Orientation Robust Text Line Detection in Natural Images

Le Kang[1], Yi Li[2], and David Doermann [1]
[1]University of Maryland, College Park, MD, USA
[2]NICTA and ANU, Canberra, Australia
[1]{lekang,doermann}@umiacs.umd.edu [2]yi.li@cecs.anu.edu.au

Abstract

In this paper, higher-order correlation clustering (HOCC) is used for text line detection in natural images. We treat text line detection as a graph partitioning problem, where each vertex is represented by a Maximally Stable Extremal Region (MSER). First, weak hypotheses are proposed by coarsely grouping MSERs based on their spatial alignment and appearance consistency. Then, higher-order correlation clustering (HOCC) is used to partition the MSERs into text line candidates, using the hypotheses as soft constraints to enforce long range interactions. We further propose a regularization method to solve the Semidefinite Programming problem in the inference. Finally we use a simple texton-based texture classifier to filter out the non-text areas. This framework allows us to naturally handle multiple orientations, languages and fonts. Experiments show that our approach achieves competitive performance compared to the state of the art.

1. Introduction

Text in natural images carries important semantic information. Since localizing text aids scene understanding, the text detection problem was studied extensively in recent work [1, 2, 3, 4, 5, 6]. The problem is also relevant to a number of computer vision applications such as internet image indexing, mobile vision and low vision aids.

Generally, text lines in natural images are curvilinear and diversified with different orientations, fonts, sizes, and scripts. However, most of the current methods focus on building models for certain range of fonts and scripts, such as detection-by-recognition approaches [2, 3, 6]. The bounding boxes of the areas for potential character regions are detected and classified, and text line structures are enforced to heuristically link bounding boxes together. These kinds of approaches may not be easily adapted to multi-orientation cases.

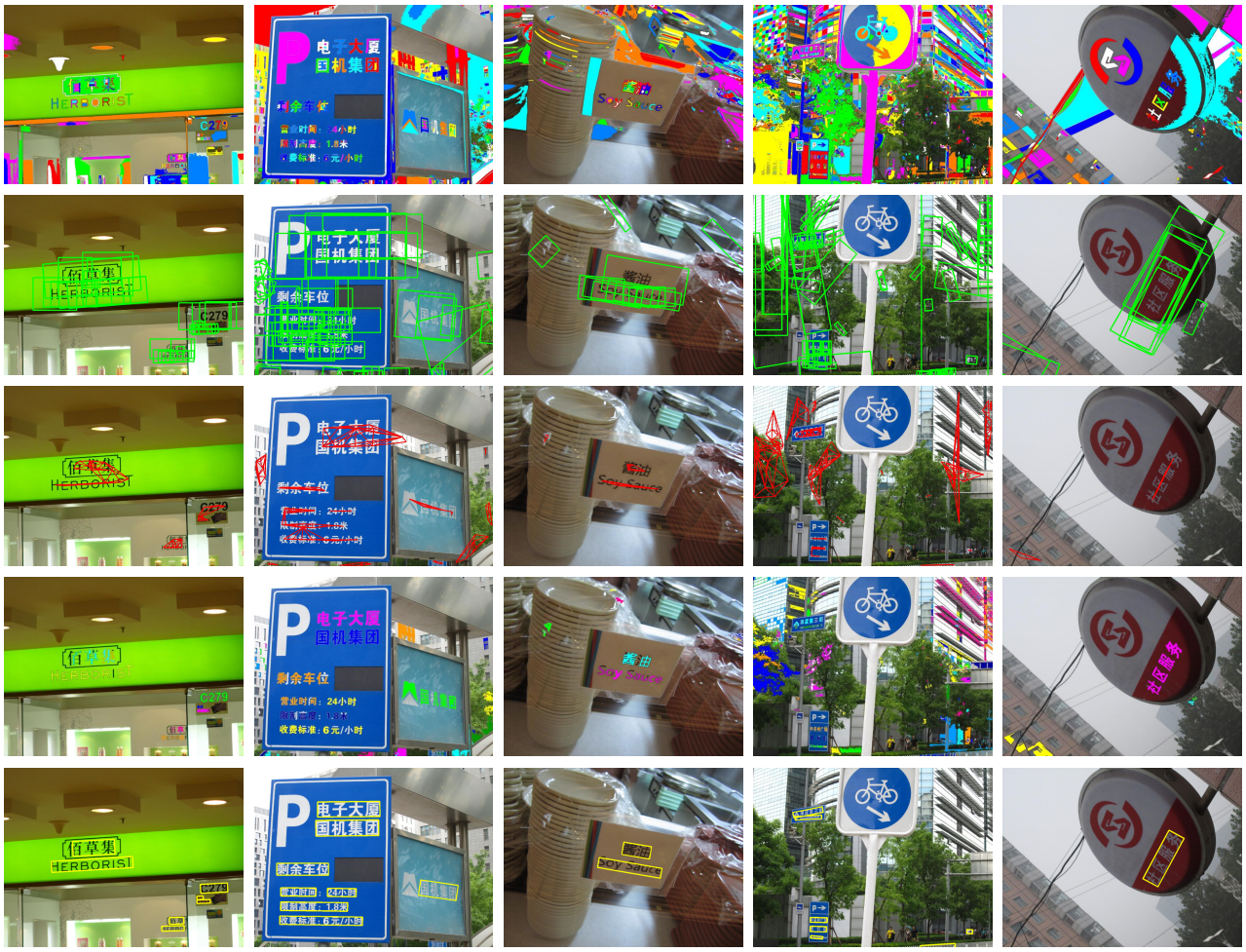
We hypothesize that text can be better identified by properties of a group rather than of individual characters (Figure 1). Individual image elements have a lot of variation and tend to cause false alarms for those methods explicitly using character models. But a group of similar elements provides more robust statistics for discriminating text from noise. Therefore, it is natural to group image elements based on pairwise and groupwise similarity, and then classify them as text or non-text regions. This can be regarded as a trade off between top down detections and bottom up heuristic rules.

