Effective Uyghur Language Text Detection in Complex Background Images for Traffic Prompt Identification

Chenggang Yan, Hongtao Xie, Shun Liu, Jian Yin, Yongdong Zhang, *Senior Member, IEEE*, and Qionghai Dai, *Senior Member, IEEE*

***Abstract*— Text detection in complex background images is a challenging task for intelligent vehicles. Actually, almost all the widely-used systems focus on commonly used languages while for some minority languages, such as the Uyghur language, text detection is paid less attention. In this paper, we propose an effective Uyghur language text detection system in complex background images. First, a new channel-enhanced maximally stable extremal regions (MSERs) algorithm is put forward to detect component candidates. Second, a two-layer filtering mechanism is designed to remove most non-character regions. Third, the remaining component regions are connected into short chains, and the short chains are extended by a novel extension algorithm to connect the missed MSERs. Finally, a two-layer chain elimination filter is proposed to prune the non-text chains. To evaluate the system, we build a new data set by various Uyghur texts with complex backgrounds. Extensive experimental comparisons show that our system is obviously effective for Uyghur language text detection in complex background images. The F-measure is 85%, which is much better than the state-of-the-art performance of 75.5%.**

***Index Terms*— Smart transportation, intelligent vehicles, Uyghur text detection, the channel-enhanced MSER.**

I. Introduction

TEXT in images, such as traffic prompts [38], often contains valuable information. With the rapid progress of intelligent vehicles, many researchers begin to study recognizing the traffic prompt in images [39]. However, how to extract the traffic prompts effectively and efficiently from images is a difficult problem [40], due to the influence of complex background and their variations in scale and illumination. As a prerequisite of text recognition, text detection in complex images is very important and essential. Many interesting literatures are published in premier journals [1], [2] and international conferences [3], [4], [10], [17], [18] around the theme of text detection and recognition. Some competitions about document analysis like "Robust Reading" are also held around every couple years [1]. But most of these literatures focus on commonly used language text detection. Although some institutions have begun to promote some minority language text detection and recognition research in the past few years, only a few methods are put forward [4]. In this paper, we focus on Uyghur language detection in complex background images. As there are 8 to 11 millions of people using Uyghur language worldwide [5], the research of Uyghur language detection and recognition in images is very significative.

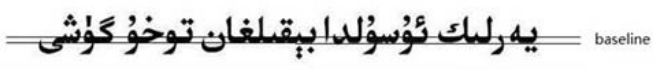


Fig. 1. The characteristics of Uyghur language.

The Uyghur language has many similar characteristics to Arabic in writing. The characteristics of the Arabic language do not allow direct adoption of many algorithms designed for other languages [5], since the recognition scheme depends on the type of character being recognized [6], so does Uyghur language. The distinctive characteristics of Uyghur language make it different from the commonly used languages, as illustrated in Fig. 1:

1) The shape of Uyghur characters is not square, and the characters or words in one sentence always conglutinate with each other.
2) Many characters have one to three dots or zigzags, and the zigzags can be in the different positions of the character.
3) There are no uppercase or lowercase letters.
4) The sentences have one "baseline" in the middle of text.

To detect the text regions in complex background images, Maximally Stable Extremal Regions (MSERs) [9] algorithm is always adopted for its good performance. But the MSERs methods cannot deal with blurry images and low contrast

Manuscript received February 12, 2017; revised May 22, 2017; accepted June 11, 2017. Date of publication October 10, 2017; date of current version December 26, 2017. This work was supported in part by the National Nature Science Foundation of China under Grant 61771468, Grant 61525206, Grant 61671196, and Grant 61327902, in part by Zhejiang Province Nature Science Foundation of China under Grant LR17F030006, and in part by the Youth Innovation Promotion Association Chinese Academy of Sciences under Grant 2017209. The Associate Editor for this paper was P. Ioannou. *(Corresponding author: Hongtao Xie.)*

C. Yan is with the Institute of Information and Control, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: cgyan@hdu.edu.cn).

H. Xie and Y. Zhang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: xiehongtao@ict.ac.cn; zhyd73@ustc.edu.cn).

S. Liu is with the National Engineering Laboratory for Information Security Technologies, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and also with the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China (e-mail: liushun@nelmail.iie.ac.cn).

J. Yin is with the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China (e-mail: yinjian@sdu.edu.cn).

Q. Dai is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: qhdai@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2749977

characters [17]. These obscure objects in color image always contrast obviously with the background in one or more of the R, G and B channel. When transforming the color images to grayscale, the contrast between objects with background usually becomes weaker, and leads to some fuzzy regions missed. Therefore, we propose the channel-enhanced MSERs algorithm in this paper. The MSERs algorithm is used in the R, G and B channel of the image to be processed, and then the MSERs on the three channels are marked on the color image. With channel-enhanced MSERs algorithm, nearly all the text regions can be extracted. But the main problem of MSERs algorithm is that many of the detected regions are actually repeating with each other or the noises (the background of the image). So, it is a key point to remove the repetitions and noises.

Accordingly, we propose an effective Uyghur language text detection system in this paper. Firstly, a new channel-enhanced MSERs algorithm is put forward to detect component candidates. Secondly, a two-layer filtering mechanism is designed to remove most non-character regions. In the first layer, several heuristic rules are applied to prune the repetitions and the apparent non-text regions. In the second layer, we propose a one word classifier, which is trained by Histogram of Oriented Gradient (HOG) features, to identify the text regions. Thirdly, the remaining component regions are connected into short chains, and the short chains are extended by a novel extension algorithm to connect the missed MSERs. Finally, a two-layer chain elimination filter is proposed to prune the non-text chains. The first layer includes some chain-level heuristic rules, which is used to prune the chains with illogical aspect ratio or area, and the second layer is a Random Forest [8] classifier trained by appropriate gradient features. The classifier performs the task to appraise those chains through the first layer. The chains passed the chain elimination algorithm are the final regions.

