal and near-horizontal text; then they designed a separate step for computing the orientations of text lines; next, they converted non-horizontal text lines into horizontal ones; finally, converted text lines are again fed into their horizontal text detection pipeline. In contrast, we propose a single and whole system with exactly the same steps for detecting both multi-orientation and horizontal text in this paper. Actually there are several distinctions between the proposed technology and our previous method [4]. First, here we propose a general adaptive hierarchical clustering framework with distance metric learning, while the used one in [4] is a specific case. Next, we further propose another pivotal method, coarse-to-fine grouping, which can effectively use the key feature, text line alignment in different orientations, to precisely group and construct text candidates. Then, we design a unified (single and whole) multi-orientation text detection system, while our previous method uses a separate step for computing the orientations of text lines. Finally, the proposed new system outperforms all state-of-the-art technologies on several public multi-orientation text data sets (MSRA-TD500 and USTB-SV1K).

3 Adaptive Hierarchical Clustering

Hierarchical clustering organizes data into a hierarchical structure according to the proximity matrix. The results are usually depicted by a binary tree or dendrogram. The dendrogram can be broken at different levels with a similarity threshold to yield different clusters of the data. Generally, there are two main important issues on hierarchical clustering: (1) How to measure the proximity of data; and (2) how to decide the number of clusters, i.e., how to select the similarity threshold for obtaining the ultimate clustering results. Conventional clustering-based text detection and segmentation methods are time-consuming and error-prone for empirically and manually tuning similarity weights and the clustering threshold [4]. Moreover, previous notable clustering algorithms with metric learning only focus on partitional clustering by learning one certain distance metric, while the number of clusters is taken as an input parameter [37], [38]. In this paper, we present an adaptive hierarchical clustering method with distance metric learning, which deals with both of these two problems together, i.e., learns the weights of metric learning and the threshold for deciding the number of clusters simultaneously.

Note that our proposed general adaptive hierarchical clustering framework here is clearly formulated with three important steps, i.e., sample selection, weight conversion, and model determination. The method in our recent work [4] which uses single-link clustering with distance metric learning for horizontal scene text detection, can be seen as a specific case of this framework. First, our general framework can easily include various hierarchical clustering algorithms, e.g., divisive hierarchical clustering, and single-link clustering, while the method in [4] is specifically designed for single-link clustering. Second, in our new framework, we give a more general representations and descriptions of the two key steps (sample selection and model determination), while the method in [4] only provides one typical representation for each.

Given a set of data patterns $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jn})^T \in \Re^n$ is the feature vector. And the similarity (distance) of two data points, $\mathbf{x}_i$ and $\mathbf{x}_j$, is represented by $d(\mathbf{x}_i, \mathbf{x}_j) > 0$. Hierarchical clustering attempts to construct a tree-like nested structure partition of $\mathbf{X}$.

Suppose there are some prior knowledge for a few data points. On the one hand, some pairwise points are must-link, i.e., each pair of these points is known to be similar enough if such information is explicitly available. We use $\mathcal{S}$ to represent these pairs of must-link points. On the other hand, we use $\mathcal{D}$ to be a set of pairs of cannot-link points, which are dissimilar with explicit information. From the similarity view, if two points are must-link, there will be

$$d(\mathbf{x}_i, \mathbf{x}_j) \leq \epsilon \quad (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}, \tag{1}$$

where ϵ is the similarity threshold. If two points are cannot-link, we will have

$$d(\mathbf{x}_i, \mathbf{x}_j) > \epsilon \quad (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}. \tag{2}$$

The ultimate clustering results (the number of clusters) can be directly obtained by this similarity threshold ϵ, which can be empirically set (in most conventional methods) or automatically learned (in our method here) from these must-link and cannot-link pairwise data points. In the following, we design a distance metric learning framework for automatically learning both the similarity threshold (for deciding the number of clusters) and similarity weights (for adaptively measuring the proximity of data) in the hierarchical clustering algorithm.

3.1 Distance Metric Learning Framework

Consider learning a distance metric of the form as

$$d(\mathbf{x}_i, \mathbf{x}_j; w) = w^T \mathbf{vec}(\mathbf{x}_i, \mathbf{x}_j), \tag{3}$$

where w is the weight vector of metric, and $\mathbf{vec}(\mathbf{x}_i, \mathbf{x}_j)$ is the similarity vector of two variables (points) $\mathbf{x}_i$ and $\mathbf{x}_j$. In our setting, the goal of metric learning is to maximize the distance of point pairs in $\mathcal{D}$ while minimizing the distance of point pairs in $\mathcal{S}$, where $\mathcal{D}$ specifies pairs of points in different clusters (cannot-link pairwise points in Equation (2)) and $\mathcal{S}$ specifies pairs of points in the same cluster (must-link pairwise points in Equation (1)). Moreover, we want to adaptively combine different similarities (e.g., color, stroke width, and compactness in Section 4.2.1) with this metric learning for grouping character candidates, and the combined similarity vector is actually a component-wise distance between two candidates.

