Detection and Segmentation Text from Natural Scene Images Based on Graph Model

XIAOPEI LIU[1,2], ZHAOYANG LU[1], JING LI[1], WEI JIANG[1]

[1]State Key Laboratory of Integrated Services Networks
[1]XiDian University
[1] Xi'an 710071
CHINA
[2]School of Communication and Information Engineering
[2]Xi'an University of Science and Technology
Xi'an 710054
CHINA
Liuxiaopei2007@163.com

Abstract: -This paper presents a new scheme for character detection and segmentation from natural scene images. In the detection stage, stroke edge is employed to detect possible text regions, and some geometrical features are used to filter out obvious non-text regions. Moreover, in order to combine unary properties with pairwise features into one framework, a graph model of candidate text regions is set up, and the graph cut algorithm is utilized to classify candidate text regions as text or non-text. As for segmentation, a two-step technique for scene text segmentation is proposed. Firstly, the K-Means cluster algorithm is employed in color RGB and HSI color space respectively, and the better result is selected as initial segmentation. Then in minimum energy framework, graph cut is employed for re-labeling verification. Experimental results show the satisfactory performance of the proposed methods.

Key-Words: - scene text detection text segmentation stroke width Hog feature Graph model

1 Introduction

With the wide use of smart phones and rapid development of the mobile internet, it has become a living style for people to capture information by using of cameras embedded in mobile terminals. Text information as a main component of scene images, it usually provides an important clue for scene understanding. Consequently, extracting text embedded in natural scene has gained more and more attention from researchers, and it has become one of the hottest topics in the area of document analysis and recognition [1].

Automatic extracting text from natural scene images usually includes three parts: text location, text segmentation and text recognition. As we all know, OCR technology has gained great success in the area of characters processing, and characters extracted (binary text) can be well recognized with the aid of current OCR techniques. Consequently, this paper mainly addresses the issues of scene text location and segmentation.

Up to now, automatic extracting text from scene text images is still an open problem. The main difficulty lies in the high variability of text appearance, for instance, it varies in color, font style, size and different languages. In addition, complex background, uneven illumination and blur often make the problem of scene text extraction much more challenging. Researchers have reported many methods to solve this problem, and some can give good results. Nevertheless, it is far from satisfactory for practical applications.

In this paper, a text extraction system scheme for natural scene images is proposed, which includes text location and text binarization, mainly aiming at Chinese characters in natural scene images. Firstly, a connected component (CC) based text location method is proposed, in which the text stroke edge is used to detect potential text CCs, the unary properties and the context information of CCs are combined into one framework, thus a graph of CCs is set up, and graph cut algorithm is then employed to classify candidate CCs as text or not. Secondly, for text segment, a two-step binarization method is adopted to deal with the problems of uneven illumination and clutter background, which greatly affect the performance of recognition.

The rest of this paper is organized as follows. Recent scene text detection and segmentation methods are reviewed in section 2. Section 3 makes a description of the proposed text location method. Section 4 introduces the segmentation algorithm. Section 5 discusses the experiments and result. In the final section the conclusions are drawn.

2. Related Works

Automatic extraction of text from natural scenes is a very challenge task. The primary difficulty lies in the variety of text, such as font, size, arrangement, illumination, complex background, and so on. Recently, there have been proposed many methods to deal with text location and segmentation in natural images and videos, comprehensive surveys can be found in [1-2]. Accordingly, we only give a brief review of current methods related to our job.

2.1 Text Location

Text location methods can be roughly classified into region-based methods and connected component based methods. Region-based methods consider text as a special texture, which usually use FFT, DCT, wavelet, Gabor filter, and etc. to extract texture features at first, then use a sliding window to search for possible text blocks through out the whole image, and a classifier is finally used to verify it as a text region or not. Ye et al. [3] used the wavelet energy feature to detect all possible text regions, and adopted SVM to classify potential text regions as text or non-text. Palaiahnakote[4] used Fourier-Statistical Features to detect video text, and K-means clustering to classify text pixels from the background. Yi[5] etc. used Gabor filter to describe the stroke component in text character.

Different from the region-based methods, CC-based methods work according to a bottom-up mode, which usually use edge, color, MSERs (Maximally Stable Extremal Regions), and etc. to extract connect components, then heuristic rules or classifiers are employed to analyze CCs. Epshtein et al[6] proposed to measure stroke width for each pixel and merge its neighbors into CCs, which formed letter candidates. Chen[7] detects MSERs as the basic CCs, then geometric and stroke width information are applied to erase non-text CCs. Recently, the thought of combining context information with unary properties of CCs together is introduced to filter non-text CCs. For example, Pan [8] combined unary properties and neighborhood information into a CRF framework to erase non-text CCs. Zhou[9] utilized a hierarchical model consisting of unary classifier and pairwise classifier in cascade to filter out the non-text CCs. Shi[10] constructed a graph for potential CCs, and employed max flow algorithm to label text CCs and non-text CCs in the framework of energy minimization.