This paper approaches the text detection problem from an image partitioning perspective, and proposes a general framework to detect multi-oriented scene text lines with less dependency on font or language. Our goal is to group similar elements first and then identify each group as text or non-text. Specifically, we use MSERs [7] as the basic elements and partition them to segments.

This results in a few important changes in the processing flow. Instead of focusing on the strong detection and strong filtering approaches, we use weak hypotheses for similarity clustering followed by region-based filtering. Correlation clustering was originally proposed by [8] as a similarity clustering approach, which models pairwise relationships between entities instead of the entities themselves and does not require specifying the number of clusters.

The elongated nature of text lines suggests that the long range dependencies among multiple nodes can be exploited. We explore the Higher-Order Correlation Clustering (HOCC) [9], which is an improvement of the original correlation clustering where higher-order relations can be incorporated. In HOCC, long range interactions can be defined by less accurate measurements (*i.e.*, weak hypotheses), because they are regarded as soft constraints in clustering. For instance, weak hypotheses of local text lines can be generated based on their spatial alignment and appearance consistency with respect to their neighbors. HOCC may re-

The partial support of this research by DARPA through BBN/DARPA Award HR0011-08-C-0004 under subcontract 9500009235, the US Government through NSF Awards IIS-0812111 and IIS-1262122 is gratefully acknowledged.



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

Figure 1. Intermediate results in our procedure. From top to bottom, MSER extraction, local text line hypotheses (green bounding boxes), pairwise edges in HOCC, results for HOCC, results for texture classification (yellow bounding boxes). Different MSERs/regions are represented by different colors.

ject these hypotheses depending on the objective function of these similarities, which is a notable differences between the higher order correlation clustering and other classical higher order Markov Random Field (MRF). Another appealing property of HOCC is that it allows large margin training. Using structured SVM [10], the parameters of HOCC can be learned from the training data [9], and we are saved from adopting too many heuristics.

In HOCC, we propose to use the regularization method [11] to efficiently solve the Semidefinite Programming (SDP) problem [12]. The original HOCC proposed a linear programming relaxation solution with a large number of inequality constraints. This complex linear system can be written elegantly in the SDP framework. This allows us to effectively solve the system with a large number of variables in a few seconds. After clustering, we use a texton-based texture classifier to filter out non-text areas.

We compare with the methods that aim at detecting multi-oriented and multi-language text. On a recently published dataset, our method generates promising results compared to the state of the art methods.