To evaluate the proposed system, we build a new image dataset, which is called IMAGE570, by various Uyghur texts with complex backgrounds. Extensive experimental comparisons show that our system is obviously effective for Uyghur language text detection in complex background images. The F-measure is 85%, which is much better than the state-of-the-art performance of 75.5%. Moreover, experiments on the parameters of component analysis stage and chain analysis stage are conducted to test the performance of the classifiers designed in our system.

The remaining sections are organized as follows. Section II gives a comprehensive review of previous works. Section III elaborates the structure and details of the system. Section IV introduces the dataset and evaluation protocol. Experimental comparisons are presented in Section V. Finally, section VI gives conclusion and discussion.

II. Related Works

There is a large amount of algorithms detect text in color images. Comprehensive surveys can refer to [2], [11]. The current algorithms of text detection can be generally categorized to five groups: edge-based methods [13], texture-based methods [12], [29], stroke-based methods [3], [10], connect component-based methods [1], [16], [17], [19], [21], [25] and deep neural network (DNN)-based methods [20], [37].

Edges are reliable features for text detection. Text always exhibits a strong gradient against the background. The pixels which have large and symmetric gradient values can be deemed as text candidates. Generally, the edge detector is used first followed by morphological operations to distinguish text from background and to remove non-text regions. Texture-based methods are utilized because texts in images have distinct textural characteristic which differentiate them from the background. These texture analysis approaches are always be used such as Local Binary Pattern (LBP) [22], Wavelet decomposition [23], Discrete Cosine Transform (DCT) [24] and Fourier transform [27]. Texture features are always adopted with other features. Ye *et al.*[29] used texture features of image combined with color and statistical features to detect text. As a fundamental component of text, strokes supply efficacious features for text detection. It can be considered that text is combined by stroke parts with various orientations. Then the combinations and distributions of the stroke elements can be served as text features. Stroke width, as it is almost constant and can distinguish text from the background, is accordingly adopted as text feature. Epshtein *et al.* [3] proposed Stroke Width Transform (SWT) algorithm and assembled neighboring pixels with similar stroke width to text candidates. With SWT, Huang *et al.* [31] proposed a filter named Stroke Feature Transform (SFT) by merging color cues of text pixels.

The connected component methods could be regarded as graph algorithms (e.g. graph matching [33] and object detection [34], [35]), which make a map based on heuristics about feature consensus. The pixels with similar features are connected into the word candidates. Then the connected component-based algorithms use classifiers to remove non-character regions and group small components with similar properties into successively larger components till all the regions are processed [1], [15], [19]. Connected component-based methods divide the text detection to two successive parts: connected component generation and text/non-text classification. In the past few years, extremal region (ER) [17] and its successor such as MSER [1], [9], [25], [26], edge-enhanced MSER [21], color enhanced contrasting extremal region [15] have increasingly proved good performance of detecting word candidates. All of the ER-based and MSER-based algorithms could be regarded as connected component-based algorithms.

In recent years, DNN based methods have obtained great success in many applications [20], and there are many end-to-end systems are built depending on DNNs. Although DNNs generally have superior detection precision than other algorithms, they always need a large number of images for training and the time complexity is always much higher. There also appeared some new methods; such as Zhang *et al.* [30] proposed a symmetry based text line detection algorithm combined symmetry feature and appearance feature and Wang *et al.* [32] introduced an approach based on the confidence map and context information to detect texts.

As described above, many researchers receive good results using MSERs to detect text in images [1], [17], [25], [26]. The main merit of MSER-based methods over traditional connect

component methods lies in that the MSERs method can detect most components even when the quality of the image is low. However, the MSER-methods have some pitfalls. They may sustain the detection of repeating and non-text components (noises) and they are also insufficient for some text detection with low contrast. Repetitions and noises are big problems for characters detection, thus most of the repeating MSERs and non-text regions should be pruned before being inputted to the character grouping stage. Thus, many strategies are proposed for MSERs pruning [1]. Kinds of classifiers or filters are employed to delete the noises. Huang *et al.* [19] trained a CNN as a classifier to prune non-text regions. Chen *et al.*[21] designed a filter using geometric and stroke width information. Hyung *et al.* [26] used an Adaboost classifier to determine the adjacent relationship of CCs by pairwise spatial features firstly and then pruned non-text regions. To detect as many text regions as possible by MSERs, some enhanced MSERs algorithms combined with other features are designed. Chen *et al.* [21] introduced the edge-enhanced MSERs. Sun *et al.*[15] proposed a color-enhanced contrasting extremal region algorithm.

III. Methodology

In this section, we elaborate the proposed channel-enhanced MSER-based Uyghur text detection framework, which is designed with the merits of MSER-based methods and the characteristics of Uyghur language. Specially, the structure of the algorithm will be presented in SubSec. *A*. The details of each step will be described in SubSec. *B*, SubSec. *C*, SubSec. *D*, and SubSec. *E*. Some heuristic rules used in the chain elimination algorithm will be discussed in Sec. *F*.

By incorporating several advantages of MSER-based methods and the characteristics of Uyghur language.

A. System Overview

The Uyghur text detection system consists of four stages: (1) Component Extraction, (2) Component Analysis, (3) Chain Linking and (4) Chain Analysis, as illustrated in Fig.2. These four stages can be essentially systematized into two processes: grouping and pruning. In grouping, pixels first form connected components and then the connected components are aggregated to chains. In pruning, non-text components and chains are successively identified and eliminated by filtering algorithms.

1) Component Extraction: in this stage, a new channel-enhanced MSERs algorithm is put forward to detect component candidates. More details are presented in SubSec. *B*.

2) Component Analysis: as many extracted components of Component Extraction are not parts of texts. In this stage, a two-layer filtering strategy is designed to prune the repetitions and non-text regions.

In the first layer, several heuristic rules are applied to prune the repetitions and the apparent non-text regions. In the second layer, we propose a one word classifier, which is trained by HOG features, to remove the non-text regions which are hard to be pruned by the first layer. The most of the non-text components can be filtered out by the filtering mechanism. More details are presented in SubSec. *C*.