2.2 Text Segmentation

The segmentation of foreground text and complex background from natural scene images is a challenging problem. Main methods include threshold based methods[11-12], color cluster based methods[13-14] and statistic model based methods[15-16]. The former two kinds of methods adapt to process relative simple text images, but to complex text images, such as uneven light, low contrast, clutter background, and etc., they usually give unsatisfactory result. Recently, statistic model based method is applied to deal with the problem of complex text segmentation. Ye[15] etc. made a GMMs model for foreground text. Mishra[16] proposed a MRF based method for scene text segmentation and obtained satisfactory result in selected dataset. But in occasions of uneven light or low contrast, the algorithm failed. In addition, it is time-consuming.

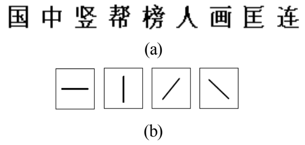
Although some methods mentioned above achieved promising performance in their own applications, none of them can extract text from images with clutter background efficiently. Considering of the fact that characters in same arrangement direction have similar color, size and stroke width, and inspired by [8-10], we introduce the context information of image to detect and segment text from clutter images, which is a necessary supplement for unary properties of CCs or pixels.

3 Text Detection and Location

The aim of text detection is to find all of text regions quickly and efficiently, so we use stroke edge to detect potential text connected components (CCs) firstly. Then heuristic rules and classifier are employed to ride off non-text CCs.

3.1 Text CCs Detection

Chinese characters are usually composed of a set of strokes in directions of horizontal, vertical, up-right and up-left (as shown in Fig.1). Therefore, we define a 3×3 edge detection mask (as shown in Fig.2) to detect stroke edges.



(a) Examples of Chinese characters (b) Stroke directions

Fig.1 Chinese Characters examples and stroke directions

P_1	P_2	P_3
P_4	P	P_5
P_6	P_7	P_8

Fig. 2 Mask of stroke edge detection

P is the pixel in the center of the mask, its edge strength E_p in one color channel can be computed as:

$$E_p = \max\{|p_8 - p_1|,|p_7 - p_2|,|p_6 - p_3|,|p_5 - p_4|\} \quad (1)$$

Where, P1~P8 are eight neighboring pixels of pixel P, In RGB color space, the edge strength of pixel P is the maximum of three channels.

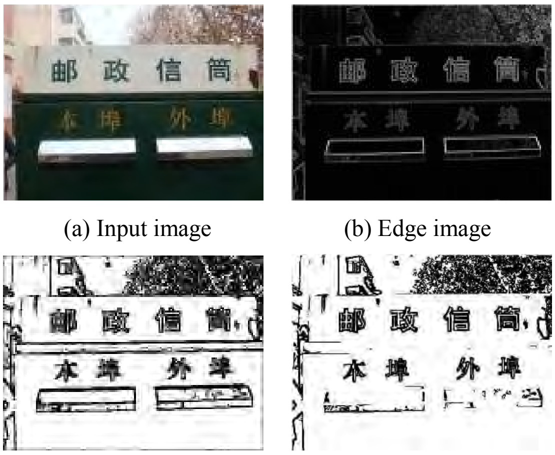
$$E_{pc} = \max\{E_{pR,} E_{pG}, E_{pB}\} \quad (2)$$

Where E_{pR}、E_{pG} and E_{pB} represent the edge strength of R, G and B channel of pixel P respectively.

To avoid losing possible text regions in conditions of uneven light and low contrast, an adaptive method of binarization is adopted to get binary edge image. Then we remove some line structure, such as long straight line, short straight lines and isolate noises. Based on this processing, we use a CC region label algorithm to find all CCs. In order to eliminate obviously non-text CCs, several empirical rules are defined as follows.

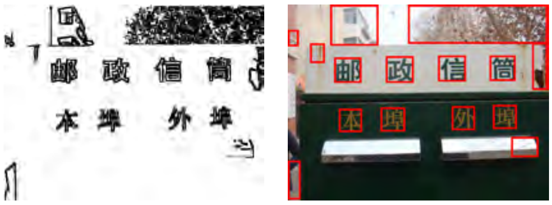
- $Area < MinArea$. To remove too small CCs.
- $Area > MaxArea$. To remove too big CCs.
- $\min(h,w)/\max(h,w) < Min\_narrow$. To remove too narrow CCs.

The procedure of CCs detection is shown in Fig.3, which detailed in our previous work [17]. In Fig.3 (f), possible text CCs are marked by red rectangle.



(a) Input image (b) Edge image

(c) Binary edge image (d) Processed binary image



(e) Filter non-text CCs (f) Detection Result

Fig. 3 The procedure of text CCs detection

3.2 Graph based text CCs analysis

Stroke edge-based heuristic methods can detect text CCs efficiently. However, there exist a lot of non-text regions with strong edges in the background, such as leaves, windows, buildings and so on. We can see it produces many false alarms, as shown in Figure 3(d). In order to reduce false alarms, general methods usually use a learning based classifier such as SVM, MLP, BP ,and etc., to verify each CC as text or not. But these classifiers merely employ individual component properties of text component, they are prone to misclassify components when the image is clutter and noisy. To solve this problem, context relevant information of candidate CCs is introduced, which is a supplement for individual component properties of text component.