2. Building graphs for detection-by-clustering

In our approach, an image is represented by a graph of Maximally Stable Extremal Regions (MSERs). The detection amounts to identifying subgraphs that contain text lines. We adopt MSER because our approach is intended to perform detection for multiple languages at multiple orientations, thus we seek generic representations of text regions.

Among graph partitioning algorithms, correlation based methods are particularly well suited for our problem. These methods rely on pairwise similarity, such as contrast, solidity (the ratio of contour area to its convex hull area), area ratio, distance in images. More interestingly, prior knowl-

edge can be incorporated as weak hypotheses of grouping.

First, MSERs are extracted and a graph of MSERs is constructed for an image. Then, MSERs are coarsely grouped considering their consistency with neighbors to create weak hypothesis. The correlation based grouping is described in the next section.

2.1. Extracting MSERs and building the graph

First, we compute MSERs for an input image (first row in Figure 1). One can see that the sizes of MSERs vary, and the MSERs may correspond to character regions or noise.

We then construct the graph of MSERs locally to avoid unnecessary edges between distant MSERs. Delaunay triangulation is employed to find pairwise edges in the graph.

2.2. Computing local consistency map

To construct the graph, we first compute a local consistency map for each MSER. A local consistency map is a probability map, which shows how an MSER is consistent with its neighbors in a small patch.

For each MSER, we consider a context patch which is an image patch centered at the MSER. The MSER is referred to as the center MSER, and other MSERs on this patch are considered its neighbors. The patch has a size 7 times the width and height of the center MSER. We compute the consistency score θ between the center MSER and its neighbors as:

$$\theta = \exp(-\alpha D_{co} - \beta D_{sw}) \quad (1)$$

where D_{co} is the Euclidean distance between the colors of two MSERs, D_{sw} is the normalized stroke width difference, and α and β are constant parameters. RGB color is used and averaged over all pixels for each MSER. The stroke width of an MSER is estimated by the largest distance from an interior point to boundary, and can be efficiently obtained using a Distance Transform.

Then a local consistency map for this context patch is constructed by transforming all pixels of MSERs on this patch to their consistency scores. Figure 2 shows two examples of context patches and local consistency maps. In Figure 2a, the center MSER is part of the cartoon cat's face, thus a high consistency score is obtained in other parts of its face and gives large intensity in the local consistency map. The text below the cartoon cat's face has very different color and stroke width, therefore it shows low consistency scores. In Figure 2b, we can see high consistency scores of the characters in the text line in the middle of the patch, while the character in the upper right corner has relatively low consistency scores due to the difference in font.

2.3. Weak hypotheses generation

We generate hypotheses based on the local consistency map. Prior knowledge of text line includes 1) text line must be elongated, and 2) the projection profile of a text line should have higher variance. Therefore, we project local consistency map in different orientations from -90 to 85 degrees (with respect to the horizon) with an interval of 5 degrees to obtain 36 projection profiles.

In practice, the projection at each orientation is performed not on the entire local consistency map but on an oriented narrow region whose width and length are 3 and 7 times those the selected MSER respectively, to include less noise in the projection profile. The raw profile of a given orientation is computed by summing all the intensities in the narrow region along that orientation. Then the raw profile is intensity-normalized by the mean of non-zero values in it, and its dimension is normalized to a predefined length by resampling to obtain the final projection profile.

The projection profile in the maximum variance orientation is intended to capture the text line structure which usually forms a peak, while the one in the orthogonal orientation tries to capture the regular intervals between characters as can be observed in most cases. Figure 2 shows the examples of projection profiles. The size of the patch is empirically determined and is not sensitive in our experiment.

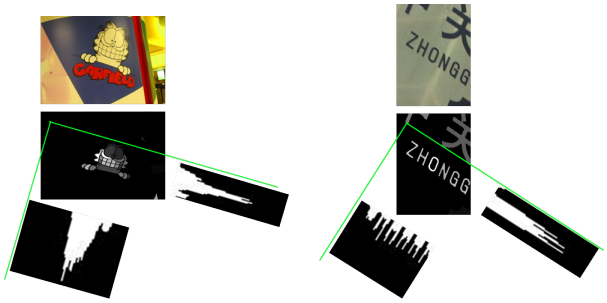
We then concatenate the projection profiles in the maximum variance orientation and its orthogonal orientation into a feature vector, and feed the feature vector to a Random Forest classifier [13]. The training samples of projection profiles from text and non-text patches were manually labeled. If a patch is determined as positive (contains text line structure) by the Random Forest classifier, all the MSERs on this patch with a similarity above a threshold are considered a local line hypothesis.

