1-1-2014

Text extraction in natural scenes using region-based method

Zhihu Huang

Jinsong Leng
Edith Cowan University, j.leng@ecu.edu.au

Text Extraction in Natural Scenes using Region-based Method

Zhihu Huang[1, 2], Jinsong Leng[3]
[1]College of Computer Science
Chongqing University, Chongqing, China
[2]Distance Education Center
Chongqing Radio & TV University
Chongqing, China
[3]School of Computer and Security Science
Edith Cowan University, WA, Australia
hzh@cqtbi.edu.cn, j.leng@ecu.edu.au

Journal of Digital Information Management

ABSTRACT: *Text in images is a very important clue for image indexing and retrieving. Unfortunately, it is a challenging work to accurately and robustly extract text from a complex background image. In this paper, a novel region-based text extraction method is proposed. In doing so, the candidate text regions are detected by 8-connected objects detection algorithm based on the edge image. Then the non-text regions are filtered out using shape, texture and stroke width rules. Finally, the remaining regions are grouped into text lines. Since stroke width is the intrinsic and particular characteristics of the text, the accuracy of the non-text filter are notably promoted. The improved Stroke Width Transform in the paper is less computing complexities and more accurate. Experimental results on sample ICDAR competition Dataset and our dataset show that the proposed method has the best performance compared with other five methods.*

Subject Categories and Descriptors
I.2.10 [Vision and Scene Understanding]: Image Analysis; **I.4.10 [Image Representation]**

General Terms: Image Processing, Content Processing

Keywords: Text Extraction, Text Localization, Document Image Analysis, Image Processing

Received: 4 April 2014, Revised 30 May 2014, Accepted 20 June 2014

1. Introduction

Text extraction from images or videos has attracted many intentions in last few decades [28, 31]. The text extracting from images and videos can be used in the fields of automatic retrieve, indexing, summarization, and searching of videos [1-4]. For example, the Informedia project from Carnegie Mellon University, text embedded in video library of newscasts and documentaries is one important source of information to provide full-content search and discovery [5].

Text in images generally provides useful high-level semantic information, which can help a computer to understand the content of the image [32]. In general, text in images can be categorized into two groups: scene text and artificial text [6]. Scene text is part of the image, which exists in scene when the image is taken, such as advertising board, poster, traffic signs. Whereas artificial text is laid over the image in a later stage, such as caption in news program, the name of somebody during an interview.

Most artificial text is caption of videos or explains of images. Artificial text is often used to describe the content of the image. On the contrary, scene text is changing in orientation, size, style, color, alignment, contrast of text. Therefore, the extraction of scene text is more difficult.

The automatic extraction from natural scene image is an extremely difficult task. The major challenges are outlined as follows: The first challenge lies in the variety of text: orientation, size, style, color, alignment and position. Secondly, text may be blurred from motion or occluded by other objects [7]. Since text exists in three-dimensional space, text in scene images may be distorted by slant,

tilt, and shape of objects on which they are found [8]. Thirdly, the text has various orientations: horizontal, vertical, slanted, and even mixed orientations within the same text area, such as text on a T-shirt or wrinkled sign [9].

A number of literatures deal with the text extraction from scene images. The methods for text extraction can be typically categorized in two groups: Region-based methods and texture-based methods [10].

Region-based methods classify the text and background by analyzing the geometrical arrangement of edges [11-14] or homogeneous color [15] and grayscale components [16] that belong to characters. The method works in a bottom-up fashion: a set of sub-regions are detected from an image, then merge these sub-regions into successively larger ones until all regions are identified, and finally the non-text regions are excluded from the resulting regions by geometrical or texture properties. The region-based method is simple to implement and efficient, especially in the cases of polychrome text strings and low-resolution and noisy images, which are widely used [17]. However, it runs into difficulties when the image is degraded, multicolored, textured or blurred, which often occurs in camera-based image.

Texture-based methods scan images at a number of scales to get the textural properties that distinguish them from the background [18-20]. The textural properties mainly include high density of edges, low gradients above and below text, high variance of gray scale, distribution of wavelet, fast Fourier transform (FFT) or Discrete cosine transform (DCT) coefficient, etc. Gabor filters, Wavelet [21-22], Neural Network [23], FFT [24], DCT, and spatial variance are used to detect the textural properties of a text region in an image. The drawback of the method is the high computational complexity in the textural detection stage because textural-based filtering methods require an exhaustive scan of the input image to localize the regions.