3.2.1 Graph constructed for CCs

Classifying candidate CCs as text CCs or non-text CCs can be viewed as a problem of binary labeling. While graph cut is one of popular algorithms for image binary label. Consequently, we employ graph based method to label candidate CCs. In the application of binary label, image is mapped to a weighted undirected graph, and pixels are seen as nodes. The optimal label result is obtained by using of min-cut algorithm, which actually converts image label into an optimization problem. So we map the initial detection result (seen in Fig.3(e)) into a graph.

Suppose $G=<V,E>$ is an undirected graph constructed for CCs, each CC is the vertex (V) of G and undirected edges (E) connect these vertices. Classifying each CC as text or non-text is to assign each CC as 1 or 0, where 1 represents text, and 0 represents non-text.

We use x_i to denote the CC i, and N_i to represent the neighborhood CC set of CC i. The rules of neighborhood relationship graph built are shown as follows.

$$\left.\begin{array}{r} d(x_i,x_j) < 2\times\min[\max(w_i,h_i),\max(w_j,h_j)] \\ \theta(x_i,x_j) < \pi/4 \end{array}\right\} \quad (3)$$

$$\Rightarrow x_i \in N_j, x_j \in N_i$$

Where

$$d(x_i,x_j)=\sqrt{(c_x^i-c_x^j)^2+(c_y^i-c_y^j)^2}\ ,$$

$$\theta(x_i,x_j)=\arctan(\frac{c_y^j-c_y^i}{c_x^j-c_x^i})$$

c_x^i and c_y^i indicate the center position of CC i, w and h are the width and height of the CC respectively.

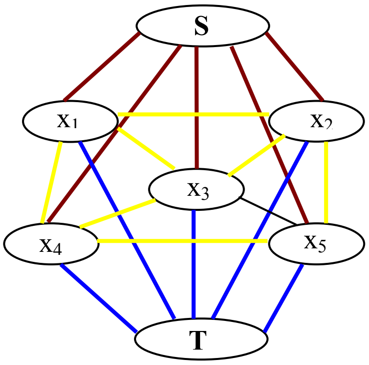


Fig.4 The structure of graph

Based on the analysis above, a graph is constructed as Fig.4. There are two special terminals connecting all CCs, the weight value of each edge is non negative, which connects the neighbor CCs. Set the number of CC is M, and $X=\{x_1,x_2,\cdots,x_M\}$ is the set of all candidate components, $L=\{l_1,l_2,\cdots,l_M\}$ is the set of corresponding labels of set X, where $l_i\in\{0,1\}$ (0 is background, 1 is foreground or text). N represents the set of neighborhood components pairs. We define the cost function as below.

$$\begin{aligned} E(L) &= R(L) + \lambda B(L) \\ &= \sum_{s \in S} R_s(l_s) + \lambda \sum_{s,t \in N} B_{s,t} \cdot \delta_{l_s \neq l_t} \end{aligned} \quad (4)$$

where

$$\delta_{l_s \neq l_t} = \begin{cases} 0 & if\, l_s = l_t \\ 1 & if\, l_s \neq l_t \end{cases}$$

$R(L)$ is the unary term, and $R_s(l_s)$ represents the individual penalties for assigning vertex s to foreground (text) or background, S is the set of all the vertices. $B(L)$ is pairwise term, and coefficient $B_{s,t}$ is penalty for a discontinuity between neighborhood nodes s and t. The coefficient λ is a relative important factor between unary term and pairwise term. In the following section, we introduce the unary and pairwise cost function in detail.

3.2.2 Unary term

Unary cost measures the individual penalty for assigning each CC (node) to foreground or background. If the probability of node p belonging to foreground is large, the unary cost of this node is very small, and vice visa. Foreground and background connect each node with unary cost R(1) and R(0) respectively. SVM is selected to estimate the probability of each node belonging to foreground or background due to its good generalization ability. However, SVM doesn't produce good probability estimation because of its poor square error and cross entropy. To correct for poor calibration, a platt calibration method is proposed in [19], which use sigmoid model to map the output of SVM to posterior probability.

$$p(l_i = 1 \mid f_i) = \frac{1}{1 + \exp(Af_i + B)} \quad (5)$$

Where f_i is the output of SVM. The parameters A and B are fit using maximum likelihood estimation from a training set (f_i, x_i)

Thus, we define the unary cost as following:

$$\begin{cases} R_p(1) = 1 - p(l_p = 1 \mid f_p) \\ R_p(0) = p(l_p = 1 \mid f_p) \end{cases} \quad (6)$$

In order to estimate posterior probabilities of text components, four types of features are used to train SVM, which are defined as following:

·Edge density. This feature is defined as the ratio between edge number to the area of the CC, which filters out the CCs with too few edge pixels or too many edge pixels.

$$Edge\_dendity = \frac{Num_{edge}}{Area_{cc}} \quad (7)$$

·Compactness. It is defined as the ratio between the bounding box area and the square of component's perimeter, which used to remove the non-text components with too complex contour.

$$Compactness = \frac{Area_{\mathrm{CC}}}{perimeter^2} \quad (8)$$

·HOG features. Each CC is divided into 4 blocks, and every block has 4 cells, extracting 8-orientation histogram of oriented gradients.