Using local consistency maps and projection profiles, we are able to capture the local text line structure on a patch. The projections may involve multiple text lines, because the random forest is trained on various profile patterns and can handle different situations. We stress that the classification on the projection profiles need not to be very accurate, i.e. false positives are expected, since the results will be fine-tuned by the correlation clustering.

After generating hypothesis in the entire image, there may exist two typical cases: 1) stand alone MSERs not covered by any local line hypotheses, and 2) disconnected subgraphs. The isolated MSERs are discarded, and disconnected graphs are processed separately (3^{rd} row in Figure 1).

3. Correlation clustering based text detection

We first briefly review the correlation clustering, and introduce the higher-order clustering (HOCC). Compared to the basic correlation clustering, the key ingredient of the HOCC is the "hyperedges", which are sets of pre-defined weak hypotheses in a graph. We further provide the solution to the HOCC using semidefinite programming.



(a) non-text (b)text

Figure 2. Local consistency map and the projection to two orthogonal directions. Please see the text for the details.

3.1. Correlation clustering

Given an undirected graph $G = (V, E)$, correlation clustering attempts to assign a binary label to each edge, indicating whether the two vertices are connected so that they are in the same cluster. Practically, this binary label is relaxed and a rounding procedure is generally required to effectively group the nodes.

3.1.1 Pairwise correlation clustering

Correlation clustering partitions nodes into clusters based on their pairwise similarities. Let $s_{ij}^p \in \{0, 1\}$ denote the pairwise similarity between node V_i and V_j (or on edge e_{ij}^p), and define x_{ij}^p as

$$x_{ij}^p = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are connected} \\ 0, & o.w. \end{cases} \tag{2}$$

The correlation clustering problem becomes an agreement maximization formulation (Q1):

$$\begin{cases} \max\limits_{x^p} \sum\limits_{i,j} s_{ij}^p x_{ij}^p \\ s_{ij} = \text{Sim}(V, i, j) = \langle \mathbf{w}^p, \phi_{ij}(V) \rangle \end{cases} \tag{3}$$

where $\phi_{ij}(V)$ denotes the features that characterize the difference between vertices i and j (See Sec. 5.2), and $\mathbf{w}^p$ is the parameter vector (to be learned from training samples). Therefore, the correlation clustering becomes an integer programming problem.

In the literature, two solutions were proposed to solve Eq. 3. Kim *et al.* [9] used a number of inequalities, such as cyclic inequality constraints and odd-wheel constraints, to create physically meaningful polyhedrons. On the other hand [12], we can rewrite Eq. 3 to matrix form as follows

$$\begin{cases} \max\limits_{X} & trace(S^T X) \\ s.t. & X \succeq 0 \end{cases}. \tag{4}$$

where $S(i, j) = s_{i,j}$, and $X(i, j) = x_{i,j}$. This SDP formulation makes the problem elegant, but is limited by the number of variables in practice. We will describe our solver in Sec. 3.2.

3.1.2 Higher-order correlation clustering (HOCC)

A hyperedge can be simply regarded as a subset of connected edges. Each hyperedge $e_k^h \in E$ contains more than one pairwise edge, *i.e.*, $|e_k^h| \geq 2$. A hyperedge can be activated or deactivated, denoted by $x_k^h \in \{0, 1\}$, and is associated with a groupwise similarity $s_k^h = \langle \mathbf{w}^h, \phi^h(V)_k \rangle$, where $\phi_k^h(V)$ denotes the features for the kth hyperedge (See Sec. 5.2). Then the objective function becomes (Q2):

$$\begin{aligned} &\max_{x^p, x^h} \sum_{i,j} s_{ij}^p x_{ij}^p + \sum_k s_k^h x_k^h \\ =&\max_{x^p, x^h} \sum_{i,j} \left\langle \mathbf{w}^p, \phi_{i,j}^p(V) \right\rangle x_{i,j}^p + \sum_k \left\langle \mathbf{w}^h, \phi_k^h(V) \right\rangle x_k^h \\ =&\max_X \langle \mathbf{w}, \Phi(V, X) \rangle \end{aligned} \tag{5}$$

where $\mathbf{w}^h$ denotes the parameter vector for the hyperedge features and $\mathbf{w}$ is concatenation of $\mathbf{w}^p$ and $\mathbf{w}^h$. $\mathbf{w}$ is learned from training data. X contains both pairwise edges and hyperedges. $\Phi(V, X)$ denotes the joint feature maps of all edges.