We utilize three strategies, i.e., *sample selection*, *weight conversion*, and *model determination*, in this metric learning framework. For *sample selection*, we focus on the hard and representative part of the problem, i.e., given the labeled cluster set $\{C_k\}_{k=1}^m$ (with m clusters), we can use the following strategy to compute $\mathcal{D}$ and $\mathcal{S}$. $\mathcal{D}$ is set to contain the close and representative pairs of points inside and outside a cluster, $\mathcal{S}$ is set to contain the far and representative pairs of points in the same cluster, i.e.,

$$\mathcal{D} = \{(\hat{\mathbf{x}}_k, \hat{\mathbf{y}}_k) = \underset{\mathbf{x}\in C_k, \mathbf{y}\in C_{-k}}{\arg\min}\ d(\mathbf{x}, \mathbf{y}; w)\}_{k=1}^m, \tag{4}$$

$$\mathcal{S} = \{(\mathbf{x}_k^*, \mathbf{y}_k^*) = \underset{\mathbf{x}\in C_k}{\arg\max} \underset{\mathbf{y}\in C_k}{\min} d(\mathbf{x}, \mathbf{y}; w)\}_{k=1}^m, \tag{5}$$

where C_{-k} is the set of points excluding points in the cluster C_k. Suppose ϵ is specified as the hierarchical clustering termination threshold (the similarity threshold), we have the same conclusions of $d(\mathbf{x}, \mathbf{y}; w)$, i.e., Equation (1) for must-link points and (2) for cannot-link points. Obviously, $\mathcal{D}$ and $\mathcal{S}$ are dynamically varied according to the changes of the weights w and the threshold ϵ.

For *weight conversion*, we use the same strategy in [4]. By setting $\theta = [-\epsilon w]^T$, we get $d(\mathbf{x}_i, \mathbf{x}_j; \theta)$. It means that we are able to learn the weights w and the threshold ϵ simultaneously.

For *model determination*, by minimizing the classification error (where the positive and negative sample set correspond to $\mathcal{S}$ and $\mathcal{D}$ respectively), we formulate the learning procedure with

$$\theta^* = \underset{\theta}{\arg\min}\, J(\theta : \mathcal{D}, \mathcal{S}). \tag{6}$$

where $J(\theta : \mathcal{D}, \mathcal{S})$ is the objective function. We select the logistic regression loss as the objective function and use the maximum log-likelihood with gradient ascent algorithm for this typical nonlinear optimization problem, as the same ones in our previous work [4]. Obviously, we can adopt other loss functions of classification [39] for this determination model strategy.

Given the labeled cluster set $\{C_k\}_{k=1}^m$, some initial values for θ have to be specified in order to generate set $\mathcal{D}$ and $\mathcal{S}$ according to Equation (4) and (5). we use a "*self-training distance metric learning*" algorithm [4], an iterative optimization algorithm where each iteration involves two successive steps: first assignments of $\mathcal{D}, \mathcal{S}$ and then optimization (in Equation (6)) with respect to $\mathcal{D}, \mathcal{S}$.

In our general distance metric learning framework, we can construct various adaptive hierarchical clustering algorithms, e.g., single-link clustering, and divisive hierarchical clustering. When using single-link clustering as a hierarchical clustering algorithm, we will get the distance metric learning method for text candidates construction in our previous horizontal scene text detection system [4]. The must-link pairwise points set $\mathcal{S}$ (in Equation (5)) used in [4] is slightly different as $\mathcal{S} = \{(\mathbf{x}_k^*, \mathbf{y}_k^*) = \arg\min_{\mathbf{x}\in C_k^1, \mathbf{y}\in C_k^2} d(\mathbf{x}, \mathbf{y}; w)\}_{k=1}^m$, where C_k^1 and C_k^2 are direct subclusters of C_k.

4 Multi-Orientation Text Detection

4.1 System Overview

In the previous work, we proposed an efficient MSER-based algorithm for scene text detection [4]. The infrastructure of this method provides the flexibility for incorporating techniques addressing multi-orientation cases. In this paper, based on the generic adaptive clustering method, we extend this framework by modifying the text candidates construction stage, and propose a unified multi-orientation text detection system, thereby enabling the effective detection of both multi-orientation and perspective-distortion scene text. We list the stages of our multi-orientation scene text detection system in the following:

1) *Character candidates extraction*. Character candidates are extracted using the MSERs algorithm; most of the repeating components are removed with a MSERs pruning algorithm by minimizing regularized variations.
2) *Text candidates construction*. Text candidates are constructed using three sequential coarse-to-fine character grouping steps, i.e., morphology clustering, orientation clustering and projection clustering. More details are presented in Section 4.2.
3) *Text candidates elimination*. The posterior probabilities of text candidates corresponding to non-text are estimated using the character classifier and text candidates with high non-text probabilities are removed.
4) *Text candidates classification*. Text candidates corresponding to true text are identified by the text classifier, which is trained to decide whether a text candidate corresponding to the true text or not.

Here, we mainly focus on the key novel part of our system, i.e., text candidates construction with adaptive clustering. The other three stages can be referred to [4], while the character and text classifiers are AdaBoost classifiers trained on horizontal and multi-orientation text samples (from ICDAR 2011 to MSRA-TD500 and to USTB-SV1K training sets) in our new system.

4.2 Text Candidates Construction

In multi-orientation text detection, the key issue is how to group character candidates into text candidates, i.e., text candidates construction. In [17], the authors first grouped character candidates into character pairs with clustering (candidate linking), then constructed text candidates by grouping these character pairs with classification (chain analysis). However, their method involves using numerous features and rules, and empirically selecting a lot of parameters for clustering and classification.