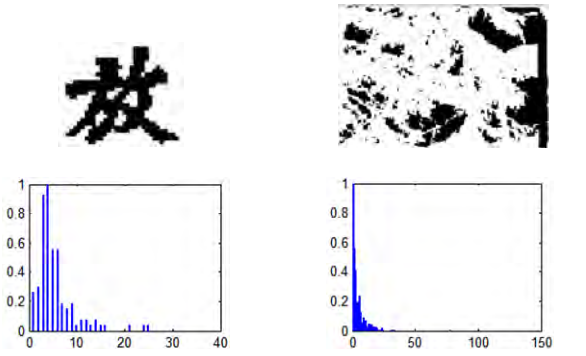
·Stroke feature. Text components have uniform stroke width. In this paper, we use run-length histogram to compute the stroke width of text component CC.

Examples of run-length histograms for text and non-text CCs are shown in Fig.5. As we can see, stroke run length frequency is highest at 4, which reflects the average stroke width of characters in Fig.5 (a) is 4. Whereas the highest frequency in the run length histogram for non-text region is unit run-length, which reflects texture mode of non-text regions is disorderly, and doesn't have the characteristics of the texture of the text mode. In order to represent this property of text component, two features are defined as following.

$$SW = \arg\max_{i \in I} RH(i), i \neq 1 \tag{9}$$

$$URN = \frac{RH(1)}{\max_{i \in I} RH(i), i \neq 1} \tag{10}$$

SW indicates stroke width, i represents for run length, $i \in I$, $I = \{1, 2, \cdots, L\}$, where L is the longest run to be counted. $RH(i)$ denotes a run-length histogram. URN is defined as the ratio between the number of unit runs and the maximal number of runs excluding unit run, which used to remove non text CCs.



(a) Text Region (b) Non-text Region

Fig.5 Stroke Run Length Histogram of text and non-text regions

3.2.3 Binary cost term

The form of text in natural scenes images is difficult to control, as a result, if only unary features are merely considered, it always cannot obtain satisfactory result. Binary term $B(L)$ can be as a complement for improving the performance. $B_{\{s,t\}}$ decreases as a function of distance between neighborhood CC$_s$ and CC$_t$. Normally, it is small when CC$_s$ and CC$_t$ are very different, and it is large when the two CCs are similar. According to features of texture and geometric, $B_{\{s,t\}}$ can be defined as following.

$$B_{\{s,t\}} = \exp(-\frac{(SW_s - SW_t)^2}{2\sigma^2}) + \exp(-\frac{dist_c}{2\sigma^2}) \tag{11}$$

where SW_s and SW_t are stroke width of CC$_s$ and CC$_t$ respectively, $dist_c$ denotes color distance between CC$_s$ and CC$_t$.

This function means that if the stroke width and color characteristic of the two neighboring areas are similar, the cost of assigning different labels is large and vice versa.

By now weights of all edges have been determined, thus the constructed graph is defined. Labeling a CC as text or non-text CC can be achieved by minimizing the energy function through minimum graph cut. Boykov and Jolly[18] have shown that the minimized energy can be computed by the min-cut through max-flow.

3.3 CC grouping

Chinese character often has several isolate constituent parts as shown in Fig.6. Since this, a CC perhaps is not a single character, but a part of it. If the CC is recognized directly, it is sure to give erroneous results. As a result, we use some rules to group CCs parts into a character.

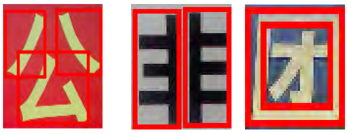


Fig.6 Constituent elements of some characters

Text CCs are usually similar in color, height and width, according to the structure of Chinese characters, we define the rules as following.

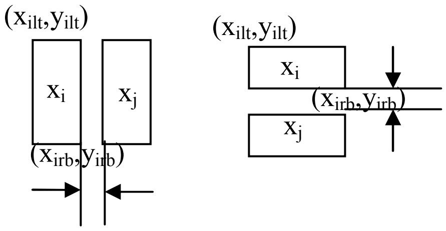


Fig.7 Layout of text CCs for merging

i) Neighborhood CCs merging:

$$\begin{aligned}&|x_{jlt}-x_{irb}|<d_1\,\&\max(H_i,W_i)/\min(H_j,W_j)>T_1\\ or&|y_{jlt}-y_{irb}|<d_1\,\&\max(H_i,W_i)/\min(H_j,W_j)>T_1\end{aligned} \quad (12)$$

Where, (x_{ilt}, y_{ilt}) and (x_{irb}, y_{irb}) represent coordinates of the top left corner and the right bottom corner of CC i respectively, H is the height of CC, while W is the width of CC.

ii) If a smaller CC is included in a larger CC, we remove the smaller one.