Binary operations were used to model the relation between higher order labels and pairwise labels. To be specific, a number of inequalities are used as follows.

$$\begin{aligned} x_k^h \leq& x_{ij}^p, \quad \forall e_{ij}^p \subset e_k^h, \\ x_k^h \geq& 1 - \sum_{i,j | e_{ij}^p \subset e^h} (1 - x_{ij}^p) \end{aligned} \tag{6}$$

The first inequality in ensures that the nodes in different clusters can not be in the same activated hyperedge, and the second ensures that the nodes in the same cluster can not have a deactivated hyperedge. Please refer to [9] for more details.

3.2. Effective solution for "long-tailed" SDP

Semidefinite programming has attracted a reasonable attention in recent years. Its applications range from kernel learning to low rank approximation [14]. While a few interior point based SDP packages are available (e.g., [15]), efficiency is the major consideration when SDP is used in real applications. Contrary to interior point based methods, boundary point based approaches, such as regularization methods, surface as alternatives. For example, the method in [11] is much easier to implement and can be taylored easily if there exist special structures in SDP.

Algorithm 1 Regularization SDP.

Input:

t : Real positive scalar; S, Y : Symmetric matrix.

Z : Semipositive definite matrix.

A, b : Linear mapping, $AX = b$.

Output:

X : SDP solution for $\min trace(S^T X)$ subject to $AX = b, X \succeq 0$;

Procedure:

Repeat until $|Z + A^T y - S|$ is small

Step 1: Solve y for $AA^T y + A(Z - S) = (b - AY)/t$

Step 2: Set $X = t(Y/t + A^T y - S)_+$

Step 3: Set $Z = -(Y/t + A^T y - S)_-$

Step 4: Set $Y = X$.

In a number of SDP problems, there exists a "long tailed" block diagonal structure: a small number of SDP variables but a large number of slack variables. Because the diagonal matrix is semipositive definite if and only if the values are nonnegative, this structure can be solved efficiently.

In these cases, regularization methods turn out to be more efficient than interior point methods. Since this type of method is not widely known in the computer vision community, we briefly describe one of these methods in this section.

In Algorithm 1, $AX = b$ is a linear mapping of X, representing the constraints. In the literature, this may also be written as $A(X) = b$. The operation $(\cdot)_{+/-}$ denotes the projection to the positive/negative definite space. Please refer to [11] for more details.

In our opinion, Algorithm 1 is an elegant solver that tackles many vision problems and has the advantages that 1) the implementation is straightforward (20+ lines in MATLAB) and 2) it is an order of magnitude more efficient than other MATLAB SDP packages.

To see how this formulation speeds up our problem, please note that the time consuming step is the eigendecomposition in the projection step in the internal problem (Step 2). In our formulation, since X has a long tailed structure, eigendecomposition only needs to be performed on the SDP variables. As a result, this solution is very efficient for small problems ($\backsim$300 SDP variables in our case), but may be slow for larger problems (*e.g.*, more than 1000 SDP variables).

3.3. Structural learning

Structured SVM [10] is used to learn the parameter vector $\mathbf{w}$. Consistent with previous notation, let $\{(V_n, X_n)\}$ dentoe N training samples, where V_n is the n^{th} training graph (with features) and X_n is its ground truth labels. Then $\mathbf{w}$ is learned by:

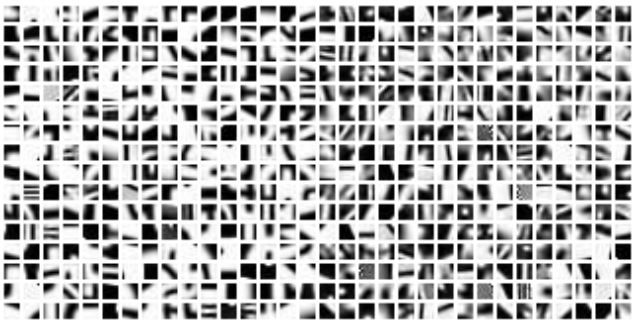


Figure 3. Samples of textons learned in the texture classification.

