

A Novel Image Text Extraction Method Based on K-means Clustering

Yan Song^{1,2}, Anan Liu¹, Lin Pang^{1,2}, Shouxun Lin¹, Yongdong Zhang¹, Sheng Tang¹

¹Key Laboratory of Intelligent Information Processing,

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, 100080

²Graduate University of the Chinese Academy of Sciences, Beijing, China, 100080

{songyan,liuanan,panglin,sxlin,zhyd,ts}@ict.ac.cn

Abstract

Texts in web pages, images and videos contain important clues for information indexing and retrieval. Most existing text extraction methods depend on the language type and text appearance. In this paper, a novel and universal method of image text extraction is proposed. A coarse-to-fine text location method is implemented. Firstly, a multi-scale approach is adopted to locate texts with different font sizes. Secondly, projection profiles are used in location refinement step. Color-based k-means clustering is adopted in text segmentation. Compared to grayscale image which is used in most existing methods, color image is more suitable for segmentation based on clustering. It treats corner-points, edge-points and other points equally so that it solves the problem of handling multilingual text. It is demonstrated in experimental results that best performance is obtained when k is 3. Comparative experimental results on a large number of images show that our method is accurate and robust in various conditions.

1. Introduction

Nowadays, we are deluged by information delivered through all kinds of medium like Internet and television. How to organize and manage these multimedia data in order to make the indexing and query convenient has become an urgent issue. In recent years, many researches [1] have been done to solve the problem. In multi-model information, text in images is an important source because it contains tremendous high-level semantic sense compared to visual and audio information. For example, texts superimposed in news videos usually generalize the content of the news reports. Besides, optical character reader (OCR) software is mature enough which is more robust than automatic speech recognition (ASR) and visual analysis techniques.

Image text recognition method generally comprises following steps: text location, text segmentation and text recognition. Text location methods can be approximately divided into three kinds: connected component based [2], texture based [3] and edge based [4]. The first method locates text quickly but tends to fail when the background is complex. The problem of texture-based methods is large computational complexity in the texture classification stage and it may confuse when text-like regions appear. The third one usually has a problem in handling large size texts. In text segmentation, methods fall into two kinds. The first one is based on color which separates text from background by thresholding. The commonly adopted thresholding methods include Otsu's in [5], Niblack's in [6] and Bernsen's in [7]. It is noticed that method of thresholding is difficult to be adapted to all kinds of situations. The other one is based on stroke which employs some filters to pick the pixels on strokes [8]. But the pixels on the intersection of strokes are usually ignored.

Considering the existing problems, a novel and universal method of image text extraction is presented which obtains satisfying results. A coarse-to-fine process is used for text location. It consists of multi-scale text location and text region refinement. Multi-scale images are used to solve the problem of handling texts with different font sizes. The refinement step utilizes the horizontal and vertical projection profiles to reject falsely located text regions and to contract text regions. For text segmentation, grayscale image is usually used in traditional method. But color information is lost which is important to differentiating text from background. K-means clustering is adopted in our method and color image is more suitable for segmentation based on clustering. Each pixel is classified with color features without the constraint of its location. Thereby, the method avoids the negative influence of neighboring pixels. It is a universal method for text segmentation which makes the process

independent on language, text font sizes and text font styles.

The remainder of the paper is organized as follows. In section 2, we introduce text location. Then we specifically illustrate text segmentation in section 3. Experimental results are shown in section 4. At last, conclusion and future work are stated in section 5.

2. Text location

Text location is to locate all kinds of texts appearing in the image. Here, we divide the method to two steps: multi-scale text location and text region refinement. Location based on edge has a disadvantage that texts with different font sizes are difficult to deal with. So we adopt a multi-scale approach to locate text without missing as much as possible. In text region refinement, a method based on projection profiles is used to reject falsely located regions.