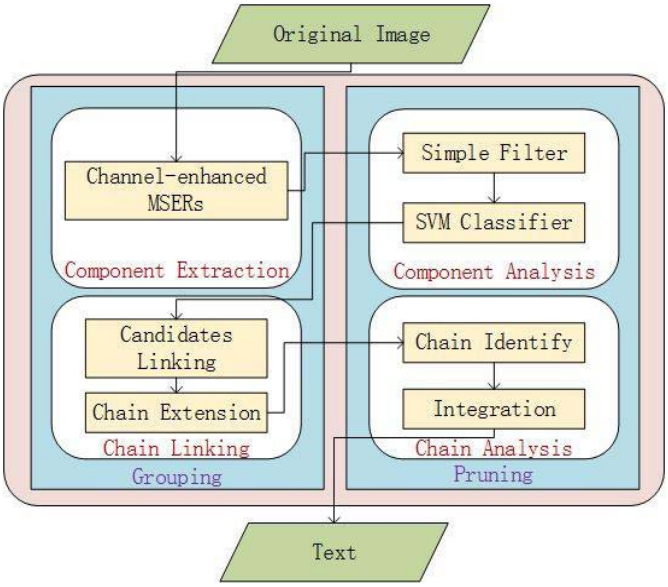


Fig. 2. The architecture of the proposed system.

3) Chain Linking: the remaining components are taken as word candidates after Component Analysis. It links the components into pairs (short chains), and the pairs are extended forward and backward by a novel extension algorithm to connect the missed MSERs. More details are presented in SubSec. *D*.

4) Chain Analysis: the chains determined at the former stage are verified by a two-layer chain elimination filter. The first layer includes some chain-level heuristic rules, which is used to prune the chains with illogical aspect ratio or area, and the second layer is a Random Forest [8] classifier trained by appropriate gradient features. The chains passed the stage are the final results. More details are presented in SubSec. *E*.

B. Component Extraction

MSERs algorithm is used to extract components from an image, for its robustness and efficiency. According to Mates *et al.* [9], an extremal region is a connected part of an image whose pixels have consistent higher or lower intensity than its outer boundary pixels. Let $p_1, p_2, p_3 \cdots p_i$ be a sequence of nested extremal regions, i.e. $p_i \subset p_{i+1}$, p_i is a MSER if

$$q(i) = |p_{i+\Delta} \backslash p_{i-\Delta}| \big/ |p_i| \tag{1}$$

has a local minimum at. In this paper, we define MSER the same as in [1]. For

$$v(i) = |p_{i+\Delta} - p_i| \big/ |p_i| \tag{2}$$

p_i is one MSER if $v(i)$ has a local minimum at i (Δ is a parameter). MSERs define an extremal region as a connected component of an image whose pixels have intensity contrast against its boundary pixels [9]. The MSERs algorithm is always used to deal with grayscale image. But the intensity contrast between the foreground and the background becomes weak when the color image is transformed to grayscale. Generally, many objects (foreground) contrast with the background more obvious in one of the R, G and B channel than in grayscale. So a new channel-enhanced MSERs algorithm is

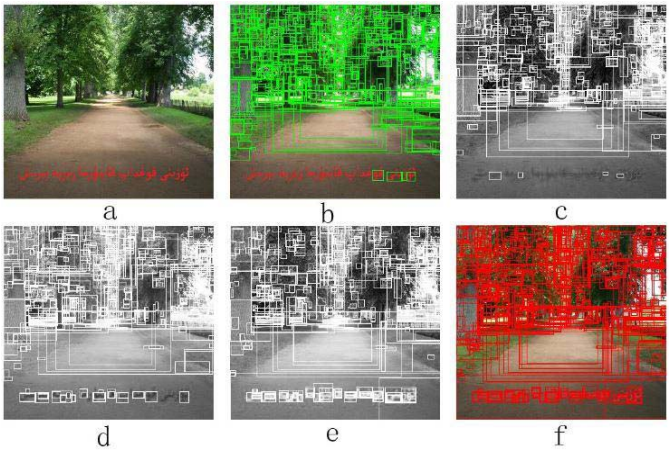


Fig. 3. a) the color image, b) the results of MSERs algorithm, with $\Delta = 1$, c) the results of MSERs algorithm in B channel image, with $\Delta = 3$, d) the results of MSERs algorithm in G channel image, with $\Delta = 3$, e) the result of MSERs algorithm in R channel image, with $\Delta = 3$ and f) the result of channel-enhanced MSERs.

put forward in this paper. To reserve the original information in the color image and extract as many text regions as possible, the MSERs algorithm [9] is executed in R, G and B channel, respectively. Then all the candidate regions are marked up in the color image. The same regions would be deleted by comparing the central coordinates and the corner coordinates of their bounding boxes. Nearly all the text regions can be extracted with the channel-enhanced MSERs algorithm. Even when the value of Δ used in this algorithm is larger than that in MSERs algorithm, the result of channel-enhanced MSERs algorithm is always better than using MSERs. Fig.3 shows the procedure and results of channel-enhanced MSERs, with a sample image.

From Fig.3 we can see that channel-enhanced MSERs is much better than MSERs. But in the candidate region of channel-enhanced MSERs, there exist many overlapping regions and noises, which is also the common shortcoming of connected component-based algorithm. So we have to remove the repeating regions and noises.

C. Component Analysis

There are many regions returned in Component Extraction stage and most of them are repetitions and non-text regions. The non-text regions could be called as noises, which should be pruned. The goal of the Component Analysis is to identify and eliminate the repetitions and noises. Thus, we design a two-layer filtering mechanism. In the first layer, several heuristic rules are applied to prune the repetitions and the apparent non-text regions. In the second layer, we propose a one word classifier, which is trained by HOG features, to remove the non-text regions which are hard to be pruned by the first layer.