Alternatively, in our system, text candidates are constructed by three sequential coarse-to-fine grouping steps with adaptive clustering and several parameters are learned by distance metric learning (see Fig. 1): *morphology clustering*, *orientation clustering* and *projection clustering*. Morphology clustering first coarsely groups character candidates with similar appearance together; then orientation clustering refines to group *character pairs* with

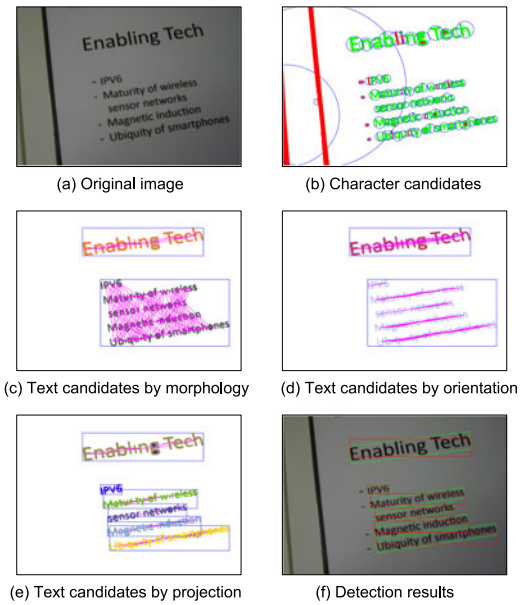

Fig. 1. Flowchart of text candidates construction: (a) the original image; (b) character candidates extracted by MSER algorithm, where the green ones are characters marked "is-character", the red ones are characters marked "non-character", and the blue circles are their circumcircles; (c) text candidates constructed by morphology clustering, where the ligature between two characters showing that they consist of one pair, the blue rectangles are the bounding rectangles of each group; (d) text candidates after orientation clustering, where pairs whose orientations have a huge difference with orientation clusters are dropped; (e) text candidates constructed after projection clustering, where the blue rectangles are the bounding rectangles of each text candidate; and (f) the final detection results, where the green bounding box is the oriented text line detected, while the red box is the deskewed result.

consistent orientation together; finally projection clustering finely separates text lines in the same orientation. Moreover, most existing multi-orientation text detection methods only use orientation knowledge while scene text is always with multiple orientations and perspective distortions. Our sequential grouping framework can deal with both multi-orientation and multi-view text by using a variety of orientation and view-based features and classifiers.

1) *Morphology clustering* (morphology-based grouping via clustering). By the character appearances (color, stroke width and location differences), character candidates are clustered into N_{mor} groups $\{G_i^{\text{mor}}\}_{i=1}^{N_{\text{mor}}}$ using single-link clustering (see Section 4.2.1).
2) *Orientation clustering* (orientation-based grouping via clustering). By the *character pair* orientation, *character pairs* from each group G_i^{mor} are then clustered into N_{ori} groups $\{G_i^{\text{ori}}\}_{i=1}^{N_{\text{ori}}}$. More details are presented in Section 4.2.2.
3) *Projection clustering* (projection-based grouping via clustering). By the character pair intercept (projection of orientation vectors), pairs in each group G_i^{ori} finally clustered into N_{pro} groups $\{G_i^{\text{pro}}\}_{i=1}^{N_{\text{pro}}}$; each group G_i^{pro} corresponds to a text candidate (see Section 4.2.3).

4.2.1 Morphology Clustering

A set of character candidates are considered similar if their morphology appearance, i.e., *color*, *stroke width* and *compactness*, are similar; similar character candidates are grouped together using the single-link clustering algorithm (Fig. 1c), in which parameters of the clustering algorithm are learned automatically using the learning algorithm presented in Section 3. Specifically, given two character candidates u and v, let $\mathbf{x}_{u,v} = (d_{clr}(u.v), d_{swd}(u,v), d_{\text{comp}}(u,v))^T$, and the color, stroke width, and compactness similarities are defined as

- Color difference

$$d_{\text{clr}}(u,v) = \frac{\sqrt{(c1_u - c1_v)^2 + (c2_u - c2_v)^2 + (c3_u - c3_v)^2}}{255}, \quad (7)$$

- Stroke width difference

$$d_{\text{swd}}(u,v) = abs(s_u - s_v)/\max(s_u, s_v), \quad (8)$$

- Compactness (location difference)

$$d_{\text{comp}}(u,v) = \frac{\sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}}{(r_u + r_v)/2}, \quad (9)$$

where (x_u, y_u) is the centroid of u's circumcircle, r_u the radius of its circumcircle, s_u the stroke width (computed in [14]) of u, h_u and w_u the height and width of the bounding rectangle of u, and $c1_u, c2_u, c3_u$ the average three channel color values of u. In order to deal with multi-orientation cases, we design a new feature, "compactness (location difference)" of character candidates compared to our previous method [4].

In morphology clustering, for two character candidates in one cluster, there is $d(u,v;w) = w^T\mathbf{x}_{uv} \leq \epsilon$, where w is the the feature weight vector and ϵ is the clustering threshold; for different clusters, there will be $d(u,v;w) = w^T\mathbf{x}_{uv} > \epsilon$. We use the above adaptive agglomerative hierarchical clustering (single-link) algorithm with distance metric learning (in Section 3). Finally, character candidates are clustered into N_{mor} partitions $\{G_i^{\text{mor}}\}_{i=1}^{N_{\text{mor}}}$.

We use the same strategies (e.g., training and running) as the clustering approach in our prior work. More details can be referred to [4]. Here three features (color, stroke width, and compactness) have been used, thus we are not able to separate characters in different text lines but with a similar morphology. Consequently, text lines in the same group are sequentially separated using orientation clustering and projection clustering.