2.1. Multi-scale text location

Small-font-size texts (as characters with height smaller than 6 pixels) are likely to be dropped while large-font-size texts (as characters with height larger than 25 pixels) tend to be located incompletely in text location. A typical example is shown in figure 1. So a multi-scale approach has been adopted in our method. Firstly, we subsample the original image to half the width and half the height. Secondly, we upsample the original image to double width and double height by linear interpolation. Each color image is converted to a 256-level grayscale image. Sobel detector [9] is adopted in our method. Since it is noticed that stronger contrast is usually set in text regions to highlight text parts, a binarization method is implemented to the edge map. Thus the edge pixels in text regions are more likely to be kept while other edge pixels are removed from the edge pixel set. By this time, three binarized edge maps are used as the input of the location step, as explained next.

The location step is to localize coarsely the texts with large-font-size, normal-font-size and small-font-size in the three edge maps respectively. Firstly, the marking map is generated. Each edge map is scanned by a window of $n \times n$ pixels sliding with step of n pixels. Here, n is set 4. Each block in the marking map corresponds with a window in the edge map. Then, each window is divided into 4 parts: the top left, the top right, the bottom left and the bottom right. The numbers of edge pixels in these four parts are counted and defined as n_1, n_2, n_3 and n_4 . Afterwards, n is



Figure 1. Text location fails with different font sizes. (a)Small-font-size text missed; (b)Large-font-size text missed

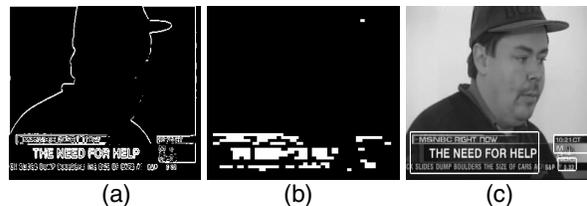


Figure 2. Stepwise results of text location. (a)Binarized edge map; (b)Marking map; (c)Located regions.

generated by multiplying the four numbers:

$$n = \prod_{i=1}^4 n_i \quad (1)$$

And each block in marking map is defined as:

$$p = \begin{cases} 1, & n > 0 \\ 0, & n = 0 \end{cases} \quad (2)$$

The marking map denotes how the edge pixels distribute in the edge map. Only those windows whose edge pixels distribute dispersedly correspond with “1” in the marking map. Thus edge pixels on the boundaries of large objects like human body are possibly removed in marking map. As the edge pixels of text distribute densely and with isotropy, the corresponding blocks in the marking map tend to be kept. After the marking map has been generated, connected component is analyzed to obtain the maximum enclosing rectangle of each component, which is called text-block.

These text-blocks are merged to obtain more potential text regions if they overlap with each other. Steps of text location are shown in figure 2. Then the subsampled and upsampled images are resized to the original size and the size of located region is transformed accordingly. The three results obtained from multi-scale edge maps are fused by the “or” operation. Thus text regions are located coarsely. Text location results of the images in figure 1 are shown in figure 3. It’s easy to notice that the small and large font size texts are located integrally.

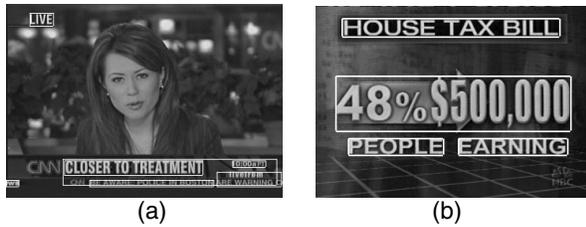


Figure 3. Located text. (a) Small-font-size texts are located; (b) Large-font-size texts are located.

2.2. Text region refinement

In text region refinement, three kinds of refinement are done:

1. A contraction is implemented around the text to locate more precisely.
2. Multi-line texts (if there are) are segmented to single lines to minimize background including.
3. Falsely located regions are rejected.

Background minimization reduces the disturbance of the non-text regions and helps the following steps work quickly. The text region refinement is implemented by computing the edge map projection profiles horizontally and vertically. As is explained by Michael R. Lyu in [10], an iterative method is designed to refine text regions. Horizontal and vertical projection profiles are used to locate the valley/peak points. And the process is done iteratively in case there are multi-line texts. An example of vertical and horizontal projection files are shown in figure 4. The method is used to locate little red circles (called cutting points here).