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\ s.t.\ &\forall n, X \succeq 0, \\ &\langle \mathbf{w}, \Delta\Phi(V_n, X)\rangle \geq \Delta(X_n, X), \\ &\xi_n \geq 0 \end{aligned} \tag{7}$$

where $\Delta\Phi(V_n, X) = \Phi(V_n, X_n) - \Phi(V_n, X)$, and $C > 0$ is a constant. More details can be found in [9].

4. Classifying text and non-text regions

After applying HOCC, we obtain a number of image regions. Each region represents a consistent group of either text or non-text MSERs (Forth row in Figure 1). The decision of text/non-text will be made on each region as a whole.

We treat this step as a texture classification problem. Liu *et al.* [16] provides a state of the art approach based on random projections. In their approach, densely sampled image patches are projected with a random matrix and mapped to a dictionary of textons through nearest neighbor to obtain a histogram. However, in our problem we find random projections harm the classification accuracy. The reason could be that both text and non-text patterns have a lot variations and random projections inevitably lose some information that may distinguish these two closely entangled classes.

Coates *et al.* [3] describes a method to detect text in a squared sliding window. In their method, whitened patches are encoded with an unsupervised dictionary using soft-thresholding and summed up over 9 blocks to form a feature vector for this image window. Yet in our experiments we found the naive nearest neighbor encoding outperformed soft-thresholding.

Our approach is similar in spirit as the two above. In our procedure, we separate the text and non-text patches when constructing the dictionary of textons. Through experiments we find this explicit separation of textons results in better classification performance compared to mixing text and non-text textons. Our procedure is as follows:

1. Pre-processing: harvest 8×8 grayscale patches from text regions and non-text regions, and apply brightness and contrast normalization.

2. Constructing dictionary: perform k-means on text patches and non-text patches respectively, and combine the two sets of textons to form a dictionary.

3. Computing histogram: for each patch, find the nearest neighbor in the dictionary to form a histogram, and normalize the histograms to unit sum to produce the feature representation for an image region.

4. Classifying: use Random Forest on the unit histograms.

Figure 3 shows samples of our textons. The textons corresponding to text and non-text patches share some common properties such as they both contain stroke-like patterns. However, there exists difference of statistical properties between text and non-text patches. Text and non-text patches may be encoded into different histograms, which enables the classification. Since no character models or other specified features are involved, this method is expected to generalize on text of different languages and styles.

5. Experiments

We demonstrate the effectiveness of our approach in this section. One of our goals is to cluster similar text prior to any classification. Thus it is natural for us to select the multi-orientation multi-scripts dataset such as the MSRA-TD500, instead of other datasets that primarily contains horizontal English text, which may be solved more effectively by strong character models and lexicons.

First, we describe two datasets used in our experiments. Then, we show the results of our method and compare them to the state of the art.

5.1. Datasets

5.1.1 MSRA-TD500 database

The MSRA Text Detection Database (MSRA-TD500) contains 500 images of indoor and outdoor scenes. This dataset is very challenging in three ways.

First the text is bilingual, including English, Chinese and mixture of the two, and they are in wide range of fonts, sizes and styles. Second, the text is in arbitrary orientation. Third, the background is diversified and complex.

To evaluate the performance on MSRA-TD500, we follow the protocol employed by [1]. We used 300 images for training and 200 for testing. A minimum rectangle is fit to the detected text region, and its orientation is also estimated. A ground truth rectangle can only be matched once, therefore many-to-one match is not allowed.

(a) (b) (c)

Figure 4. More results in the MSRA dataset. From top to bottom, input image, MSER extraction, HOCC results and final detection results. Please see Figure 1 for the meaning of the colors / bounding boxes.