The procedure of CC merging is performed between neighborhood CCs, we make an analysis on the neighborhood CCs of each CC based on the rule (3) defined in 3.2. Therefore, we only need judge the relationship of each CC and its neighborhood CCs. The whole merging procedure is shown in Fig.8.

Text CCs merging algorithm

```
1: According to rule (3), neighborhood CCs of each
CC are refreshed;
2: FOR i=0: NumofCCs
      Set CCj_Flag = TURE;
      FOR j=0: Num_neighborhood_CCi
         IF $CC_i$ and $CC_j$ satisfy Rule (13) THEN
             Merge $CC_i$ and $CC_j$;
             CCj_Flag = FALSE;
         ELSE $CC_j$ is included in $CC_i$
             CCj_Flag = FALSE;
         END IF;
       END FOR;
   END FOR;
3: Refresh CCs Set
   FOR i=0:NumofCCs
      IF (CCi_Flag)
         CCi push to New_CC_set;
      END IF
   END FOR
4: If there is no CCs to merge, go to step 5;
Otherwise, go to 1.
5. Output the New_CC_set
```

Fig.8 Overall procedure of CCs merging

4 Text Region Segmentation

Located character regions are required to be binarized before recognition. Generally, characters can be distinguished from background with color, thus color clustering is a common option for binarization. However, characters in natural scene images often suffer from uneven light, reflex and shadow, which likely to make stroke broken and bring about some isolate noise, so that the performance of color clustering is greatly affected. To deal with this problem, Thillou[14] proposed a text extraction method by using of the tactics of selective metric-based color clustering, and obtained better results. Inspired by this paper, we propose a two-step segment method. Firstly, color cluster is preformed in RGB space and HSI space respectively, and then select better result as the initial segmentation. Secondly, a graph cut based labeling procedure is used to improve the results of initial binariazation. The block diagram of segmentation method is showed as Fig.9.

4.1 Initial Segmentation

Generally, color clustering in RGB space can get satisfactory result, while in conditions of uneven light, diffuse and shadow, color clustering in HSI space always get better result. To obtain the optimal initial result, clustering algorithm is performed in RGB space and HSI space respectively, and K-means algorithm is employed to classify each pixel as text pixel or background, the clustering results are shown as Fig.10 (b) and (c).

The optimal result selected by log-Gabor filter which used as in [14]. The Initial result is shown in Fig.10 (d).

4.2 Graph cut based relabeling

To remove noise and connect broken stroke segments, graph cut algorithms is used to relabel the initial binarization result. Firstly, GMMs are employed to make model for foreground color and background color. Set $Y=\{y_1, y_2, \cdots, y_N\}$ is the set of image pixels in RGB space, $X=\{x_1,x_2,\cdots,x_N\}$ denotes the corresponding label set. Thus every color pixel follows the distribution as below.

$$p(y_i\mid x_i,\theta,k_i)=\frac{\pi(x_i,k_i)}{\sqrt{\det(\Sigma(x_i,k_i))}}\times$$
$$\exp(\frac{1}{2}(y_i-\mu(x_i,k_i))^T\Sigma^{-1}(y_i-\mu(x_i,k_i))) \qquad (13)$$

Where, $\boldsymbol{\theta}=\{\mu(\mathbf{x},\mathbf{k}),\Sigma(\mathbf{x},\mathbf{k}),\pi(\mathbf{x},\mathbf{k})\}$ is the parameter of the model, and μ,Σ,π are mean, covariance, and mixture weight coefficient respectively.

Refer to section 2.1, the energy function is constructed as following.

$$\begin{aligned}E(X)&=D(X)+\lambda B(X)\\&=\sum_i D(x_i)+\lambda\sum_{i,j}B(x_i,x_j)\\&=-\sum_i(\log(p(y_i\mid x_i,\theta,k_i))\\&+\log(\pi(x_i,k_i))+\lambda\sum_{i,j}B(x_i,x_j)\end{aligned} \qquad (14)$$

Where, smoothness term B is computed using Euclidean distance in RGB color space.

$$B(x_i,x_j)=\exp(-\beta\|y_i-y_j\|^2)[x_i\neq x_j] \qquad (15)$$

Based on above analysis, the initial binarization result is refined, the final result is as shown in Fig.10(e).

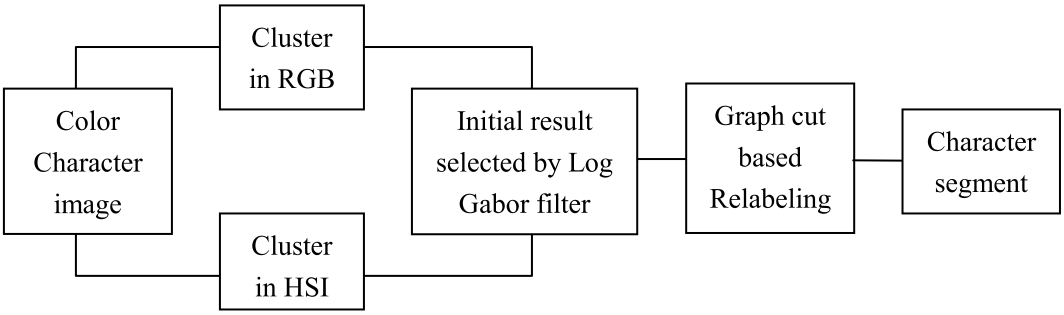

Color Character image
Cluster in RGB
Cluster in HSI
Initial result selected by Log Gabor filter
Graph cut based Relabeling
Character segment