Obviously, the first two kinds of refinement can be done by the approach mentioned above. The last one can be achieved by throwing out the block if no cutting point is found.

3. Text segmentation

To recognize the texts in the images, binarized images are input to the commercial OCR software. The segmentation (also called binarization) results influence the final recognition results directly. Text segmentation is to label the text pixels and non-text pixels. Although traditional method is to find optimal global or local threshold to binarize the gray image, thresholding has a problem with handling complex background.

Clustering is an effective and popular method in image segmentation. Here, color image is used rather than gray image because color images have three channels and each pixel is described by three dimensions. Our method of clustering based on color

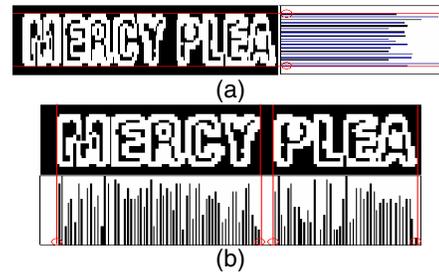


Figure 4. Text region refinement. (a) Vertical refinement; (b) Horizontal refinement.

depends on the assumption that text pixels are in consistent color. As long as the difference is not very large, text pixels tend to be clustered together because the difference between text and non-text pixels is usually larger. Exceptions exist in practical but it is very rare that there is great color inconsistency in text.

K-means is adopted in our method because of its simplicity and efficiency. The k-means algorithm partitions a set of elements into k clusters so that the intracluster similarity is high but the intercluster similarity is low. In text segmentation, RGB values of pixels are input to k-means clustering algorithm and the output is the cluster label for each pixel. Here we get two important issues:

1. How to decide K, the number of clusters?
2. Given that k clusters have been classified, how to decide which cluster or clusters to be text pixels or background pixels?

Researches about k-means clustering for grayscale text segmentation have been done in [11]. They take k as 2, 3 and 4 and choose the best OCR result. Intuitively, 2 is a good choice for the first issue. But it maybe more complicated in practical considering the contours around the text and background complexity. Thus we have done several experiments to find out the most appropriate value for k. In this paper, we only consider situations that when k is less than 5. This is because that if more than 4 clusters have been generated, it is complicated to decide which clusters are text clusters and which are not.

Theoretically, text can be of any color, that is to say, can be any cluster in the k-means clustering result. But in practical, to highlight the texts, they are usually set to be the lightest or the darkest. Consequently, the cluster with the highest grayscale or the lowest grayscale is most probably the text cluster. Many works called color polarity classification have been done to differentiate these two cases, see [12] [13]. In this paper, we only consider normal text (text is brighter and background is darker) for simplicity. So the cluster with high grayscale center is considered text in our experiment and it is proved feasible. So when k is 2, the cluster with higher grayscale cluster-center is

labeled as text and the other one is non-text. When k is 3, the clusters with highest grayscale cluster-center or the first two highest grayscale cluster-centers are considered text. It is deduced by analogy when k is 4.

Experimental results show that when k is 3 and the highest grayscale cluster is considered text, the best performance is obtained. An example is shown in figure 5.

4. Experimental results

Our method is tested on a set of video frames. The videos come from the data set of High Level Feature Extraction in TRECVID 2005 and 2006. It contains television programs from CCTV, NBC, CNN and NDTV. 360 frames containing texts in Chinese and English are extracted to make the test set.