4.2.2 Orientation Clustering

Given a morphology clustered group G_i^{mor}, the orientation clustering algorithm clusters *character pairs* $\{P_k\}_{k=1}^{N_{\text{cp}}}$ in G_i^{mor} into groups according to their orientations, where $\{P_k\}_{k=1}^{N_{\text{cp}}} = \{(u,v)|d(u,v;w) \leq \epsilon, u, v \in G_i^{\text{mor}}\}$; character pairs with a consistent orientation or perspective view will be clustered into the same group. The orientation $o_{u,v}$ of a character pair (u,v) is defined as

$$o_{u,v} = \arctan\left(\frac{y_u - y_v}{x_u - x_v}\right). \quad (10)$$

As the number of pairs in each group can be very large (one character always belonging to many pairs) and we do not generate a complete hierarchy all the way down to individual character candidates, we utilize a fast clustering method, a divisive hierarchical clustering approach, named as *divisive binary clustering*. This is a "top down" approach, which starts with the complete data set and hierarchically divides it into partitions.

Let $S = \{x_1, x_2, ..., x_N\}$ be the set (a set of scalars, e.g., orientations of character pairs in Equation (10)) to be clustered, we compute the number of elements in $[a,b]$ (the current *interval*) $n_{a,b} = |\{x|a \leq x \leq b\}|$, *mean* $\mu_{a,b} = \frac{1}{n_{a,b}}\sum_{i=1}^{n_{a,b}} x_i$, and *central moment* $\mu_{a,b}^1 = \frac{1}{n_{a,b}}\sum_{i=1}^{n_{a,b}} |x_i - \mu_{a,b}|$. We argue that each consistent partition after grouping includes character pairs which should fall into a "narrow" space (represented with the partition's *interval*) with a

```
1: procedure CLUSTER-SET(S, [a, b], w_1, w_2, ε*, ε_3)
2:     if w_1 μ^1_{a,b} + w_2|b − a| > ε* then
3:         if n_{a,b}/N < ε_3 then
4:             return ∅
5:         end if
6:         U_1 = CLUSTER-SET(S, [a, (a + b)/2], w_1, w_2, ε*, ε_3)
7:         U_2 = CLUSTER-SET(S, ((a + b)/2, b], w_1, w_2, ε*, ε_3)
8:         return U_1 ∪ U_2
9:     else
10:        return U = U ∪ {μ_{a,b}}
11:    end if
12: end procedure
```

Fig. 2. The divisive binary clustering algorithm.

distance "compactness" (represented with *central moment*). Here, one data set is divided into two (*binary*) partitions with the same interval size $[a,(a+b)/2]$ and $((a+b)/2,b]$ when $\mu^1_{a,b}$ is greater than ϵ_1 or $|b-a|$ is greater than ϵ_2. Moreover, we will drop the set in which $n_{a,b}/N \leq \epsilon_3$. The algorithm returns $\mu_{a,b}$ of each cluster. Finally we can get N_{c} clusters with *mean* elements

$$\{C_i^{\mathrm{c}}\}_{i=1}^{N_{\mathrm{c}}} = \{\mu_{a_i,b_i} | \mu^1_{a_i,b_i} \leq \epsilon_1, |b_i - a_i| \leq \epsilon_2, n_{a_i,b_i}/N > \epsilon_3\}. \quad (11)$$

Obviously and interestingly, $\mu^1_{a_i,b_i}$ and $|b_i - a_i|$ are in the same scale. As a result, we suppose $\epsilon_1 = \epsilon_2 = \epsilon^*$, and get the following similarity (with two weights w_1 and w_2) for character pairs in one cluster,

$$d(\mu^1_{a_i,b_i}, |b_i - a_i|; w_1, w_2) = w_1 \mu^1_{a_i,b_i} + w_2 |b_i - a_i| \leq \epsilon^*. \quad (12)$$

In two different clusters, there is $w_1 \mu^1_{a_i,b_i} + w_2|b_i - a_i| > \epsilon^*$. Similarly, we use the distance metric learning framework (in Section 3) for learning these parameters (the weights w_1 and w_2 of clustering similarities and the threshold ϵ^* for deciding the number of clusters).

We summarize this divisive binary clustering method in Fig. 2 with the pseudo-code algorithm. The input of the algorithm includes the set of scalars (S) to be clustered and the given *interval* $[a,b]$, the learned weights (w_1, w_2) and threshold (ϵ^*) for clustering, and the threshold ϵ_3. If the condition for dividing is satisfied, i.e., $w_1 \mu^1_{a,b} + w_2|b-a| > \epsilon^*$ in Fig. 2, the binary divisive procedure is recursively conducted. Finally, this algorithm returns a set of scalars each of which is the *mean* of one resulting cluster. In orientation clustering of the experimental system, the orientation range and ϵ_3 are empirically set with $[-90, 90]$ and 0.26 respectively.

Then, we propose the concept of "misregistration" to verify the consistence of orientations. After investigating multi-orientation text samples, we find several empirical observation results. For one orientation, a pair of two similar characters with a fair distance (i.e., about the average character length), resulting in a fair compactness difference (d_{comp} in Equation (9)), can be included in this orientation's group with a high probability (by a large "misregistration" value); in contrast, a pair with two different and distant characters (very large compactness) will be excluded with a fairly high probability. Note that for a pair with two highly close characters (very small compactness) will also be probably excluded because a little change of location for the two characters can bring to a large orientation difference.