For two overlapping regions, if the intersection area is over 80 percent of the union area, the region with larger MSER variation ($v(i)$) is pruned. If they have same variation value, the region with smaller area is pruned. As the definition of MSER, we know that a big text region may contain some but not many small MSER regions. So we set a threshold ($= 8$),

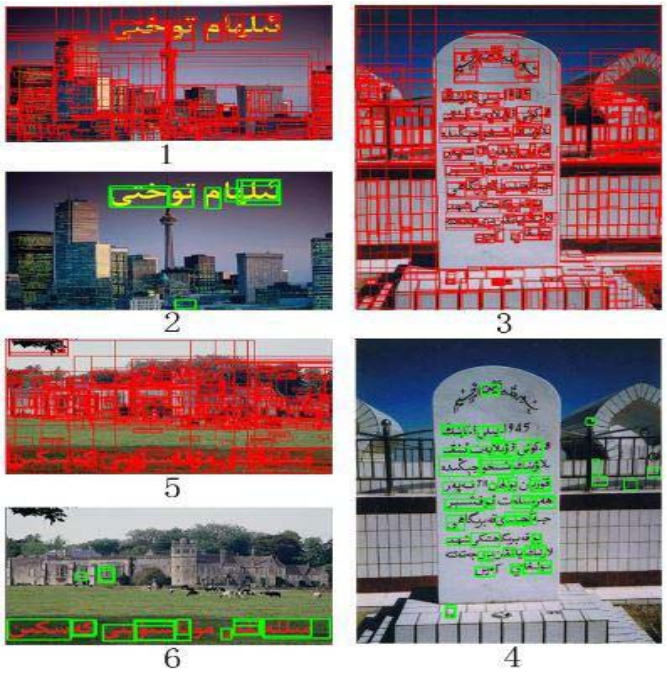


Fig. 4. 1), 3) and 5) the images with all MSERs, 2), 4) and 6) the images processed after the component analysis stage.

for a region if the number of containing little regions is larger than the threshold, it is considered to be noise and removed. With the process of the first layer, about 30 to 50 percent of MSERs are pruned.

In the second layer, we propose a one word SVM classifier [28] to remove the non-text regions which are hard to be pruned by the first layer. The SVM classifier is trained with the HOG feature [18] of the components. In terms of the characteristic of Uyghur word and the phenomenon of many experimental comparisons, the regions are all resized to 24×32 ($height \times width$) pixels (the results with different size have been discussed in the experiments). For Uyghur components (words) detection, the 8×8 cells blocks of 4×4 pixel cells and 9 bins for HOG feature work best. The SVM classifier of polynomial kernel is applied as the strong classifier.

With the process of Component Analysis, according to the data of experiments, about 98.2% MSERs are pruned by the two-layer filtering mechanism. Fig.4 shows some images processed by the filtering mechanism.

D. Chain Linking

In this stage, two operations are executed: connecting component candidates into chains and extending short chains to complete ones.

After Component Analysis, only the text regions and a small number of non-text regions are left. For text detection, the component candidates should be connected into chains. We first link the candidate regions into pairs. The linking standard is determined by their horizontal positions, heights and the distance between them. Every region's central coordinates have been recorded in the Component Analysis stage. Before linking two word candidates into a pair, the value of Y axis

of their central coordinate must be compared to make sure that they are in the similar horizontal position (the difference value of two regions does not exceed half of the smaller value of their box height). The two candidates must have similar height (the height of one must be 0.5 times larger and 2 times less than another's) and are close enough (distance between them is less than 2 times of the larger region's height). After the above prerequisites are checked, the applicable regions are connected together. When connecting two candidates, a new rectangle would be computed to contain the two regions. The process is taken from the left of the image to the right. At the end, one or more short chains appear on one horizontal line.

However, the short chains are usually only parts of one line of text. A few text regions are removed by the first layer of the filtering mechanism as noises, because they contain too many little components, even they are parts of the Uyghur text chains. The missing MSERs will be aggregated into the short chains by the following extension strategy. To connect the missing MSERs into text chains, every short text chain sequentially searches forward and backward. For a MSER, which is not one of the existing candidates, if it is found in the left or right of a text chain, some discriminant conditions will be checked to decide whether it can be connected into the chain. These requirements are similar to the conditions that two component candidates connecting into a pair. Whether the center of the MSER is in the same horizontal line with the chain, the height of the MSER is similar to the chain's and the distance between the MSER and the chain is not far (the distance of them is less than 2 times of the region's height). If all these criteria are met, then the SVM classifier in the Component Analysis is used for identifying. If it is classified as a text region, the MSER will be linked into the chain. After the extension of short chains, the extended chains will be connected into integral chains. For any two chains, if they are close (the distance between them is less than 2.5 times of the larger one's height) or even crossed together and are in same horizontal position, they are connected together. In executing the chains linking, the rectangles around chains are modified again. Finally, the chains got from this stage are the chain candidates. Fig.5 shows the procedure of Chain Linking.

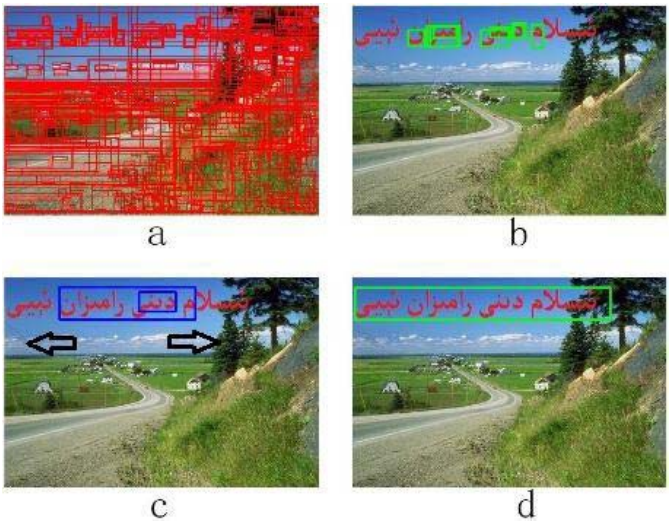


Fig. 5. a) all the MSERs, b) the candidate regions left after Component Analysis, c) candidate regions are connected into short chains, d) the two short chains extend forward and backward and connect together, getting the chain candidate.