Text extraction involves four major steps: text detection, text localization, text extraction, and text recognition [10]. Text detection is to determine the presence of text in an image or video. Text localization is to localize text in the image and to generate bounding boxes around the text. Text extraction is to segment text from the background to facilitate its recognition. Text recognition is to transform extracted text into plain text using optical character recognition (OCR) technology. The first two steps are important to achieve high-quality text recognition results when applying an OCR system.

Although many methods have been proposed for this task in the last decade [11-24], there is still room for improvement. Firstly, the accuracy is not high. The result from two world competitions (Text Location Competition at ICDAR 2003 and 2005 [29]) shows recall = 67% and precision = 62%). Secondly, many methods are used to a single language. Few of them address the multilingual problem.

In this work, we propose an improved region-based approach for automatic detection of text from natural scenes. The approach includes four processing stages: (i) pre-processing, (ii) generating candidate text regions, (iii) filtering out non-text regions, and (iv) grouping the remaining text regions.

The contributions of our method are detailed as follows: Firstly, the combination of edge detection and 8-connected objects detection increases the accuracy of candidate text detection. Secondly, shape, texture and SWT rules detailed in Section III are proposed to efficiently and robustly filter out the non-text regions from candidate text regions. Thirdly, the proposed method is a truly multilingual text detection algorithm. Finally, experiments have demonstrated that the proposed approach is very efficient and accurate.

The rest of the paper is organized as follows: Section 2 describes the algorithm to compute the stroke width of text. In Section 3, we illustrate the text extraction approach. The experimental results are analyzed and compared in Section 4. Section 5 concludes this paper.

2. The Stroke Width TRANSFORM

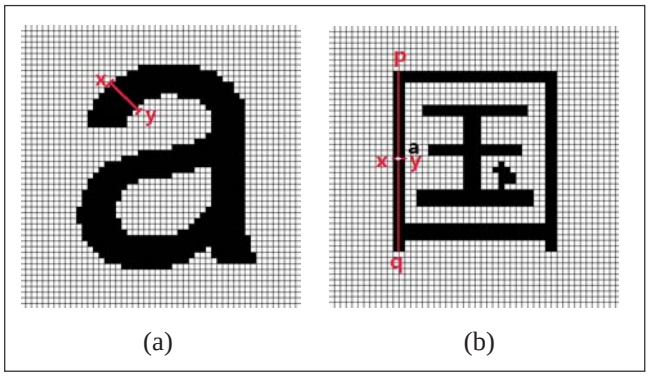


Figure 1. Computation of SWT

Strokes are the intrinsic and particular feature to distinguish texts from backgrounds in images. Many non-text regions which cannot be exclude by shape and texture rules may be filter out by rules based strokes. The Stroke Width Transform (SWT) is a good way to get stroke features. SWT is to compute the stroke width of each pixel which belongs to the stroke of text [25-26]. The output of the SWT is an image whose size equals to input image. The value of pixel belonged to stroke is the value of stroke width, the value of remaining pixel is set to 0. For example, the pixels belonged to letter a (shown in Figure 1a) are set to the values of stroke width, others are set to 0.

2.1 Original Stroke Width Transform (SWT)

The original algorithm of SWT is detailed in the following [26]:

Step 1: The initial value of each pixel of SWT image is set to ∞.

Step 2: Compute the edges of the input image using Canny edge detector, and get edge image Ie [27];

Step 3: Scanning edge image Ie, if the pixel x belong to edge (Figure 1 (a)), the gradient direction dx must be perpendicular to the orientation of the stroke;

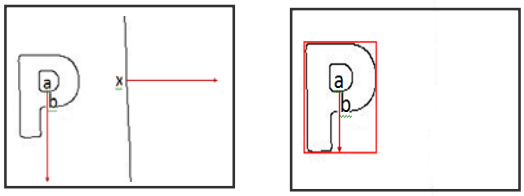
Step 4: Start from the pixel x, follow the ray $r = x + n * dx$, $n > 0$ until another edge pixel y is found or until to the boundary of the edge image le;

Step 5: If found the pixel y and the gradient direction of dy at the pixel y is roughly opposite to dx ($dy \in (-dx - \pi/6, -dx + \pi/6)$), the all pixels of the straight line which connects pixel x and pixel y are set the stroke width $\| x - y \|$ unless it already has a lower value (Figure1(b));

Step 6: If the corresponding pixel y is not found, or if dy is not opposite to dx, the ray is discarded.