Consequently, we construct the misregistration measure with a Gaussian Mixture Model (GMM), where we use a mixture of three Gaussians each of which corresponds to one situation above (small, fair or large compactness respectively). That is, for a pair (u,v) in each morphology cluster G_i^{mor}, we specifically design the misregistration measure as

$$misreg(u,v) = \beta \sum_{j=1}^{3} \alpha_j N_j(d_{\mathrm{comp}}(u,v), \mu_j, \delta_j), \quad (13)$$

where $\sum_{j=1}^{3} \alpha_j = 1$, $d_{\mathrm{comp}}(u,v)$ is the compactness similarity (location difference) between u and v (Equation (9)), μ_j and δ_j are the mean and variance of the Gaussian distribution (single normal distribution) N_j, and β is the adjust parameter for scaling with the orientation value with unit. After empirically setting β on the training set, Equation (13) is a simple and typical GMM easily solved with Expectation-Maximization (EM) algorithm [39], where we manually label $misreg(u,v)$ values with $d_{\mathrm{comp}}(u,v)$ for pairs of character candidates on the training set. Interestingly, this "misregistration" physically measures the variance of clusters, then it can be similarly measured using mean square error as in K-Means clustering.

After the above clustering (in Fig. 2), we get N_{c} mean orientations ($\{C_k^{\mathrm{c}}\}_{k=1}^{N_{\mathrm{c}}}$ in Equation (11)). Now, for each orientation C_k^{c}, we will check all character pairs in G_i^{mor}, and construct a group (called as an *orientation cluster*) by selecting pairs the orientation of which is satisfied with

$$o_{u,v} \in (C_k^{\mathrm{c}} - misreg(u,v), C_k^{\mathrm{c}} + misreg(u,v)). \quad (14)$$

Finally after dealing with all morphology clusters $\{G_i^{\mathrm{mor}}\}_{i=1}^{N_{\mathrm{mor}}}$, we get N_{ori} *orientation clusters* $\{G_i^{\mathrm{ori}}\}_{i=1}^{N_{\mathrm{ori}}}$, where $N_{\mathrm{ori}} = N_{\mathrm{c}}$.

4.2.3 Projection Clustering

After the *orientation clustering* stage, pairs in each orientation cluster G_i^{ori} may have a similar orientation but in different text lines. Consequently, we use projection clustering to separate text lines in the same orientation. This procedure is essentially the same as orientation clustering. First, a modified intercept (projection of orientation vectors onto a line perpendicular to the orientation) for one pair (u,v) in G_i^{ori} is defined as

$$b_{u,v} = (y_u - x_u \cdot \tan(C_i^{\mathrm{c}})) \cdot \cos(C_i^{\mathrm{c}}), \quad (15)$$

where C_i^{c} (in Equation (11)) is the representative orientation of G_i^{ori}. Then, we get N_{ct} clusters of *mean* intercepts $\{C_k^{\mathrm{ct}}\}_{k=1}^{N_{\mathrm{ct}}}$ by using the same divisive binary clustering to cluster intercepts of pairs in each group G_i^{ori} with parameters $(\{b_{u,v}\}, [avg(b) - 10avg(r), avg(b) + 10avg(r)], w_1', w_2', \epsilon'^*, 0.05)$, where $avg(b)$ and $avg(r)$ are the average intercept of pairs and the average circumcircle radius of characters in each group respectively, and w_1', w_2', ϵ'^* can be similarly learned as in Equation (12). Next, we use the misregistration ($misreg'(u,v)$) not with orientation but with *intercept* to construct projection clusters in each orientation cluster G_i^{ori} by verifying

$$b_{u,v} \in (C_k^{\mathrm{ct}} - misreg'(u,v), C_k^{\mathrm{ct}} + misreg'(u,v)). \quad (16)$$

Finally after dealing with all orientation clusters $\{G_i^{\mathrm{ori}}\}_{i=1}^{N_{\mathrm{ori}}}$, we will get N_{pro} *projection clusters* $\{G_i^{\mathrm{pro}}\}_{i=1}^{N_{\mathrm{pro}}}$, each of which corresponds to a text candidate.

5 Experiments

All experimental results of our system are performed on a Linux laptop with a 2.20 GHz processor. Please note that without specification, multi-orientation text detection is evaluated with the protocol proposed by Yao et al. [17], and horizontal text detection is measured as the same as ICDAR 2011 Competition [40].

5.1 USTB-SV1K Database

In SVT database [41], the scene images are harvested and copied directly from the computer browser and annotated by *workers* in Amazon's Mechanical Turk, where the *workers* are asked to select and annotate some specific images with specific text (business signs and names) only in the horizontal orientation, and a part of words appeared in the images are not annotated. For

TABLE 1
Performance Comparison on MSRA-TD500 Database

Method	Recall	Precision	f	Speed
Our method	0.63	0.81	0.71	1.4
Yin et al.'s method [4]	0.61	0.71	0.66	0.8
Yao et al.'s method [17]	0.63	0.63	0.60	7.2
Yao et al.'s recent method [18]	0.62	0.64	0.61	3.5
Kang et al.'s method [30]	0.62	0.71	0.66	–

MSRA-TD500 data set [17], a typical multi-orientation text data set, the images are mostly clean without blurry text. However, scene text images captured in general situations are always with low-quality. Another representative multi-orientation text data set is NEOCR [23] which contains scene images captured in real-application situations with much rich annotations. However, about 20 percent text regions contain multiple (two or more) text lines. Consequently, the SVT, MSRA-TD500, and NEOCR databases are rather difficult to be used in general and open scene text detection and recognition systems.