E. Chain Analysis

The candidate chains obtained by the previous stage might include false positives, which are combinations of text regions and background clutters or the non-text regions produced in Component Analysis stage. In Chain Analysis, a two-layer filter is devised to eliminate these false positives. The first layer includes some heuristic rules about Uyghur text chains. The chain-level pruning rules are different from the component-level rules in the Component Analysis stage. In terms of the usual properties of Uyghur text chains, the chains with too small or big bounding rectangles would be deleted in the first layer, which are always be non-text chains. What's more, in the Chain Linking stage, the bounding boxes of some chains may be changed. Their aspect ratio values become too large instead of the appropriate ones because of the connection of some non-text regions or the influence of complex background. The function of these simple rules is to prune the simple non-text chains and these chains which connected some background objects. The details of them are discussed in 3.6. In the second layer, a Random Forest [8] classifier is adopted to identify the remaining chains of the first layer. An appropriate set of chain level gradient features, describing the differences in textural attributes between text chains and non-text chains, are used to train the classifier.

The procedure of feature extraction is:

1) All the chains are resized to 26×122 pixels. For color images, the gradient value of each pixel from the 2nd column to 25th column and from the 2nd row to 121th row of the R, G and B channel is separately calculated. The gradient of the pixel in the color image is assign as the one with the largest norm. With this calculation, a two-dimension array whose size is 24×120 with the gradient value is established.
2) L_2-norm is applied to normalize the gradient value:

$$v \rightarrow v \Big/ \sqrt{\|v\|_2^2 + \varepsilon^2}, \tag{3}$$

where v is the un-normalized gradient vector and the factor ε is set to be 0.1.
3) Pooling is conducted. Consequent 2-by-2 patches are extracted, and the max and min values in these patches are selected to form the input vector.
4) 1440 features in total are collected from the chain.

The probability of one chain is the fraction of votes for positive text from the trees, which is a double type. The chains with probabilities lower than a pre-defined threshold θ (set to be 0.3) are eliminated. For chain repetitions, if the intersection area of one chain and the other chains is larger than 0.9 of their union area, these chains are regarded as repetitions. Then every chain is examined. If a chain is repeated with the others, the one with the largest area is kept and the others are deleted.

With all the above process, the remaining chains are the final text regions.

Fig. 6. Examples of IMAGE570 training subset.

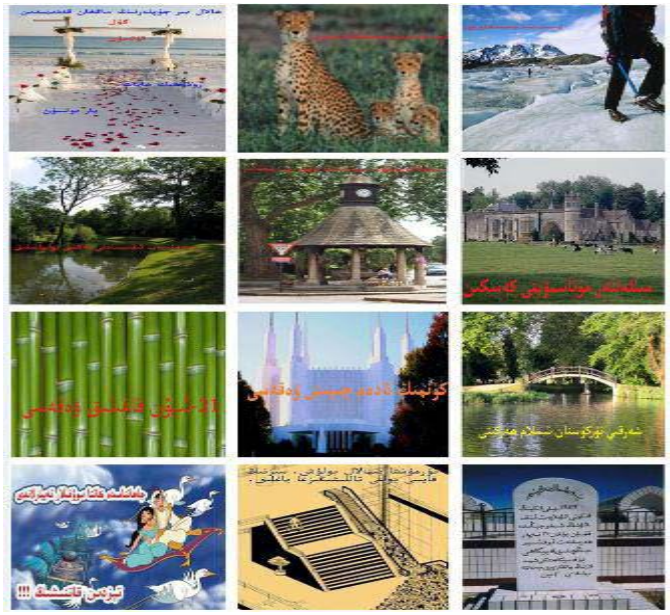


Fig. 7. Examples of IMAGE570 testing subset.

F. The Heuristic Rules of Uyghur Text Chains Filter

1) The Height: If the height value of the bounding rectangle is too small or too large, the chain would be discarded. Only when the height value is more than 10 pixels and less than 300 pixels, the chain is accepted.

2) The Width: If the width value of the bounding rectangle is too small, the chain would be discarded. The minimum value of the width we could accept is 10 pixels.

3) Aspect Ratio: Aspect ratio is the value of bounding rectangle's height dividing its width. If respect ratio is greater than 4, the chain would be discarded.

4) The Area: If the bounding rectangle area of one text chain is greater than 50 percent of the area of the image, the chain would be discarded.

IV. Dataset and Evaluation Protocol

In this section, a new dataset is built to evaluate our system, which contains images with complex background. The evaluation method is very similar as the one proposed in [10].

As the existing benchmark dataset all focus on commonly used languages [36], a new evaluation dataset with horizontal texts is built. The vast majority of texts in the images are Uyghur, and the few rest texts are Chinese or English. They are all in deferent fonts, sizes and colors. As the dataset contains 570 images in total, it is called as IMAGE570, and all the images are crawled from the Web. IMAGE570 is divided into two subsets: 370 images are assigned to the training subset and the remaining 200 images are assigned to the testing subset. All the images in this dataset are fully annotated. Fig.6 shows some examples of the training subset, and Fig.7 shows some examples of the testing subset.

IMAGE570 is of big challenge, due to both the diversity of the texts, the complexity of the image backgrounds and the integration of texts and backgrounds. Each image in IMAGE570 may contain one or more Uyghur text lines. The image backgrounds may involve repeated patterns (e.g. buildings and fences), vegetation (e.g. grasses and trees), and diverse scenes (e.g. flowers, street and natural landscape). All these complex backgrounds are easy to confuse with the texts, so it challenging for text detection.