Fig.9 Block diagram of proposed segmentation method

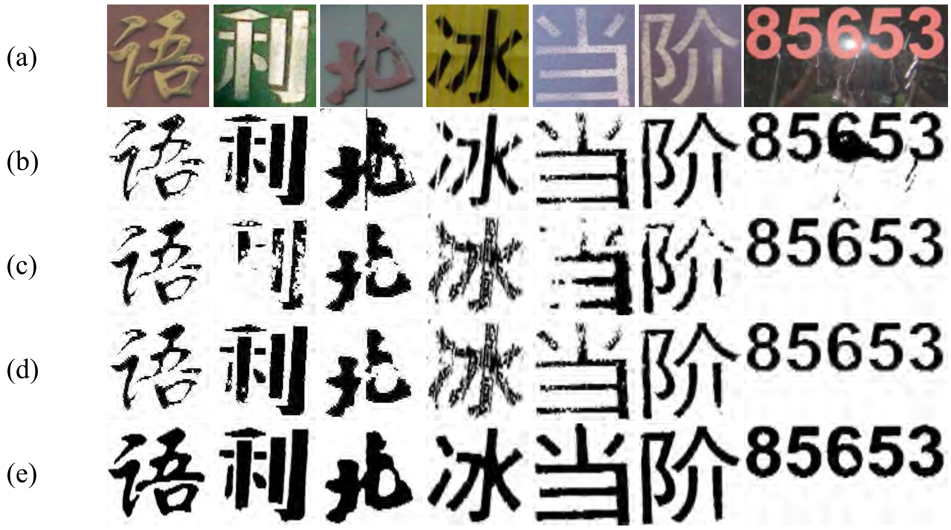


Fig.10 Examples of text binarization: (a) Original character images, (b) Binarization by color clustering in RGB space, (c)Binarization by color clustering in HSI space, (d) Initial binarization result,(e) Refined by Graph cut algorithm

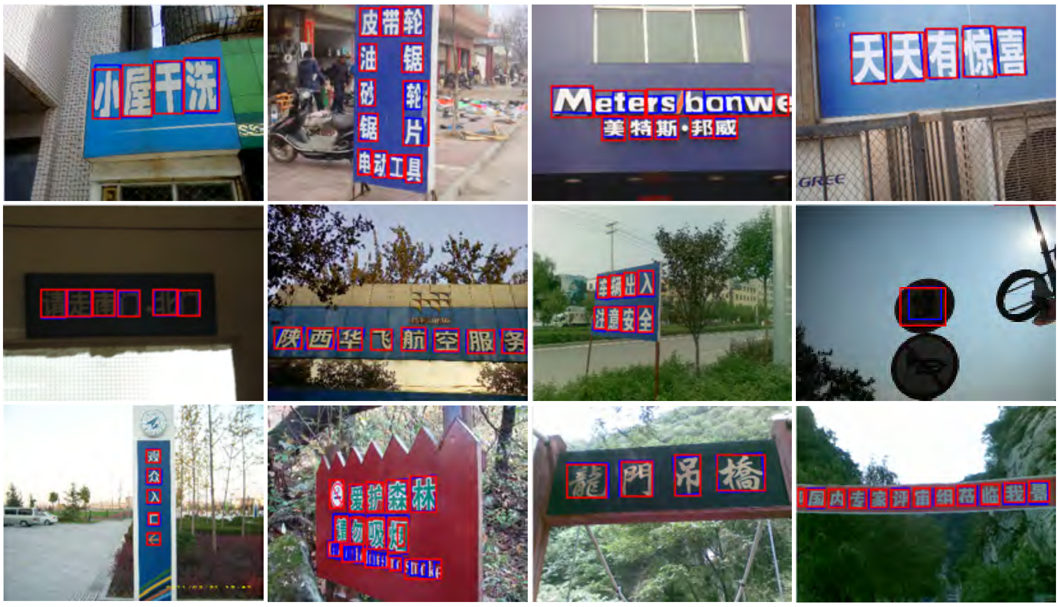


Fig.11 Text detection examples on our dataset

5 Experimental Results

In this section, we present the experimental results of text location and segmentation on our benchmark database. The dataset contains about 500 scene text images, text appearance varies with colors, orientations, font sizes, sunlight and complex background. Images are captured by smart-phone and digital camera, with an image resolution of 1600×1200. We use 230 images as training dataset, and about 270 images as test dataset.

5.1 Performance of the location method

We adopt the evaluation scheme of ICDAR2005 robust reading contest, which uses precision, recall and f measure to evaluate text location algorithm, which detailed in [20].

To evaluate the performance of graph cut based labeling process, we compare it with SVM based verification. In the experiment, 2317 text CCs and 5832 non-text CCs obtained from training dataset are used to train SVM classifier. The evaluation result is given in Table 1.

