Jurnal Teknologi

Full Paper

TEXT LOCALIZATION IN IMAGES USING REVERSE THRESHOLDS ALGORITHM

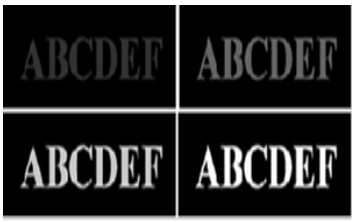
Lih-Fong Wong, Mohd Yazid Idris*, Abdul Hanan Abdullah

Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

Article history
Received
3 December 2013
Received in revised form
2 July 2014
Accepted
25 November 2014

*Corresponding author
yazid@utm.my

Graphical abstract



Abstract

High color similarity between text pixels and background pixels is the major problem that causes failure during text localization. In this paper, a novel algorithm, Reverse Thresholds (RT) algorithm is proposed to localize text from the images with various text-background color similarities. First, a rough calculation is proposed to determine the similarity index for every text region. Then, by applying reverse operation, the best thresholds for each text region are calculated by its similarity index. To remove other uncertainties, self-generated images with the same text features but different similarity index are used as experiment dataset. Experiment result shows that RT algorithm has higher localizing strength which is able to localize text in a wider range of similarity index.

Keywords: Text localization, color similarity, reverse threshold

1.0 INTRODUCTION

Text in image is an important source which it can represent the semantic content of the image itself. By extracting and processing the text, it can provide useful information to the user such as: geolocation, object information, audio subtitles, and so on. Text extraction and recognition can be done by using optical character recognition (OCR) software, but due to variety pattern of text, image quality and complicated image background, extracted text will not be accurate. Hence, localizing the position of text pixels and differentiation process between text pixels and background pixels are needed before sending to OCR. However, the processes may come out with the worst result if it encounters with the situation where the color of text pixels and color of background pixels are very similar or almost the same. This failure mainly caused by the fundamental concept of algorithm of text detection and localization where it analyses integer value of color in each pixels and determine whether the pixel belongs to text or background by specific rules. When text pixels and background pixels are having almost similar value, both will be treated as same pixel type (either text or background) and will be ignored during differentiation and text extraction algorithm. This will cause miss extraction (when the whole text is treated as background) and false extraction (when some key stroke of a word are treated as background, recognition software will recognize it as another word). The existence of high color similarity between text and background may cause by several reasons: exposure of flash or light source, transparent or engraved text and complex background color. Figure 1 shows the sample of the text images.

Figure 1 Image with high similarity of color between text pixels and background pixels. Top: engraved text; bottom left: exposure of light; bottom right: complex background

Since image processing is dealt with color pixels, theoretically it is impossible to differentiate if both text and background pixels are exactly the same. However, if there is even a tiny different between text and background pixels, it should be able to differentiate them. The interest of this research is to know how effective of a text localization algorithm can be done when handling cases where text and background pixels are almost similar and to propose a solution to improve the localizing efficiency. Based on review of previous research, the basic concept of existing text localization algorithms can be divided into three: Connected-Component based (CC), Texture based and Edge based.

CC based method groups the pixels which are connected to each other and having the value of color in a pre-set deviation range. The grouping continues on the upper level (group the result of grouping) until it remains two groups which represent text and background. For example, in [1], Sun *et al*. find the candidate of text and verify them by combining color and size range features. At the same time, Sushma and Padmaja also proposed CC based method by merging the homogeneity region of Y channel in YUV color space [2]. In year 2010, Liu *et al*. used color clustering and heuristic CC merging algorithm to group all the similar CC components [13]. In the method [3], Meng and Song extract salient region from image, and separate them by CC analysis method.

Texture based analyses the distribution of color in the whole image either by the origin image or transformed image (using wavelet transform, cosine transform etc.). By implementing machine learning or neural network approach, text pixels and position can be determined by matching the unique pattern of color distribution. In 2009, Emmanouilidis *et al*. proposed an idea to extract a texture feature called document structure elements in images and localize text by using SVM to classify those features [4]. Next in 2010, Angadi and Kodabagi used Discrete Cosine Transform (DCT) on every 8x8 block in image and classified the blocks by texture function and discriminant function to determine and localize text region [5]. On the other hand, Li *et al*. used wavelet transform to decompose image signal and perform 2D wavelet transform by using Haar wavelet to locate the text [6].

Edge based method detects the changes between value of color in pixels. Any sharp change of color, which exceeding the pre-set threshold will indicate as an edge. The edge will act as a separator which differentiates text pixels and background pixels. For example, Shivananda and Nagabhushan proposed an edge detection method using Canny's method in RGB color model [7]. Later in 2010, Dinh *et al*. used both Canny edge detector and gradient magnitude on images to localize text [8]. Last but not lease, Lelore and Bouchara [9] used Canny edge detector to find the edge image, and localize text by EM algorithm [10].