To mark the text lines on one image, a rectangle around every detected text should be drawn out. However, it is a problem that how to judge whether a text line is correctly detected. For the estimated rectangle D and the ground truth rectangle G, the overlap ratio is calculated to judge whether D is correctly detected. The overlap between D and G is defined as:

$$r(G, D) = \frac{A(G \cap D)}{A(G \cup D)}, \tag{4}$$

where $A(G \cup D)$ and $A(G \cap D)$ represent the areas of the union and intersection of G and D. According to the overlap ratio between the estimated rectangles and the ground truth rectangles, we can determine the detections are true or false positives. If the overlap ratio is larger than 0.5, the estimated rectangle is considered as a correct detection. For single-line Uyghur text, if there are more than one estimated rectangle but they are not repeated, D is the union of these estimated rectangles. If there have repetitions, one of them is counted and the others are regarded as wrong estimations. For the multi-line texts, G is the union of all the line contained in D. The definition of precision (p) and recall (r) are:

$$\begin{aligned} p &= |TP| / |E| \\ r &= |TP|/|T|, \end{aligned} \tag{5}$$

Where TP is the true positive detection set, and E and T are the sets of estimated rectangles and ground truth rectangles, respectively. The F-measure (f), as a general measure of system performance, is calculated with the precision and recall:

$$f = 2pr/(p + r). \tag{6}$$

TABLE I
PRECISION, RECALL AND F-MEASURE (%) ON IMAGE570 SET

	PRECISION	RECALL	F-MEASURE
OUR METHOD	**85.3**	**84.7**	**85.0**
YIN[1]	76.0	75.0	75.5

V. EXPERIMENTS

In this section, extensive experimental comparisons are conducted to evaluate the proposed system. Firstly, the overall performance of our method and the state-of-the-art methods are compared on our dataset IMAGE570 in section 5.1. Then, evaluation analysis is carried out in section 5.2, to investigate the influences of various parameters in the Component Analysis stage. Finally, experiments are presented to test the impact of the free threshold in the Chain Analysis stage in section 5.3.

A. Performance Comparison on Dataset IMAGE 570

The performance is assessed by the proposed evaluation protocol. We compare our system with the method proposed by Yin [1], which is the best multi-language detection method at present, on IMAGE570. As we only focus on Uyghur text detection, our system is only trained on Uyghur language. So the precision and recall in the table I are only regarding on Uyghur text detection.

From table I, we can observe that the precision, recall and F-measure of our method all have higher values than Yin's method [1]. Compared to the Yin's method, the F-measure of our method can be increased by 12.6%. The improvement of our method mainly attributes to the following factors:

1) The newly designed channel-enhanced MSERs algorithm is very effective and can detect most text regions. Also, the extension strategy in Chain Linking stage expands the incomplete chains to complete ones. These two processes guarantee the high recall of our system.
2) The two-layer filtering mechanism in Component Analysis can effectively distinguish the text components from non-text components. Meanwhile, the two-layer chain elimination filter in Chain Analysis can precisely determine the text regions. These two processes guarantee the high precision of our system.

Fig.8 presents a few successful detection cases on several challenging images. We can see that our system is effective in dealing with large variation in texts, including diverse font size, low contrast, and very complexity backgrounds. Fig.9 displays two failure examples. Through a step by step analysis, we get the reasons for failure. For the left image, a part of text regions are regarded as noises by the classifier in Component Analysis, as the influence of water shimmer. Thus, the existing chain cannot be expanded by the extension operation in Chain Linking. For the right image, the candidate regions are removed by the two-layer filtering mechanism in Chain Analysis, as the change of bounding boxes.

B. Evaluation Analysis on Component Elimination

In the Component Analysis stage, a two-layer filtering mechanism is designed. The first layer includes some heuristic

Fig. 8. The successful detection cases.

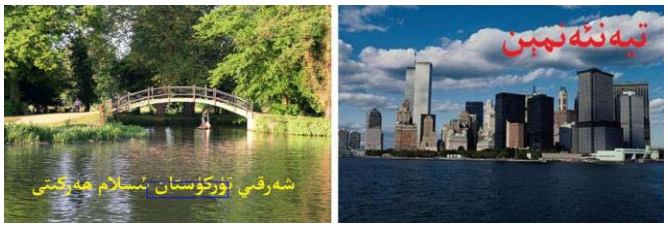


Fig. 9. The failure detection cases.

TABLE II
PERFORMANCE (%) ON IMAGE570 SET

	ACCURACY	PRECISION	RECALL	F-MEASURE
24×24	98.20	85.20	78.40	81.70
24×32	**98.21**	**85.30**	**84.70**	**85.00**
32×32	98.19	85.60	83.10	84.30

rules, and the second layer is a SVM classifier with a polynomial kernel trained by 12000 samples using HOG features. All the training samples are divided into two sets: the positive set including 6000 samples and the negative set including 6000 samples. There are some repetitions in the positive set. Most of the positive images contain one word, and few images contain two or more words. The negative images contain all kinds of background objects with different fonts. When doing training, the samples are resized into the same size. The performance of the classifier can affect the result of the system directly. If the component classifier could prune as many noises regions as possible, the precision of the system would be higher. But in order to extract the HOG features, the MSERs are resized into the same size. However, if the size is set too large or small, it always loose word information. In the experiment, the size of the training samples is resized to 24×24, 24×32 and 32×32 ($height\times width$) three dimensions to compare the accuracy on pruning MSERs, the system precision and recall. The accuracy is computed by counting the MSERs pruned by the two-layer component classifier in the test set.

As illustrated in Table II, the accuracy of pruning MSERs is almost same in the 24 × 24, 24 × 32 and 32 × 32 three

TABLE III
ACCURACY (%) OF LINEAR SVM, RBF SVM AND POLY SVM

	5000	6000	7000	8000	9000	10000	11000	12000
LINEAR	92.05	90.55	90.60	89.95	88.55	87.35	84.35	83.80
RBF	88.20	89.95	90.85	91.10	92.90	93.75	93.30	94.20
POLY	**96.35**	**96.70**	**96.80**	**96.95**	**97.35**	**97.70**	**97.60**	**97.95**

Fig. 10. Samples of word region images in SVM test set.

Fig. 11. Samples of non-word region images in SVM test set.

size, but the system have the highest F-measure value when the regions are set to 24×32 size.