In contrast, we collect and construct a new, more practical and challenging multi-orientation natural scene text data set (**USTB-SV1K**)[1], images of which are directly crawled from Google Street View.

Data collection. For each GPS location, the 360-angle-view image in Google Street View is actually composed of 91 patch images with 512×512 size. We decode the web link from Google, and automatically download all 91 patches which are captured by *video cameras* with multi-orientation and perspective-distortion text. We manually select patches with text which can be seen by person.

Data set description. We annotate an image in which a list of words to label with bounding boxes by the coordinates of the left-top point, width, height and inclination angle along with the ground truth word, which is similar to MSRA-TD500. We collect 1,000 (500 for training and 500 for testing) street view (patch) images from six USA cities, i.e., New York, Boston, Los Angle, Washington DC, San Francisco, and Seattle. The set from each city includes about $160 - 180$ images, about half of which are for training, and the rest for testing. The whole data set includes 2,955 text regions, and the mean and standard deviation of the number of text regions per image are 2.96 and 2.08 respectively. Summarily, there are three main challenges for detection and recognition on this data set (see samples in Fig. 4). First, in many cases, text is with multiple orientations and perspective distortions. About 75, 10 and 15 percent images are with (near) horizontal, multi-orientation, and perspective-distortion (always with skewed distortions) text respectively. Second, this data set includes a lot of small or blurred text (about 28 percent), where a small character means the height of the character is less than 15 pixels. Third, about one fourth of text regions are specific street and business names, or parts of words, and can't be found in a common dictionary. Overall, our data set, USTB-SV1K, has a general open and challenging situation for natural (street view) scene text detection and recognition.

5.2 Experiments with Multi-Orientation Text

The MSRA-TD500 database[2] is a multi-orientation database with 500 images. The average size of pictures is $1{,}600 \times 1{,}200$. Same as [17], we use 300 images to do training, and the rest to test. We compare our proposed method to four state-of-the-art methods: Yao et al.'s method [17] and their recent one [18], Kang et al.'s method [30], and Yin et al. method [4] (our previous method).

1. The USTB-SV1K data set is available at http://prir.ustb.edu.cn/TexStar/MOMV-text-detection/.

2. This data set is available at http://www.iapr-tc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_(MSRA-TD500).

Fig. 3. Experimental results of samples on MSRA-TD500. Our method can handle most multi-orientation scenes. But for curve text (the last image in second row) and short similar multi-line text (the last image in third row), the orientation clustering may fail.

First, as can be seen from Table 1, the overall performance of our method is much better than all other methods. Specifically, compared to Yao et al.'s methods, our method largely improve the precision and f-measure by about 0.2 and 0.1 respectively. Unsuccessful detection (see Fig. 3) may be due to the following reasons: (1) The text candidates construction is unable to handle some short, similar and sparse multi-line text scenarios where characters in multiple text lines have both a close distance and a very similar structure, and can be easily grouped as some vertical lines; and (2) our proposed method currently can not deal with cursive text.

Second, our system is fast as shown in Table 1, where speed is represented with seconds. It has a similar speed to Yin et al.'s method, but runs almost five times faster than Yao et al.'s method [17], which is profiled on a machine with a 2.53 GHz processor. Our method is also much faster than Yao et al.'s recent method [18].

We then evaluate our technology on USTB-SV1K, a new challenging multi-orientation scene text data set, and comparative results (on the testing set) are shown in Table 2. Our new approach outperforms Yin et al.'s method [4] and Yao et al.'s recent method [18]. Some examples are shown in Fig. 4. Our method can successfully detect similar multiple text lines with a skewed distortion, which is challenging for Yin et al.'s. It is worth noting that a significant part (75 percent) of USTB-SV1K samples contain (near)

Fig. 4. Experimental results of samples of our method on USTB-SV1K. Detecting blurred, very small, or seriously distorted text is challenging (in the last row).

TABLE 2
Performance (%) Comparison on USTB-SV1K Data Set

Method	Recall	Precision	f
Our method	45.41	49.85	47.53
Yin et al.'s method [4]	45.18	45.00	45.09
Yao et al.'s recent method [18]	44.05	45.80	44.91

horizontal text, which is preferable for Yin et al.'s method as their original system targeted at specific horizontal text detection. Yet, our method (also including Yin et al.'s method or the approach in [18]) meets some challenges for detecting rather blurred, very small and highly perspective distorted scene text for this real challenging multi-orientation scene text data set (see discussions in Section 5.1). As a result, all comparative methods on USTB-SV1K data set in Table 2 have much lower performance compared to MSRA-TD500 data set in Table 1.

We also evaluate our proposed system on another challenging data set NEOCR [23][3]. The NEOCR data set contains a total of 659 images with 5,238 text regions with multiple orientations and perspective distortions, however 973 of which (about 20 percent text regions) are annotated with multiple (two or more) text lines. It is irrational and even impossible to measure the text detection performance on such multi-text-line regions, since all current text detection measures (on ICDAR and MSRA-TD500 data sets) are based on one-word position or one-line-region position. It may be incomplete to compare with different methods (e.g., [4] and [18]) on this data set. Nevertheless, we still perform experiments on the whole NECOR data set of our system. The recall, precision, and f-score are 25.38, 40.72 and 31.27 percent respectively. Challenges of NEOCR are mainly from seriously perspective or wrapping distorted text, and close illumination distributions between text and background.