Step 7: Go to Step 3 until all of edge pixels are scanned. As shown in Figure 1b, when the pixel has two or more value of stroke width, i.e. $\| x - y \|$ or $\| p - q \|$, the value of SWT is set to the smallest one $\| x - y \|$.

2.2 The improvement of SWT



(a) (b)

Figure 2. Computation of SWT. (a) original SWT (Red arrows are the direction to find point pair). (b) improved SWT (Red rectangle are remaining candidate text region)

The original SWT has some distinct drawbacks. As shown in Figure 2a, the edges of letter p is incomplete, it has a gap at edge point b. Starting from point a, following the red arrow, the corresponding edge point b cannot be found due to b is not belong to edges. Therefore, the values of stroke width of pixels from a to b will be set to 0. In other words, the pixels from a to b are not belong to strokes, which is a distinct error. Moreover, the scan should be finished when the ray comes to edge point b, but the scan will continue until to the bottom boundary of the image. Such approach cannot found correct point pair "a *and* b" but also increases the scanning time. The line x lies on right part of Figure 2a is not belong to edge of a text, but SWT is still conducted on it. This is another drawback of the SWT.

To overcome the drawbacks of original SWT, the following improvements are proposed, detailed as follows:

Firstly, we increase two steps before step 3 of SWT in order to reduce edge pixels scanned by SWT. The candidate text regions are detected in edge image Ie by 8-connected objects detection [28]. And then shape rules and texture rules are conducted to filter out the non-text regions from candidate text regions. The Figure 2b shows the line x in Figure 2a are filtered out as non-text region. Therefore, the SWT operation on line x are also deleted. That is to say, the computing time of SWT is distinctly shorted.

Secondly, SWT is only conducted to candidate text region but not to the whole image. There is only one candidate text region in Figure 2b, therefore, SWT are only conducted to the candidate text region which are marked with a red rectangle. The computing time of SWT is reduced due to the reduction of SWT range.

Thirdly, the detection of edge point pair is modified. When the edge point pair is not found following the ray, the detected range is expanded to the 2 pixels next to the ray. Thus, the SWT can be correctly computed when the gap of edge is not exceeding 4 pixels.

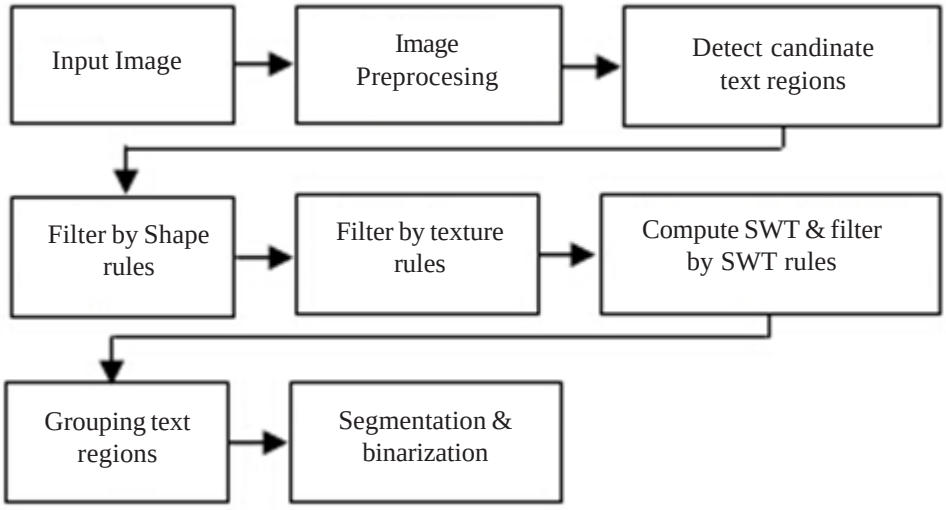


Figure 3. The Flowchart of the Algorithm

The improved SWT not only notably reduce the computing time but also increase the robustness

3. The Proposed Approach

In this section, our approach is illustrated, which can be divided into several elementary tasks. The flowchart of the algorithm is shown on Figure 3.