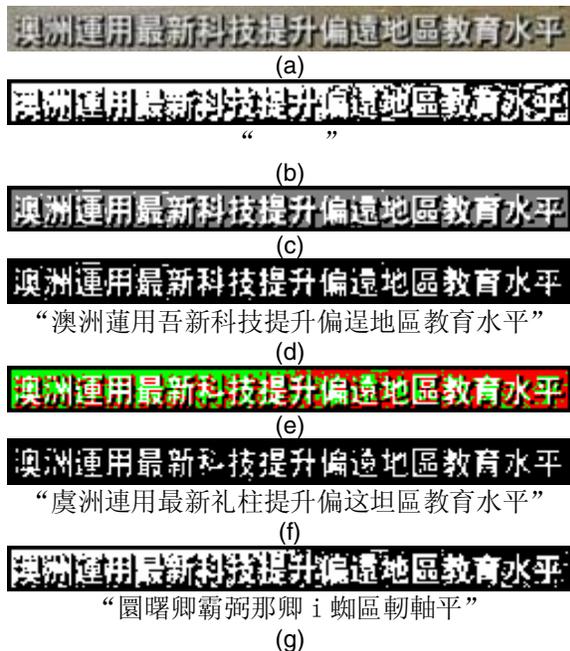


Figure 5. Clustering and recognition results by k-means clustering. (a)The original image; (b)Clustering and recognition results by k-means clustering when k is 2; (c)Clustering result by k-means when k is 3;(d)Binarization and recognition results by k-means clustering when k is 3;(e)Clustering result by k-means when k is 4;(f)Binarization and recognition results by k-means clustering when k is 4 and the highest grayscale cluster is labeled as text;(g) Binarization and recognition results by k-means clustering when k is 4 and first two highest grayscale cluster are labeled as text.

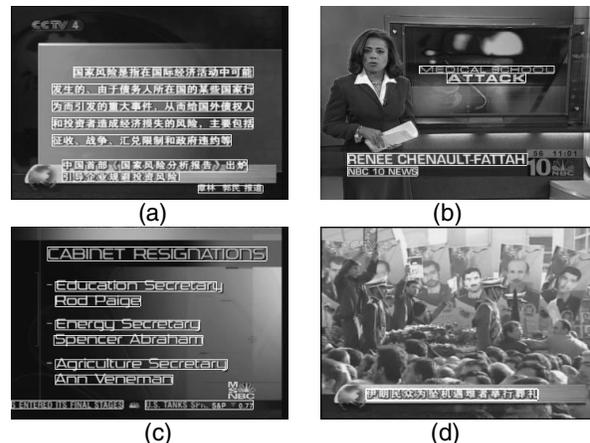


Figure 6. Examples of experimental results of text location.

4.1. Text location

In this section, method for location is evaluated on the test set. In figure 6, some examples of experimental results have been illustrated. It is shown that the algorithm is independent on language and it works well with different font sizes and font styles.

The idea of evaluation for text location in [14] is adopted. It is considered as a correctly located text block if the overlap area of the located text block (LB) and the ground-truth text block (GB) is more than 75% of the LB and more than 90% of the GB. Then two measurements of the text location algorithm are defined:

- 1) Recall: indicated by the rate of correctly located text blocks and the ground truth.
- 2) False alarm rate: evaluates by the rate of the falsely located and all located text blocks.

To evaluate the two steps in text location, result of each step is listed in table 1. As our purpose of the first step is to locate texts as much as possible, little attention is paid to false alarm rate. So it is easy to understand that the recall of the first step is high and the false alarm rate is also high. This high false alarm rate can be reduced in the next step. In table 2, we illustrate the influence of the multi-scale text location. The aim of multi-scale location is to makes sure that the texts with large-font-size are located wholly and the texts with small-font-size are not dropped. So the non-multi-scale location is implemented with the original image input to the location step while multi-scale location is with three images. In the experiment, we find that texts higher than 25 pixels tend to be located incompletely without multi-scale location algorithm. Similarly, texts with height smaller than 8 pixels tend to be missed and they are usually found in multi-scale location algorithm.