5.3 Experiments with Horizontal Text Databases

In order to investigate the robustness of our method, we also evaluate our multi-orientation scene text detection system on notable horizontal and near-horizontal databases, i.e., ICDAR 2011 and 2013 Robust Reading Competition (Challenge 2: Reading Text in Scene Images) databases.[4] In these experiments, in order to measure the performance of our system, text candidates identified as text by our system are further partitioned into words by classifying inner character distances into character spacings and word spacings using an AdaBoost classifier [14]. Note that for the ICDAR 2013 data set, we use the ICDAR 2013 Competition online website to evaluate our system.

Table 3 shows the performance of our multi-orientation text detection system, one Neumann and Matas' method ($\text{NM}_{iccv2013}$ [16]), Shi et al.'s method [27]) and the top three scoring methods (Kim's method, Yi's method, and TH-TextLoc system) from ICDAR 2011 Competition [40]. As can be seen from Table 3, our method, which is designed for text detection in arbitrary orientations, produces better overall performance over all other methods which are all specifically designed for horizontal text detection. Moreover, our system offers speed advantage over some of the listed methods. The average processing speed of the proposed system is 0.33 s per image. The speed of Shi et al.'s method [27] on a PC with a 2.33 GHZ processor is 1.5 s per image. The speed of $\text{NM}_{iccv2013}$ (including text recognition) are 35 s [16] per image on a "standard PC".

Table 4 shows the performance of our multi-orientation text detection system, the top three scoring methods (USTB_TexStar, TextSpotter, and CASIA_NLPR) from ICDAR 2013 Competition [29], where "USTB_TexStar" is our previous method same as $\text{Yin}_{pami2014}$ [4] (horizontal). As can be seen from Table 4, our method, which is generally designed for detecting text in arbitrary orientations, has a competitive performance with the top three winning methods which are all specifically designed for horizontal text detection.

We also compare our new method with our previous method ($\text{Yin}_{pami2014}$) [4]. In Tables 3 and 4, the "multi-oriented" is the multi-orientation scene detection system of our previous method, and the "horizontal" method can only detect the scene text in the horizontal direction. Note that the base system ("horizontal") of our previous multi-orientation text detection method ("multi-oriented") is targeted at specific horizontal text detection. As shown in Tables 3 and 4, our new multi-orientation scene text detection system is even slightly better than our previous multi-orientation system. These experimental results also verify the effectiveness of our new proposed system.

TABLE 3
Performance (%) Comparison on ICDAR 2011 Set

Method	Recall	Precision	f
Our method	66.01	83.77	73.84
$\text{NM}_{iccv2013}$ [16]	66.4	79.3	72.3
Shi et al.'s method [27]	63.1	83.3	71.8
Kim's method	62.47	82.98	71.28
Yi's method	58.09	67.22	62.32
TH-TextLoc System	57.68	66.97	61.98
$\text{Yin}_{pami2014}$ [4] (multi-oriented)	66.63	82.00	73.52
$\text{Yin}_{pami2014}$ [4] (horizontal)	68.26	86.29	76.22

6 Conclusion

This paper presents a new and robust multi-orientation scene text detection method with adaptive clustering. Summarily, there are three main underlying principles involved in our approach. The first one is the adaptive hierarchical clustering algorithm. In our text candidates construction stage, we use an adaptive hierarchical clustering method with distance metric learning, which automatically learns similarity weights (to adaptively combine different feature similarities) and the clustering threshold (to automatically determine the number of clusters). The second underlying principle is the coarse-to-fine grouping algorithm (morphology clustering, orientation clustering and projection clustering) for text candidates construction, which can effectively use the key feature, text line alignment in different orientations, to group text candidates. The third one is the efficient scene text detection infrastructure. Extensive experiments show that our proposed multi-orientation text detection system exhibits superior performance on a variety of public databases. Moreover, we also collect and release a practical challenging multi-orientation and perspective-distortion scene text data set (USTB-SV1K). One near future issue is how to detect highly blurred texts in low-resolution images, as there are a lot of blurred text regions in real scene images (e.g., street views in USTB-SV1K). Another issue is to extend our proposed method to

TABLE 4
Performance (%) Comparison on ICDAR 2013 Set

Method	Recall	Precision	f
Our method	65.11	83.98	73.35
USTB_TexStar	66.45	88.47	75.89
TextSpotter	64.84	87.51	74.49
CASIA_NLPR	68.24	78.89	73.18
$\text{Yin}_{pami2014}$ [4] (multi-oriented)	64.80	82.36	72.53
$\text{Yin}_{pami2014}$ [4] (horizontal)	66.45	88.47	75.89

3. NEOCR is available at http://www.iapr-tc11.org/mediawiki/index.php/NEOCR:_Natural_Environment_OCR_Dataset.

4. The ICDAR 2011 and 2013 Robust Reading Competition data sets are available at http://robustreading.opendfki.de/wiki/SceneText and http://dag.cvc.uab.es/icdar2013competition/?ch=2, respectively.

detect both multi-orientation and cursive text in scene images and construct an end-to-end scene text recognition system.