3.1 Image Preprocessing

The image preprocessing includes: the conversion from color image to gray image, the image filter and the image edge detection. The destination of image filter is to reduce image noise in order to improve the precision of edge detection. The median filter is employed in our method. Since the edge detection methods of Sobel, Prewitt, Laplacian of Gaussian and Zero-cross usually generate "*double age*", which is harmful to text localization. The canny method is conducted to detect edges of image.

In general, the contrast between the text and background in scene image is strong in order to easily read for people. In other words, there are strong edges between texts and background. Therefore, edge-based localization of text is currently major method to detect candidate text regions. The candidate text regions are detected by 8-connected objects detection algorithm. To increate the accuracy of candidate text region, the gap of canny edge are patched by conducting two morphological close operations. The structuring elements are respectively horizontal and vertical 1×5 ones shown in Figure 4.

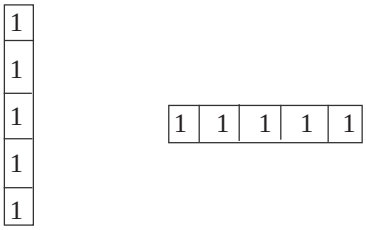


Figure 4. Structuring elements. (a) vertical structuring elements, (b) horizontal structuring elements

The original image and resulting image are respectively shown in Figure 5a and Figrue 5b. The candidate text regions are marked by white rectangle (called bounding box). All texts in the image are marked out, but many non-text regions are also mistakenly marked. In the subsequent steps, the non-text regions will be filtered out step by step.

3.2 Filter By Shape Rules

Shape rules mainly include the length, width and size of bounding box. The parameters of each rule are learned from the training set of ICDAR competition Dataset. The shape rules are detailed as follows.

1). The ratio of between width and height of bounding box. The ratio of text region should be limited a value between

(a)

(b)

Figure 5. The detection of candidate text regions. (a) original image, (b) the marked candidate text regions

0.1 and 10. The long and narrow components like telegraph pole, tree trunk etc. will be excluded by the ratio.

2). The ratio of between the size of bounding box and the number of text pixels within the candidate text region. The candidate text regions are filtered out when their ratio greater than 10. The regions only contain an insular or a few pixels are excluded.

3). The regions whose size is too small or too large are filtered out. It is difficult to read when the size of character less than 5×5 pixels, moreover the accuracy of OCR is also notably decrease. In general, the candidate text regions are filtered out when their size less than 5×5 pixels or greater than 70% of the image size.

The resulting image is shown in Figure 6a, a part of non-text regions are filtered out compared with Figure 5b.

3.3 Filter by texture rules

Text region in images has the particular texture features. For the true text regions, the background and foreground

(a)

(b)

(c)

Figure 6. Non-text region filter. (a) The outcome of shape rules. (b) The outcome of texture rules. (c) The outcome of SWT rules

has bigger contrast than non-text regions. In this section, statistical approaches are employed in texture rules. They are detailed in the following.

Let z be a random variable denoting intensity and let $p(zi)$, $i = 0,1,2,\ldots,L-1$, be the corresponding histogram, where L is the number of distinct intensity levels. The n^{th} moment of z is defined by equation (1).

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \quad (1)$$

The contrast, variance (smoothness), uniformity features are defined by the following equations.

1). Contrast δ.

$$\delta = \sqrt{\mu_n(z)} = \sqrt{\sum_{i=0}^{L-1} (z_i - m)^2 p(z_i)} \quad (2)$$

2). Variance (smoothness) R.

$$R = 1 - \frac{1}{1 + \mu_2(z) / (L-1)^2} \quad (3)$$

R is the relative smoothness. R is 0 for areas of constant intensity. The more change of intensity, the bigger until to 1.

3). Uniformity U

$$U(z) = \sum_{i=0}^{L-1} p^2(z_i) \quad (4)$$

When all intensities of pixels are equal, U comes to its maximum.

For the true text region, the contrast and variance are bigger, but the uniformity is smaller than the values of non-text regions like foliage, forest etc. For the training set of ICDAR competition Dataset, the parameters of the contrast, variance, uniformity is set to mean value of all candidate regions. In other words, the regions are filtered out if their value of contrast and variance less than mean value or their value of uniformity greater than mean value. The outcome of texture rules are shown in Figure 6b. The most non-text regions like foliage are excluded by texture rules.