We define the overlap ratio between a detected rectangle and a ground truth region as the ratio of the areas of their intersection and union. The rectangle is considered correct if the orientation difference is less than $\pi/8$ and the overlap ratio exceeds 0.5. We would like to note that the 0.5 overlapping criteria is different from other text detection criteria, but is consistent with PASCAL challenge in for object detection.

5.1.2 OSTD dataset

The OSTD dataset contains 89 images of indoor and outdoor scenes (Figure 6). Text in this dataset is diversified in orientations, view perspectives, fonts and styles. Following [1], the proposed algorithm trained on MSRA-TD500 runs on all images of OSTD. To make a fair comparison, we employ the same protocol as [17].

5.2. Features for clustering

The proposed method relies on the pairwise similarity and groupwise similarity. This section defines our features for correlation clustering in the text line detections.

Pairwise features We used a 18-dimensional features in pairwise comparison.

- Stroke width difference and ratio.
- RGB distance in: Euclidean, χ^2, EMD [18], L_1.

- CIELAB distance in: Euclidean, χ^2, EMD, L_1.
- Distance of contrast vectors (CIELAB of MSER subtracting its immediate background) in four metric : Euclidean, χ^2, EMD, L_1.
- Solidity difference.
- Area ratio difference
- Distance between two MSERs normalized by sizes and patch orientation.

Higher order features We used a 12-dimensional features to describe group properties in hyperedges.

- Variances in the RGB values, respectively.
- Variances in the CIELAB values, respectively.
- Variance in area normalized by the median.
- Variance in solidity.
- Variance in stroke width normalized by the median.

With the features defined, we are able to learn w^p and w^h for transforming the feature maps to the similarity measurement. We used the loss function in the [9], but assign different weights due to the intrinsic properties of text images.

5.3. Results

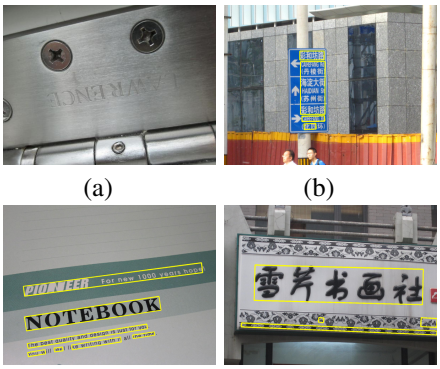
5.3.1 MSRA-TD 500

We first present results for the MSRA-TD500 dataset (Figure 4). As shown in Figures 1 and 4, our method handles challenging cases. For example, Figure 1a has text of various fonts and mixed languages. Text in Figure 1d is very small and the background (trees) makes some text very difficult to detect (*e.g.*, the road name sign). Further, the bike pattern on the traffic sign has similar and consistent stroke widths as the text. Text in Figure 1e is overly slanted. We also notice from 4a that some handwritten text is correctly detected even though their shapes and colors are less consistent compared to the printed text.

Shown in Table 1 are the quantitative results. Among three methods we compared, Yao *et al.* [1] handles oriented text explicitly, and [5] is the baseline, where the stroke width transform was proposed.

The proposed algorithm is able to detect text lines in different fonts and orientations. In complex backgrounds like leaves, grass and some very challenging architectural patterns, we observe promising detection results. Our method achieves a higher precision with similar recall rate, compared to the state of the art method.

The HOCC plays an important role in this process, since it generates larger homogeneous regions to provide robust statistics for the discrimination between text and non-text. Avoiding making local decisions is essential to the proposed approach. Without an effective grouping process, it is difficult for such a simple texton-based text detection algorithm



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

Figure 5. Error analysis. a) MSERs are not well extracted due to lighting; b) text lines are too close and merged; c) text lines are broken into multiple parts; d) mistakes exist in texture classification.

Table 1. Performance comparison on MSRA-TD500.

	Precision	Recall	F-measure
Our method	0.71	0.62	0.66
Yao *et al.* [1]	0.63	0.63	0.60
Epstein *et al.* [5]	0.25	0.25	0.25
Chen *et al.* [19]	0.05	0.05	0.05

to achieve superior performance. For example, [3] reports less satisfying detection rates compared to other methods using highly specialized features.