In the Component Analysis, the SVM classifier is used to purify the candidate MSERs. To examine the mutual relations between the classifier accuracy with different kernels and the quantity of training samples, 2000 image patches are extracted from the testing subset of IMAGE570, to construct SVM testing set, which is divided into two subsets, word region subset and non-word region subset. The word region subset includes regions with one word, regions with two or more words, regions with word and background and regions with incomplete word. The word region subset has 1000 images, and some examples of word region subset are showed in Fig.10. The non-word region set includes architecture regions, plant regions, people regions and other regions. Different non-word regions are different in sizes and backgrounds, in total 1000 images. Some examples of non-word regions are showed in Fig.11. The SVM classifier is trained with LINEAR kernel, Polynomial (POLY) kernel and Radial Basis Function (RBF) kernel respectively, using different number of images selected from the training subset of IMAGE570. As described above, the SVM training images include two kinds, word region images and non-word region images. The amount of SVM training images ranges from 5000 to 12000, accompanied by the change of the accuracy of SVM classifier.

As shown in Table III, POLY SVM obtains the highest accuracy, while the accuracy of LINEAR SVM decreases with the increase of images and the accuracy of RBF SVM grows but is lower than that of POLY SVM in all cases. The details of changing are showed in Fig.12. From Table III, we can see that the accuracy of Poly is higher than that of the Linear and RBF all the time, and the highest accuracy of Poly SVM reaches

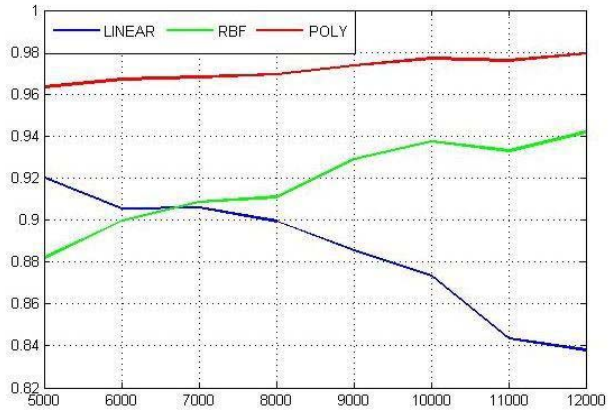


Fig. 12. Accuracy of SVM with LINEAR, POLY and RBF kernel. The accuracy of POLY and RBF grow with the increase of training images. But the accuracy of LINEAR decreases with the increase of training images.

Fig. 13. Positive samples in chain training set.

to 97.95%. When the number of images is 5000, the Linear SVM achieves the highest accuracy to 92.05%. When the number of images is 12000, the RBF SVM has the highest accuracy of 94.20%. But, both of these two values are lower than the highest accuracy of Poly SVM.

C. Experiments on Chain Elimination

In the Chain Analysis, a Random Forest classifier is trained by the gradient features of images in the second layer of the chain filter. The chain training set has 8387 samples, including 3510 positives and 4877 negatives. In the positive samples, most images contain two or more words and some images are repeated. In the negative samples, the objects are extracted randomly. Some examples are showed in Fig.13 and Fig.14. A free threshold θ controls the probability of the classifier to decide whether the chain is a text chain. If the threshold is changed, the precision and recall value would be changed accordingly. An experiment is conducted to analyze the relationship between θ and the system result. Keeping other parameters constant, we change the θ, and get the

Fig. 14. Negative samples in chain training set.

TABLE IV
INFLUENCE (%) OF DIFFERENT θ

θ	PRECISION	RECALL	F-MEASURE
0.2	73.4	88.2	80.1
0.3	**85.3**	**84.7**	**85.0**
0.4	89.7	77.1	82.9
0.5	93.8	61.4	74.2

changed precision, recall and f-measure separately. The results are showed in Table IV.

As showed in Table. IV, with the increase of θ, the precision increases and recall decreases. When θ is set to be 0.5, the precision is the highest 93.8% and the recall is the lowest 61.4%. When θ is set to be 0.2, the recall is the highest 88.2% and the precision is the lowest 73.4%. When θ is equal to 0.3, the F-measure value is the highest 85.0%. In this paper, the θ is set to be 0.3.

VI. CONCLUSION

In the paper, an effective Uyghur text detection system in complex background images is proposed, which can accurately extract text in traffic scene for intelligent vehicles. The main contribution of the system lies in designing channel-enhanced MSER algorithm, to locate almost all the component candidates and having highly discriminative capability combined with the characteristics of Uyghur to distinguish texts from various non-text components. The excellent performance on the challenging dataset convincingly verifies the performance of the proposed method and the efficiency of text extraction for intelligent vehicles. The system could not only be used on the detection of Uyghur Language, but also on the detection of other language with changing some heuristic rules and parameters.

REFERENCES

[1] 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, no. 5, pp. 970–983, May 2014.

[2] Q. Ye, and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015, doi: 10.1109/TPAMI.2014.2366765.

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

[4] G. Chen, "Large-scale visual font recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3598–3605.

[5] *Uyghur Language*. Accessed: Sep. 3, 2017. [Online]. Available: http://en.wikipedia.org/wiki/Uyghur_language

[6] H. Al-Yousefi and S. S. Udpa, "Recognition of arabic characters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 8, pp. 853–857, Aug. 1992.

[7] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.

[8] B. Leo, "Random forests," *Mach. Learn.*, vol. 45, no. 6, pp. 422–432, 2001.

[9] J. Matas, O. Chum, T. Pajdla, and M. Urban, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, Sep. 2002.

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

[11] H. Zhang, K. Zhao, and Y. Z. Song, "Text extraction from natural scene image: A survey," *Neurocomputing*, vol. 122, no. 51, pp. 310–323, 2013.

[12] K. I. Kim, K. Jung, and J. H. 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.

[13] X. Liu and J. Samarabandu, "Multiscale edge-based text extraction from complex images," in *Proc. IEEE Int. Conf. Multi-Media Expo*, Jul. 2006, pp. 1721–1724.

[14] N. Mavaddat, T.-K. Kim, and R. Cipolla, "Design and evaluation of features that best define text in complex scene images," in *Proc. IAPR Conf. Mach. Vis. Appl.*, 2009, pp. 94–97.