Table 1 Results of two steps in text location

	Recall	False alarm rate
Multi-scale location	98.38%	21.59%
Text region refinement	91.65%	1.94%

Table 2 Results of multi-scale location and non-multi-scale location

	Recall	False alarm rate
Non-multi-scale location	90.16%	1.97%
Multi-scale location	91.65%	1.94%

4.2. Text segmentation and recognition

Firstly, to evaluate the algorithm of text segmentation more accurately, the text part is cut out manually from the frames to make the test set. There are 1343 Chinese characters in 104 images and 2082 English letters in 121 images. Then the binary images are input to the commercial software HWOCRSDK1.2. Segmentation performance is evaluated by the resulting OCR recognition rate (RR) and precision rate (PR).

Recognition rate is defined as:

$$RR = \frac{\text{num}(\text{correctly recognized character})}{\text{total num}(\text{true character})}$$

Precision rate is defined as:

$$PR = \frac{\text{num}(\text{correctly recognized character})}{\text{total num}(\text{recognized character})}$$

K-means algorithm based on RGB color space is run when k is 2, 3 and 4. The experimental results of Chinese texts are illustrated in table 3. When k is 3, the cluster with the highest grayscale center is considered as text and the others are non-text. When k is 4, two kinds of situations are considered. The cluster with the highest grayscale center and the clusters with the first two highest grayscale centers are picked. $K=4^1$ denotes first situation and $K=4^2$ denotes the second. When k is 3 we get the highest RR and PR. And table 4 shows the English results. Similar conclusion is obtained. When k is 3 we get the highest RR and PR. It is noticed that when k is 3 both Chinese and English texts are segmented and recognized well and other methods tend to fail in Chinese text segmentation with RR below 80%. This is because Chinese texts tend to be in more complicated background in our test set and the algorithms when k is 2 and k is 4 are more sensitive to it. And also it is noticed that Chinese OCR is more sensitive to the segmentation results than English. Otsu's method used in [5] and Niblack's method used

in [6] are compared to k-means clustering. An example of results is laid out in figure 7. Here, k is 3. And the highest grayscale cluster is considered as the text cluster while the others are non-text. It is obviously noticed that Otsu's method is incapable of segmentation when contrast is weak. A parameter should be decided in Niblack's method. A fixed value is not suitable in this situation that some parts of the texts are missed. The k-means clustering based on color is robust in complicated background situation and no parameter setting is needed.

The statistical results are shown in table 5 and table 6. It is easily noticed that our method obtains the best results. And similarly it works well with both Chinese and English.

4.3. The overall method experimental result

To evaluate the whole algorithm, 50 images in Chinese and 50 images in English from the test set are chosen to be tested with method including text location, text segmentation and recognition. The RR and PR for Chinese is 75.63% and 77.05% and the RR and PR for English is 86.24% and 88.17% respectively. Because the error introduced in text location influences text segmentation, the ultimate result is a bit worse than that is shown in 4.2

5. Conclusion and future work

This paper proposes a novel method of image text extraction. The method consists of two steps: localization and segmentation. Localization is a coarse-to-fine process containing multi-scale text location and text region refinement. The first step is to locate text with different font sizes. And in the second step, falsely located text regions are rejected and texts are contracted to make location more precise. K-means

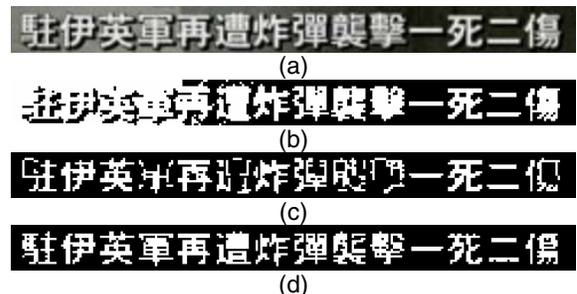


Figure 7. Binary results of different methods. (a) Original image; (b) Binary result of method used in [6]; (c) Binary result of method used in [7]; (d) Binary result of k-means clustering when k is 3.

clustering based on color are used in text segmentation. And it is demonstrated in the experimental results that when k is 3 the best performance is obtained. It is illustrated that our method is accurate and robust in different situations.

Future work includes researching more deeply in the issue about k-means clustering in text segmentation and utilizing temporal information in video text extraction.