3.4 Compute SWT & Filter By SWT Rules

The remaining non-text regions are very similar to the true text regions. Obviously, these non-text regions can be easily discriminated by character strokes. In this section, the improved SWT are conducted to the output of the last step. Then, the following rules of SWT are applied to filter out the non-text regions.

Mean value of SWT. The mean value SWTme of each candidate text region and the mean value SWTma of all candidate text regions are respectively computed. Then the deviation D is calculated by the equation (5). The candidate text regions are filtered out if their D greater than 0.5.

$$D = \frac{SWT_{me}}{|SWT_{me} - SWT_{ma}|} \quad (5)$$

The remaining candidate text regions are considered the true text regions, which are shown in Figure 6c.

3.5 Grouping Text Regions Into Text Lines

The text regions of the output from last step are not the text lines, but letters or words. Therefore, these text regions should be grouped into text lines. Text on a line is expected to have similarities, such as similar horizontal ordinate, similar stroke width, similar letter width and height, similar color, similar spaces between the letters and words etc. The text regions which simultaneously satisfied the following rules are grouped into a text line.

1). Computing the value x of horizontal ordinate of central coordinates of the bounding box for all text regions. The text regions will consider as a text line if the deviation of their value x is not greater than 4 times.

2). The deviation of between mean value SWTme of each region and their median stroke width does not exceed 2.

3). The distance of letters must not exceed 3 times the width of the wider one.

The resulting image is shown in Figure 7.

Figure 7. Grouping text regions into text lines

3.6 Segmentation And Binarization

Segmentation is to extract text based on bounding box. Then the binarization is conducted using Otsu algorithm [30] for each segmented text line. The binarized images can be input into an OCR system to transform text images into plain text.

4. Experimental Results

At present, there is no established database for text extraction. In many literatures, ICDAR competition Dataset is used to verify the performance of algorithm. ICDAR dataset are used to two international competitions. And the results of competitions are opened. The ICDAR dataset has 258 images in training set, and 251 images in test

set. The images are color, and the size of image varies from 307 × 93 to 1280 × 960 pixels. All image texts are English.

For a quantitative evaluation, we adopt the evaluation measurements: precision, recall, f-measure and time [29], which is detailed as follows.

The output of each algorithm is a set of rectangles called bounding boxes which are the smallest rectangles around the text regions. The ground truth boxes which are provided by dataset called target set T. The box set returned by algorithms is called estimate set E. The number of estimates which are correct, we denote c.

Precision p is defined as the number of correct estimates divided by the total number of estimates:

$$p = \frac{c}{|E|} \tag{6}$$

Recall r is defined as the number of correct estimates divided by the total number of targets.

$$c = \frac{c}{|T|} \tag{7}$$

The match mp between estimate and target box as the area of intersection divided by the area of the minimum bounding box containing both rectangles. The value is 1 for identical rectangles and 0 for no intersection between rectangles. For each rectangle in the set of estimates, the closest match is found in the set of targets, and vice versa. Therefore, the best match $m(r, R)$ for a rectangle r in a set of Rectangles R is defined as follows:

$$m(r, R) = max\{m_p(r, r') \mid r' \in R\} \tag{8}$$

Then, the last precision r' n and recall r' are defined as follows:

$$p' = \frac{\sum_{r_e \in E} m(r_e, T)}{|E|} \tag{9}$$

$$r' = \frac{\sum_{r_t \in T} m(r_t, E)}{|T|} \tag{10}$$

The f-measure is used to combine the precision and recall figures into a single measure of quality. The parameter α are used to control the relative weights. In our work, $\alpha = 0.5$ denotes equal weight for precision and recall.

$$f = \frac{1}{\frac{\alpha}{p'} + \frac{1-\alpha}{r'}} \tag{11}$$

The Time is indicated by the average processing time for all the images in dataset. Our system was coded with Matlab and all experiments were evaluated on a desktop computer with Intel Core i5 1.7G CPU, 4G RAM, and Window 7 OS.

We implemented Epstein's method [26] because it has more similarity with our method. The data of Ashida, HWDavid, Wolf, and Todoran are excerpt from the results of completions [29]. The results of precision, recall, f-measure and time are shown in Table 1.