[15] L. Sun, Q. Huo, and W. Jia, "A robust approach for text detection from natural scene images," *Pattern Recognit.*, vol. 48, no. 9, pp. 2906–2920, 2015.

[16] R.-J. Jiang *et al.*, "Using connected-components' features to detect and segment," *J. Image Graph.*, vol. 11, no. 11, pp. 1653–1656, 2006.

[17] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, vol. 157. no. 10, pp. 3538–3545.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Jun. 2005, pp. 886–893.

[19] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced MSER trees," in *Computer Vision—ECCV 2014* (Lecture Notes in Computer Science), vol. 8692, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014.

[20] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 3304–3308.

[21] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *Proc. Int. Conf. Image Process.*, Sep. 2011, pp. 2609–2612.

[22] M. Anthimopoulos, B. Gatos, and I. Pratikakis, "A two-stage scheme for text detection in video images," *Image Vis. Comput.*, vol. 28, no. 9, pp. 1413–1426, 2010.

[23] Q. Ye, Q. Huang, D. Zhao, and W. Gao, "Fast and robust text detection in images and video frames," *Image Vis. Comput.*, vol. 23, no. 6, pp. 565–576, 2005.

[24] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 385–392, Apr. 2000.

[25] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Computer Vision—ACCV*. Berlin, Germany: Springer, 2010, pp. 770–783.

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

[27] P. Shivakumara, T. Q. Phan, and C. L. Tan "New Fourier-statistical features in RGB space for video text detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1520–1532, Nov. 2010.

[28] H. William, S. A. Teukolsky, A. Saul, W. T. Vetterling, and B. P. Flannery, "Section 16.5. Support vector machines," in *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. New York, NY, USA: Cambridge Univ. Press, 2007.

[29] Q. Ye, J. Jiao, and J. Huang, "Text detection and restoration in natural scene images," *J. Vis. Commun. Image Represent.*, vol. 18, no. 6, pp. 504–513, 2007.

[30] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 2558–2567.

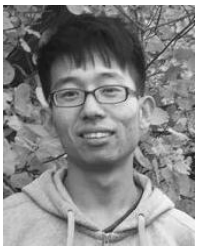
[31] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proc. ICCV*, 2013, pp. 1241–1248.

[32] R. Wang, N. Sang, and C. Gao, "Text detection approach based on confidence map and context information," *Neurocomputing*, vol. 157, pp. 153–165, Jun. 2015.
[33] H. Xie, Y. Zhang, J. Tan, L. Guo, and J. Li "Contextual query expansion for image retrieval," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 1104–1114, Jun. 2014.
[34] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, and Q. Dai, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, to be published.
[35] W. Wang, D. Zhang, Y. Zhang, and J. Li, "Robust spatial matching for object retrieval and its parallel implementation on GPU," *IEEE Trans. Multimedia*, vol. 13, no. 6, pp. 1308–1318, Dec. 2011.
[36] J. Bai, Z. Chen, B. Feng, and B. Xu, "Chinese image text recognition on grayscale pixels," in *Proc. IEEE ICASSP*, Florence, Italy, May 2014, pp. 1380–1384 .
[37] J. Bai, Z. Chen, B. Feng, and B. Xu, "Image character recognition using deep convolutional neural network learned from different languages," in *Proc. IEEE ICIP*, Pairs, France, Oct. 2014, pp. 2560–2564.
[38] Q. Wang, J. Fang, and Y. Yuan, "Adaptive road detection via context-aware label transfer," *Neurocomputing*, vol. 158, pp. 174–183, Jun. 2015.
[39] Y. Yuan, Z. Xiong, and Q. Wang, "An incremental framework for video-based traffic sign detection, tracking, and recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1918–1929, Jul. 2017.
[40] Y. Yuan, D. Wang, and Q. Wang, "Anomaly detection in traffic scenes via spatial-aware motion reconstruction," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1198–1209, May 2017.

Chenggang Yan received the B.S. degree in computer science from Shandong University in 2008, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2013. He was an Assistant Research Fellow with Tsinghua University. He is currently a Professor with Hanzhou Dianzi Univeristy. He has authored or co-authored over 30 refereed journal and conference papers. His research interests include machine learning, image processing, computational biology, and computational photography.

He co-authored and received the Best Paper Awards in International Conference on Game Theory for Networks 2014, and SPIE/COS Photonics Asia Conference 9273 2014, the Best Paper Candidate in International Conference on Multimedia and Expo 2011.

Hongtao Xie received the Ph.D. degree in computer application technology with the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2012. He is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences, China. His research interests include multimedia content analysis and retrieval, similarity search, and parallel computing.



Shun Liu received the B.E. degree in computer science and technology from Shandong University, Weihai, China, in 2013. He is currently pursuing the Ph.D. degree with Shandong University, Weihai, China. His research interests include multimedia content analysis and retrieval, similarity search.

Jian Yin received the Ph.D. degree in Shandong University. He is currently an Associate Professor with Shandong University, Weihai, China. His research interests include computer software and theory, computer graphics.

Yongdong Zhang (M'08–SM'13) received the Ph.D. degree in electronic engineering from Tianjin University, Tianjin, China, in 2002. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His current research interests are in the fields of multimedia content analysis and understanding, multimedia content security, video encoding, and streaming media technology.

He has authored over 100 refereed journal and conference papers. He was a recipient of the Best Paper Award at the PCM 2013, ICIMCS 2013, and ICME 2010, the Best Paper Candidate at the ICME 2011. He serves as an Editorial Board Member of *Multimedia Systems Journal* and *Neurocomputing*.

Qionghai Dai (SM'05) received the B.S. degree in mathematics from Shanxi Normal University, Xian, China, in 1987, and the M.E. and Ph.D. degrees in computer science and automation from Northeastern University, Shenyang, China, in 1994 and 1996, respectively. He has been a member of the Faculty, Tsinghua University, Beijing, China, since 1997. He is currently a Cheung Kong Professor with Tsinghua University and also the Director with the Broadband Networks and Digital Media Laboratory. His current research interests include signal processing and computer vision and graphics.