Table 1 Performance (%) Evaluation of location algorithm on our dataset

SVM			Graph Cut		
Precision	Recall	f	Precision	Recall	f
71.26	82.19	76.3	73.81	83.57	78.4

We can see from the Table 1, the performance of location based on graph cut outperforms that of the location method only using SVM. And some location examples on our dataset is showed in Fig.11, which indicates that the proposed method can detect characters under different conditions of various arrangement, complex background, bad illumination, and so on.

Our method is mainly designed for Chinese environment, so we compare our method with other methods by using a multilingual (include Chinese and English, show in Fig.12) dataset, which initially used by Pan[8] et al.. The training dataset contains 248 images and the testing dataset contains 239 images. Table 2 shows the comparison result.

Table 2 Performance (%) Comparison with other method on multilingual dataset

Methods	Precision	Recall	f	Time(s)
Pan' method[8]	65.9	64.5	65.2	3.11
Our method	63.2	67.8	65.4	0.58

As we can see, the comparison between our method and Pan' method[8] in Table 2 shows the performance of our method is comparable to that of Pan' method, but our method is less time-consuming than Pan' method.

5.2 Evaluation of the segmentation algorithm

To evaluate the segment performance of the proposed algorithm, the experiment is done with 300 scene text images, which include uneven light, clutter background. Obviously, the effectiveness of text segmentation can affect considerably the performance of OCR. Therefore, the performance of text segmentation is evaluated based on the results of OCR and pixel based precision, which is defined as following.

$$CPR = \frac{N_c}{N} \quad (16)$$

$$PPR = \frac{M_p}{M} \quad (17)$$

Where, CPR and PPR are recognition rate and segmentation precision rate respectively. N_c represents the number of characters recognized correctly, and N is number of all the characters in the test dataset. M_p is the number of pixels segmented correctly, and M is the number of pixels included in the segmentation image.

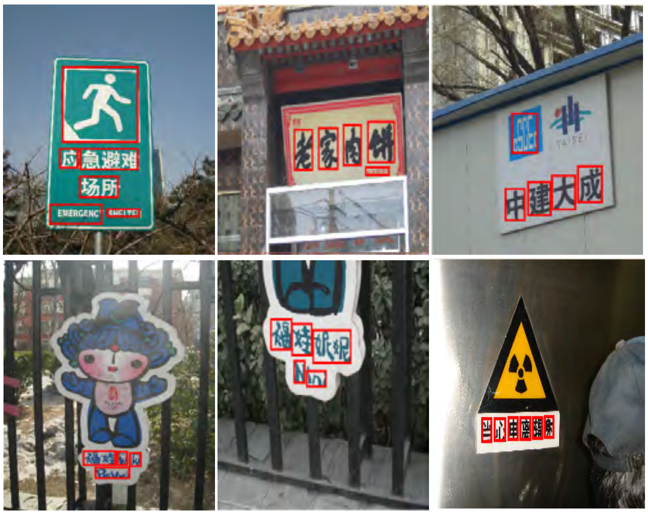


Fig.12 Text detection examples of our method on multilingual dataset

Fig.12 shows the result of our segment method on different images, which makes it clear that segmenting foreground text and background scenes is a difficult task. Nevertheless, our approach still achieves reasonably good results.

In the experiment, we employ Tesseract-OCR[21] for recognition, and compare the proposed segmentation approach with local threshold based method[12], cluster based method[13] and MRF based method[16]. The experiments are done with two different datasets which come from ICDAR2011 words dataset, one is simple data set (images with good quality), while the other is complex data set (images with poor quality which suffer from uneven light, low contrast and clutter background). Fig.13 and Fig.14 show the experimental results. For the dataset 1, our method is comparable to other methods, but for the second dataset, our method outperforms the other three methods obviously. That is to say that our proposed method gets better result, especially in poor images.

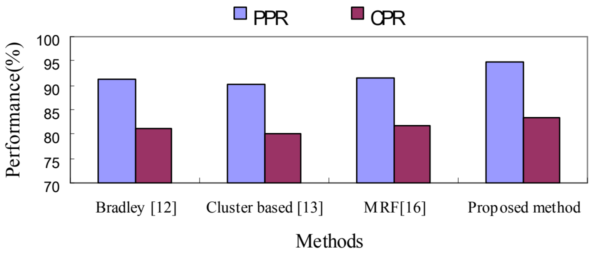


Fig.13 Result of comparison with other methods on dataset 1

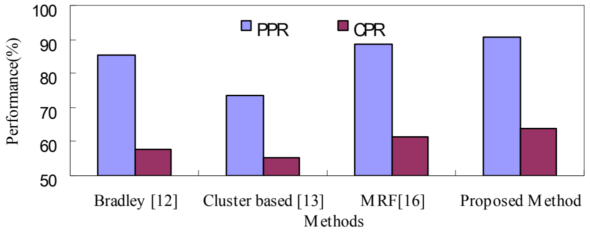


Fig.14 Result of comparison with other methods on dataset 2

6 Conclusion

In this paper, a text location and segmentation scheme for scene images is proposed. For text location, stroke edge is employed to detect text regions, and combine unary properties and neighborhood CCs relationship into a graph based framework, which improves the accuracy of location compared with classifiers with the region features of CCs. In addition to this, location output is single character set rather than text line, which greatly improves the efficiency of location algorithm, and avoid recognition errors brought about in the period of text line segmentation. For text segmentation, firstly, color clustering is performed in RGB and HSI color space, of which the better clustering result is selected as initial segmentation result, and then graph cut algorithm is employed to re-label pixels as text or non-text pixel. Furthermore, an open OCR package is applied to show the effectiveness of our method. The recognition results show the proposed segment method gets better performance comparing with other methods especially in poor quality.