Figure 8. Text extraction in various languages

Algorithm	p'	r'	f	Time (sec.)
Our method	0.65	0.71	0.64	4.2
Epstein	0.62	0.67	0.62	12.1
Ashida	0.55	0.46	0.50	8.7
HWDavid	0.44	0.46	0.45	0.3
Wolf	0.30	0.44	0.35	17.0
Todoran	0.19	0.18	0.18	0.3

Table 1. Performance comparison of text extraction algorithm

The Table 1 shows the p', r' and f of our method are all the best. Moreover, the Epstein's time is about 3 times our method. Therefore, the improved SWT has notable promotion compared with original SWT.

In our method, the filter based improved SWT are the last one. In fact, the shape and texture rules are also exclude a part of non-text regions. In order to check the influence of SWT in our method, the filter of SWT is deleted from our work and the modified method is conducted. The results of p', r' and f are dropped from 0.65, 0.71, 0.64 to 0.53, 0.55, 0.63. Naturally the computing time are also dropped from 4.2 to 1.7. This experiment shows the performance are promoted by the filter of SWT, but the computing time are also increase more than one times.

In order to check the performance of our algorithm to non-English, we established a test database including Chinese, Japanese, Russian, Spanish, and Arabic etc. The database consists of 30 color images of size ranging from 640 × 480 to 1024 × 768, and the images collect from Internet. Our algorithm's performance on the database is as follows: precision p': 0.58, recall r': 0.63, f-measure: 0.52. The parts of results of text extraction are shown in Figure 8.

5. Conclusions and Future Work

The text extraction from natural scene images is still one of the most challenging researches. In this paper, a text

extraction algorithm in natural scene images using region-based method is proposed. The key of region-based method is to find a way to filter out non-text regions from candidate regions. In addition to the heuristic rules based on shape and texture, an improved stroke width rules are also applied in filtering out non-text regions. Unlike shape and texture features, the stroke width is the intrinsic and particular feature of texts. The major contribution is to improve the method to compute stroke width from two aspects: Shortening the computing time of SWT by reducing SWT range, and improving the accuracy of searching of edge point pair by expanding the search pixels. Experimental results show that our algorithm reached the first place in six published algorithm, and was 3 times faster than the speed of original SWT. Additionally, the detection of stroke width is independent in language and orientation of text.

Our method is best in accuracy and robustness, but is the third at speed. The major computing time spends the filter of non-text regions. It is necessary for complex images to conduct shape, texture and SWT filter. But for simple images, may only need a part of filters or do not need any filter. In the subsequent research, a classification for images will be considered. Input images are classified into several classes from simple to complex. The method of text extraction will dynamically select zero, a part of or all filters. The average processing time may reduce based on the average complexity in an image database.

6. Acknowledgement

This work was supported by the project kJ121606 from Chongqing Education Commission.

References

[1] Dimitrova, N. Hong-Jiang Zhang, et al. (2002). Applications of video content analysis and retrieval. *IEEE Multimedia*, 9 (3) 43–55.

[2] Lyu, M.R., Jiqiang Song, Min Ca. (2005). A comprehensive method for multilingual video text detection, localization, and extraction. *IEEE Trans*, CSVT 15 (2) 243–255.

[3] Datong Chen, Jean-Marc Odobez, et al. (2004). Text detection and recognition in images and video frames. *Pattern Recognition*, 37 (3) 595–608.

[4] Qixiang Ye, Qingming Huang, et al. (2005). Fast and robust text detection in images and video frames. *Image Vision Comput*, 23 (6) 565–576.

[5] Wactlar H. D., Christel M. G., et al. (1999). Lessons learned from building a terabyte digital video library. *Computer*, 32 (2) 66–73.

[6] Lienhart, R., Wernicke, A. (2002). Localizing and Segmenting Text in Images and Videos. *IEEE Transact. on Circuits and Systems for Video Technology*, 12 (4) 256-268.

[7] Jian Liang, David Doermann, et al. (2005). Camera-based analysis of text and documents: a survey. *International Journal of Document Analysis and Recognition*, 7 (2-3) 84-104.

[8] Jun Ohya,Shio, A. (1994). Recognition of characters in scene images. *IEEE Trans. Pattern Anal Machine Intell*, 16 (2) 214–220.

[9] Li Huiping, Doermann D., Kia O. (2000). Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing*, 9 (1) 147–156.

[10] Keechul Jung, Kwang In Kim, Anil K. Jain. (2004). Text information extraction in images and video: A survey. *Pattern Recognition*, 37 (5) 977–997.