The errors are mainly from three sources. First, some MSERs are not well extracted due to lighting, fragmentation, blur (Figure 5a) etc, in the graph construction stage. Second, some text lines are too close and are merged into one rectangle during clustering, due to the difficulty to resolve their local linear structures (Figure 5b). On the other hand, text lines are broken into multiple parts when relatively large gaps exist between characters or words due to elimination of some MSERs (Figure 5c). Third, mistakes exist in texture classification, where artificial and plant textures may be classified into text (Figure 5d).

5.3.2 OSTD

We also test the proposed algorithm on the Oriented Scene Text Dataset [17]. Figure 6 shows some results from the OSTD dataset. We achieved the best precision rate and the F measure, and our recall is tied with the state of the art.

We observe that the text are also correctly grouped and detected even though the properties of these text are quite different from the MSRA-TD500 dataset.

6. Conclusion

We have described a higher-order correlation clustering (HOCC) based framework for detecting multi-oriented text lines in natural images. The detection is treated as a graph

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

Figure 6. Examples in the OSTD dataset. Detection results are shown in yellow bounding box and overlayed on the input images.

Table 2. Performance comparison on OSTD.

	Precision	Recall	F-measure
Our method	0.80	0.73	0.76
Yao *et al.* [1]	0.77	0.73	0.74
Yi *et al.* [17]	0.56	0.64	0.55
Epstein *et al.* [5]	0.37	0.32	0.32
Chen *et al.* [19]	0.07	0.06	0.06

partitioning problem, where each node is represented by the MSER. The regularization method is used to solve the Semidefinite Programming problem in HOCC. Finally we used a texture classifier to filter the non-text areas. Experiments show our method is superior to the state of the art.

References

[1] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu, "Detecting texts of arbitrary orientations in natural images," in *CVPR'12*, 2012, pp. 1083–1090.

[2] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *CVPR*, 2012, pp. 3538–3545.

[3] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, Tao Wang, D.J. Wu, and A.Y. Ng, "Text detection and character recognition in scene images with unsupervised feature learning," in *ICDAR 2011*, sept. 2011, pp. 440 –445.

[4] Jung-Jin Lee, Pyoung-Hean Lee, Seong-Whan Lee, Alan L. Yuille, and Christof Koch, "Adaboost for text detection in natural scene," in *ICDAR*, 2011, pp. 429–434.

[5] Boris Epshtein, Eyal Ofek, and Yonatan Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR'10*, 2010, pp. 2963–2970.

[6] Kai Wang, Boris Babenko, and Serge Belongie, "End-to-end scene text recognition," in *ICCV'11*. IEEE, 2011.

[7] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[8] Nikhil Bansal, Avrim Blum, and Shuchi Chawla, "Correlation clustering," *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.

[9] Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli, and Chang D. D. Yoo, "Higher-order correlation clustering for image segmentation," in *NIPS*. 2011.

[10] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer, "Large margin methods for structured and interdependent output variables.," *Journal of Machine Learning Research*, vol. 6, no. 9, 2005.

[11] Jérôme Malick, Janez Povh, Franz Rendl, and Angelika Wiegele, "Regularization methods for semidefinite programming," *SIAM J. on Optimization*, vol. 20, no. 1, pp. 336–356, Apr. 2009.

[12] Micha Elsner and Warren Schudy, "Bounding and comparing methods for correlation clustering beyond ILP," in *In NAACL-HLT Workshop on Integer Linear Programming for Natural Language Processing (ILPNLP) 2009*, 2009.

[13] Leo Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] Ganzhao Yuan, Zhenjie Zhang, Bernard Ghanem, and Zhifeng Hao, "Low-rank quadratic semidefinite programming," *Neurocomput.*, vol. 106, pp. 51–60, Apr. 2013.

[15] R. H. Tucuncu, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathematical Programming*, vol. 95, pp. 189–217, 2003.

[16] Li Liu and P.W. Fieguth, "Texture classification from random features," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 574–586, 2012.

[17] Chucai Yi and YingLi Tian, "Text string detection from natural scenes by structure-based partition and grouping," *Image Processing, IEEE Transactions on*, vol. 20, no. 9, pp. 2594–2605, 2011.

[18] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, "A metric for distributions with applications to image databases," 1998, ICCV'98.

[19] Xiangrong Chen and Alan L Yuille, "Detecting and reading text in natural scenes," in *CVPR'04*. IEEE, 2004, vol. 2, pp. II–366.