Acknowledgement

This research is supported by Huawei University Innovation Research Program (IRP-2012-03-06), and partly funded by Key Science and Technology Program of Shaanxi Province (2012K06-16).

References:

[1] J Zhang, R Kasturi, Extraction of Text Objects in Video Documents: Recent Progress, DAS, 2008, pp.5-17.

[2] Nabin Sharma, Umapada Pal, Michael Blumenstein, Recent Advances in Video Based Document Processing: A Review, 2012 10th IAPR International Workshop on Document Analysis Systems, 2012, pp.63-68.

[3] Qixiang Ye, Qingming Huang, Wen Gao, Debin Zhao, Fast and robust text detection in images and video frames, Image and Vision Computing, Vol.23, No.6, 2005, 23:565–576.

[4] Palaiahnakote Shivakumara, Trung Quy Phan, Chew Lim Tan. New Fourier-Statistical Features in RGB Space for Video Text Detection. IEEE Trans. Circuits Syst. Video Techn, Vol.20, No.11,2010, pp.1520-1532.

[5] Chucai Yi, Yingli Tian .Text Detection in Natural Scene Images by Stroke Gabor Words, ICDAR2011, 2011, pp.177-181.

[6] Epshtein, B., Ofek, E., Wexler, Y., Detecting text in natural scenes with stroke width transform. In: IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2010, pp. 2963–2970.

[7] Chen, H, Tsai S, Schroth G., Chen D et al. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. Proceedings of Internal Conf. on Image Processing (ICIP), 2011, pp. 2609-2612.

[8] Yifeng Pan., Xinwen Hou., Chenglin Liu, A hybrid approach to detect and localize texts in natural scene images, IEEE Trans. on Image Processing, Vol.20, No.3, 2011, pp. 800-813.

[9] Gang Zhou, Yuehu Liu, Scene text detection based on probability map and hierarchical model, Optical Engineering, Vol.51, No.6, 2012, pp. 067204.

[10] Cunzhao Shi, Chunheng Wang, Baihua Xiao et al, Scene text detection using graph model built upon maximally stable extremal regions, Pattern Recognition Letters ,Vol.34, 2013, pp.107-116.

[11] Messaoud I B, Amiri H. New Binarization Approach Based on Text Block Extraction, Proceedings of 2011 International Conference on Document Analysis and Recognition. Washington DC: IEEE Computer Society, 2011, pp.1205-1209.

[12] Bradley D, Roth G, Adaptive Thresholding Using the Integral Image, ACM Journal of Graphics Tools. Vol.22, No.2, 2007, pp.13-21.

[13] Yan Song, Anan Liu, Lin Pang, et al, A Novel Image Text Extraction Method Based on K-means Clustering, Proceedings of the 8th International Conference on Computer and Information Science, Washington DC: IEEE Computer Society, 2008, pp. 185-190.

[14] Ce´line Mancas-Thillou, Bernard Gosselin, Color text extraction with selective metric-based clustering, Computer Vision and Image Understanding, vol.107, 2007, pp.97-107.

[15] Qixiang Ye, Wengao, Qingming Hang, Automatic text segmentation from complex background, Proceedings of 2004 International Conference on Image Processing. Washington DC: IEEE Computer Society, 2004, pp. 2905-2908.

[16] Mishra A, Alahari K, Jawahar C V, An MRF Model for Binarization of Natural Scene Text, Proceeding of the 11th International Conference on Document Analysis and Recognition. Washington DC: IEEE Computer Society, 2011, pp. 11-16.

[17] Xiaopei Liu, Zhaoyang Lu, Jing Li, natural scene text location oriented Chinese environment, Proceeding of SPIE 8009, 2011, pp.80092Y.

[18] Yuri Boykov, Olga Veksler, Ramin Zabih, Efficient Approximate Energy Minimization via Graph Cuts, IEEE transactions on PAMI, Vol. 20, No.12, 2001, pp.1222-1239.

[19] PLATT J C, Probabilities for SV machinestes, Proc of Advances in Large Margin Classifiers, Cambridge: MIT Press, 2000, pp.61- 74.

[20] S. M. Lucas, ICDAR 2005 Text Locating Competition Results, in Proceedings of the 8th International Conference on Document Analysis and Recognition, Seoul, S. Korea, 2005, pp. 80-84.

[21] http://code.google.com/p/tesseract-ocr/downloa ds/ list.