[11] Jin-liang Yao, Yan-Qing Wang, et al. (2007). Locating text based on connected component and SVM. Wavelet Analysis and Pattern Recognition. ICWAPR'07. *International Conference on. IEEE*, 3, p. 1418-1423.

[12] Xilin Chen, Lei Yang, Jing Zhang, Waibel, Alex. (2004). Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on Image Processing*, 13 (1) 87-99.

[13] Kumar, M., Gueesang Lee. (2010). Automatic text location from complex natural scene images. Computer and Automation Engineering (ICCAE), The 2nd International Conference on, 3, p. 594-597.

[14] Ge Guo, Jin Jin, Xijian Ping, Tao Zhang. (2007). Automatic Video Text Localization and Recognition. *In*: Proc of Fourth International Conference on Image and Graphics, p. 484-489.

[15] Yan Song, Anan Liu, et al. (2008). A Novel Image Text Extraction Method Based on K-Means Clustering. *In*: Proc of Seventh IEEE/ACIS International Conference on Computer and Information Science, p. 185-190.

[16] JiSoo Kim, SangCheol Park, Soohyung Kim. (2005). Text locating from natural scene images using image intensities. *In*: Proc of Eighth International Conference on Document Analysis and Recognition, p. 655-659.

[17] Hae-Kwang Kim. (1996). Efficient automatic text location method and content-based indexing and structuring of video database. *J. Visual Commun. Image Representation*, 7 (4) 336–344.

[18] Kwang In Kim, Keechul Jung , Jin Hyung Kim. (2003). Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 (12) 1631-1639.

[19] Gao, J., Lei Yang. (2001). An adaptive algorithm for text detection from natural scenes. *In*: Proc of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), p. 84-89.

[20] Xiaodong Huang, Huadong Ma. (2010). Automatic Detection and Localization of Natural Scene Text in Video. *In*: Proc of 20th International Conference on Pattern Recognition (ICPR), p. 3216-3219.

[21] Saoi, T., Goto, H., Kobayashi, H. (2005). Text detection in color scene images based on unsupervised clustering of multi-channel wavelet features. *In*: Proc of Eighth International Conference on Document Analysis and Recognition, p. 690-694.

[22] Gllavata, J., Ewerth, R., Freisleben, B. (2004). Text Detection in Images Based on Unsupervised Classification of High-Frequency Wavelet Coefficients. *In*: Proc. of Int'l Conf. on Pattern Recognition, p. 425-428.

[23] Yan Hao,Zhang Yi, Hou Zeng-guang, Tan Min. (2003). Automatic Text Detection In Video Frames Based on Bootstrap Artificial Neural Network and CED. *Journal of WSCG*.

[24] Shivakumara, P., Trung Quy Phan, et al. (2010). New Fourier-Statistical Features in RGB Space for Video Text Detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 20 (11) 1520-1532.

[25] Cheolkon Jung, Qifeng Liu, Joongkyu Kim. (2009). A stroke filter and its application to text localization. *Pattern Recognition Letters*, 30 (2) 114-122.

[26] Brois Epstein, Eyal Ofek, Yonatan Wexler. (2011). Detecting Text in Natural Scenes with Stroke Width Transform. *In*: Proc of IEEE Conference on 2011 on Computer Vision and Pattern Recognition, p. 2963-2970.

[27] Canny, John. (1986). A Computational Approach To Edge Detection. *IEEE Trans*. *Pattern Analysis and Machine Intelligence*, 8 (6) 679-698.

[28] Hasan S M, Adjeroh D A. (2011). Detecting Human Sentiment from Text using a Proximity-Based Approach. *Journal of Digital Information Management*, 9 (5) 206-212.

[29] Lucas, S. M., Panaretos, A., et al. (2003). ICDAR 2003 robust reading competitions. *In*: Proc of Seventh International Conference on Document Analysis and Recognition, p. 682-687.

[30] Otsu, N. (1975). A threshold selection method from gray-level histogram. *Automatica*. 11 (285-296) 23-27.

[31] Hua Hu. (2014). Research on ontology construction and information extraction technology based on wordnet. *Journal of Digital Information Management*, 12 (2) 114-119.

[32] Yuan Debao, Cui Ximin, Xu Wanyang, et al. (2014). Research on the Application of SIFT Algorithm in UAV remote sensing image feature extraction. *Journal of Digital Information Management*, 12 (2) 67-72.