6. Acknowledge

The work was supported by National Basic Research Program of China (973 Program, 2007CB311100), the National High Technology and Research Development Program of China (863 Program, 2007AA01Z416), Beijing New Star Project on Science & Technology (2007B071).

Table 3 Chinese recognition results with k=2, 3, 4

	RR	PR
K=2	66.57%	69.12%
K=3	83.25%	84.19%
K=4 ¹	72.27%	70.44%
K=4 ²	56.74%	60.0%

Table 4 English recognition results with k=2, 3, 4

	RR	PR
K=2	94.11%	95.93%
K=3	94.60%	94.70%
K=4 ¹	89.09%	90.77%
K=4 ²	90.01%	93.92%

Table 5 Chinese recognition results of our method and other methods

	RR	PR
K means with k=3	83.25%	84.19%
Method in [5]	51.27%	59.37%
Method in [6]	62.40%	67.80%

Table 6 English recognition results of our method and other methods

	RR	PR
K means with k=3	94.60%	94.70%
Method in [5]	94.33%	95.51%
Method in [6]	80.33%	93.29%

7. Reference

- [1] Y. A. Aslandogan, C. T. Yu, "Techniques and systems for image and video retrieval," in IEEE Trans. Knowledge Data Eng., vol. 11, 1999, pp. 56-63.
- [2] A. K. Jain, B. Yu, "Automatic text location in images and video frames," in Pattern Recognition, vol. 31, 1998, pp. 2055-2076.
- [3] K. I. Kim, K. Jung, H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," in IEEE Trans. on PAMI, vol. 25, 2003, pp. 1631-1639.
- [4] V. Wu, R. Manmatha, E.M. Riseman, "Textfinder: an automatic system to detect and recognize text in images," in IEEE Trans. on PAMI, vol. 20, 1999, pp. 1224-1229.
- [5] C. Zhu, Y.X. Ouyang, L. Gao, Z.Y.Chen, Z.Xiong, "An Automatic Video Text Detection, Localization and Extraction Approach," in Proc. of Inter. Conf. on Signal-Image Techno. & Internet-based Syst. , 2006.
- [6] J. Xi, X.S. Hua, X. R. Chen, L. W. Y., H. J. Zhang. "A Video Text Detection and Recognition System," in Proc. IEEE Int. Conf. Multimedia Expo, 2001, pp. 873-876.
- [7] Bersen J. "Dynamic thresholding of gray-level images," Proc. of 8th Intel Conf. on Patt. Recon [C], 1986, pp.1251-1255.
- [8] T. Sato, T. Kanade, E.Hughes, M. Smith, "Video OCR for Digital News Archives," in IEEE Workshop on Content-Based Access of Image and Video Database, Bombay, India, 1998, pp.52-60.
- [9] I. Sobel, "An isotropic 3*3 image gradient operator," in Machine Vision for Three-Dimensional Scenes, H. Freeman, Ed. N.Y.: Academic, 1990, pp.376-379.
- [10] M. R. Lyu, J. Q. Song, M. Cai, "A Comprehensive Method for Multilingual Video Text Detection, Localization, and Extraction," in IEEE Trans. on Circuits and Syst. For Video Technol. vol.15, no.2, 2005, pp.243-255.
- [11] D. T. Chen, J. M. Odobez, H. Boulard, "Text detection and recognition in images and videos frames," in Pattern Recognition, vol.37, 2004, pp. 595-608.
- [12] A. Wernicke, R. Lienhart, "On the segmentation of text in videos," in Proc. IEEE Int. Conf. Multimedia Expo, vol. 3, 2000, pp. 1511-1514.
- [13] S. Antani, D. Crandall, and R. Kasturi, "Robust extraction of text in video," in Proc. 15th Int. Conf. Pattern Recognition, vol. 1, 2000, pp. 831-834.
- [14] Q. X. Ye, Q. H. Huang, W. Gao, D. B. Zhao, "Fast and robust text detection in images and video frames," in Image and Vision Computing, vol. 23, 2005, pp. 565-57.