ACKNOWLEDGMENTS

The authors are grateful to Xuwang Yin, Prof. Kaizhu Huang, Dr. Jacqueline Feild and Chia-Jung Lee for helpful discussions, and to the anonymous reviewers for their constructive comments. The authors would like to thank Prof. Xiang Bai for providing the results of their method [18] on USTB-SV1K. The research is partly supported by National Natural Science Foundation of China (61105018, 61175020, 61473036). X.-C. Yin is the corresponding author.

REFERENCES

[1] J. J. Weinman, E. Learned-Miller, and A. Hanson, "Scene text recognition using similarity and a lexicon with sparse belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1733–1746, Oct. 2009.

[2] J. J. Weinman, Z. Butler, D. Knoll, and J. Feild, "Toward integrated scene text reading," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 375–387, Feb. 2014.

[3] X.-C. Yin, H.-W. Hao, J. Sun, and S. Naoi, "Robust vanishing point detection for MobileCam-based documents," in *Proc. Int. Conf. Document Anal. Recognit.*, 2011, pp. 136–140.

[4] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, pp. 970–983, May 2014.

[5] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 4321–4328.

[6] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014. DOI: 10.1109/TPAMI.2014.2366765

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

[8] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, "Adaboost for text detection in natural scene," in *Proc. Int. Conf. Document Anal. Recognit.*, 2011, pp. 429 –434.

[9] K. Kim, K. Jung, and J. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1631–1639, Dec. 2003.

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

[11] C. Yi and Y. Tian, "Localizing text in scene images by boundary clustering, stroke segmentation, and string fragment classification," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4256–4268, Sep. 2012.

[12] C. Yi and Y. Tian, "Text extraction from scene images by character appearance and structure modeling," *Comput. Vis. Image Understanding*, vol. 117, no. 2, pp. 182–194, 2013.

[13] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 800–813, Mar. 2011.

[14] X. Yin, X.-C. Yin, H.-W. Hao, and K. Iqbal, "Effective text localization in natural scene images with MSER, geometry-based grouping and AdaBoost," in *Proc. Int. Conf. Pattern Recognit.*, 2012, pp. 725–728.

[15] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Accurate and robust text detection: A step-in for text retrieval in natural scene images," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2013, pp. 1091–1092.

[16] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 97–104.

[17] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1083–1090.

[18] C. Yao, X. Bai, and W. Liu, "A unified framework for multi-oriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, Nov. 2014.

[19] P. Shivakumara, Q. P. Trung, and C. L. Tan, "A Laplacian approach to multi-oriented text detection in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 412–419, Feb. 2011.

[20] P. Shivakumara, R. P. Sreedhar, T. Q. Phan, S. Lu, and C. L. Tan, "Multioriented video scene text detection through Bayesian classification and boundary growing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 8, pp. 1227–1235, Aug. 2012.

[21] P. Shivakumara, T. Q. Phan, S. Lu, and C. L. Tan, "Gradient vector flow and grouping-based method for arbitrarily oriented scene text detection in video images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1729–1739, Oct. 2013.

[22] P. Shivakumara, H. Basavaraju, D. Guru, and C. L. Tan, "Detection of curved text in video: Quad tree based method," in *Proc. Int. Conf. Document Anal. Recognit.*, 2013, pp. 594–598.

[23] R. Nagy, A. Dicker, and K. Meyer-Wegener, "NEOCR: A configurable data set for natural image text recognition," in *Proc. Int. Workshop Camera-Based Document Anal. Recognit.*, 2011, pp. 53–58.

[24] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. Int. Conf. Pattern Recognit.*, 2012, pp. 3304–3308.

[25] T. Q. Phan, P. Shivakumara, and C. L. Tan, "Detecting text in the real world," in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 765–768.

[26] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3538–3545.

[27] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao, "Scene text detection using graph model built upon maximally stable extremal regions," *Pattern Recognit. Lett.*, vol. 34, no. 2, pp. 107–116, 2013.

[28] H. Koo and D. Kim, "Scene text detection via connected component clustering and nontext filtering," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2296–2305, Jun. 2013.

[29] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, "ICDAR 2013 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2013, pp. 1115–1124.

[30] L. Kang, Y. Li, and D. Doermann, "Orientation robust text line detection in natural images," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 4034–4041.

[31] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: A survey," *Int. J. Document Anal. Recognit.*, vol. 9, no. 2-4, pp. 123–138, Apr. 2007.

[32] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger, "Script-independent text line segmentation in freestyle handwritten documents," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 8, pp. 1313–1329, Aug. 2008.

[33] F. Yin and C.-L. Liu, "Handwritten Chinese text line segmentation by clustering with distance metric learning," *Pattern Recognit.*, vol. 42, no. 12, pp. 3146–3157, 2009.

[34] H. Koo and N. Cho, "Text-line extraction in handwritten Chinese documents based on an energy minimization framework," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1169–1175, Mar. 2012.

[35] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1162–1173, Nov. 1993.

[36] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "ICDAR2013 handwriting segmentation contest," in *Proc. Int. Conf. Document Anal. Recognit.*, 2013, pp. 1434–1438.

[37] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Proc. Int. Conf. Neural Inform. Process. Syst.*, 2002, pp. 505–512.

[38] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 11–18.

[39] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[40] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," in *Proc. Int. Conf. Document Anal. Recognit.*, 2011, pp. 1491–1496.

[41] K. Wang and S. Belongie, "Word spotting in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 591–604.