Based on observation, these three basic concepts have a common feature where they require certain amount of differences in the value of color between text pixels and background pixels in order to locate the text. If there is situation where the value of color for text pixels and background pixels are about the same, those methods will encounter with problems. For example, CC based algorithms will merge both text and background into one group if both of the value fall in the pre-set range. Texture based algorithm will not be able to track down the pixels pattern if the color changes are small. Edge based algorithms will ignore the edges from pixels which are having different of color value below the threshold. Therefore, color similarity between text pixels and background pixels denote to the major problem which degrade the text extraction result accuracy.

Previous researches do not focus exactly on text-background color similarity issue; they instead focus on the input from natural scene image where involve a lot of uncertainties other than color similarity. For example, Meng and Song tried to detect the text on natural scene image by looking for salient region [3]. They believe that accuracy of text localizing will be mainly affected by the complexity (which include text-background color similarity) of the images. Hence, they proposed an algorithm to simulate human visual behavior on purposely looking for text in an image by common salient features. During the same year, Liu *et al*. claim that precision of the text localizing result is mainly cause by color of text in natural images which may appears to be too bright or too dark [14] which in the sense that a brighter text or darker text used to be affect by its environment color. They tried to solve the problem by extracting text letter using Maximally Stable Extremal Regions and finally classify them by AdaBoost. In early 2013, Pillai *et al*. claims that the difficulty of current localization algorithm is due to the confusion between text component and non-text component [15]. They then proposed a hybrid algorithm from region based and CC based methods to encounter with this situation. Those researches review the issue of text localization in a higher level (image complexity) whereas in this paper, the issue is narrows down into a more specific region: the text-background color similarity, which is believed to be the major impact on affecting localizing result.

With that, it is important to have a measurement and calculation in order to determine the text-background color similarity in an image. The similarity index shall be the key point to break through the obstacle of current localization problem.

2.0 PROPOSED ALGORITHM

To deal with uncertainty of differences in colors between text pixels and background pixels that may exist in an image, a similarity index measurement is proposed to calculate the differences. Next, a solution

to localize text in the images with various similarity index ranges is proposed by using the similarity index itself.

2.1 Text-Backgroud Similarity Index

Based on previous research, so far there is no direct measurement method to show the index of color similarity of an image and the effectiveness of a text localization algorithm when dealing with it. To show the color similarity between text pixels and background pixels, a measurement index called Text-Background Similarity Index (*TBSI*) is introduced which is defined as the degree of likeness for pixels value between text and background in an image region.

To calculate *TBSI* for an image, it is first convert into a grayscale image . Next, marking on each of the ground truth regions of text in a rectangle box need to be done manually. Let $G = \{G_i \mid I = 1,2, \dots, N\}$ where *G* is the ground truth region and *N* is the total number of text region exist in. For each region *Gi Otsu's* binarization algorithm [2] is applied to get the approximated text pixels, G_i (t_i) and background pixels G_i (b_j). It does not matter if the value of G_i (t_j) and G_i (b_j) are reversed. Next, the average pixel value for each region G_i is obtained where G_i (t) the average value for text pixels is and G_i (b) is the average value for background pixels. Calculate the absolute color different between text pixels and background pixels $D(G_i)$ where

$$D(G_i) = |G_i(t) - G_i(b)| \quad (1)$$

Finally, the average value of *D(G)* for *I* is determined by the formula:

$$D(G) = \frac{1}{N}\sum\left(D(G_i)\right) \quad (2)$$

Note that, the average value of *D(G)* for *I* is determined by the sentence, as in

$$TBSI = 1 - \frac{D(G)}{D_{max}} \quad (3)$$

Where *Dmax* refers to maximum possible value different between text and background pixels. In this research, where the grayscale image is used, the *Dmax* will be 255 (Maximum different scenario is when two pixels with value of 0 as black and 255 as white). TBSI has a value range in between zero and one, where the bigger the value of *TBSI*, the higher the similarity between text pixels and background pixels. It should be considered invalid when *TBSI = 1* or *D(G)= 0* as this represent there is no different between text pixels and background pixels. This situation takes place when there is only one color in the ground truth text region.

2.2 Localizing Text by Multi Adaptive Threshold

TBSI measures the degree of likeliness between text pixels and background pixels. Hence, it is perfectly suitable to utilize it as the adaptive thresholds for edge detection by applying a low threshold value for high *TBSI* images and high threshold value for low *TBSI* images. Different from other approaches, the approach in this research uses multi adaptive thresholds on each of the text regions within an image to calculate the most suitable threshold values before apply edge detection algorithm. First, for an image *I*, Canny edge detector is applied to obtain an initial binary edge image *E* . Then, edge pixels are divided into several groups *E* , and let $E = \{e_i \mid I = 1, 2, \dots, N\}$ + where refers to a set of continuous edge pixels where the eight-connected neighbor of each edge pixels contain at least one of another edge pixel in the same group and denotes *N* to the total edge pixel groups for *E*. Next, for each edge pixel group e_i, a region $R_i(x_{min}, y_{min}, x_{max}, y_{max})$ is set up, where the region is enclosed by a minimum x and y coordinates (x^i_{min}, y^i_{min}) as well as maximum x and y coordinates (x^i_{min}, y^i_{min}) result from the overall edge pixels in e_i. This simply indicates that the region has covered the minimum surface area for each and each region is assumed to be containing only a single character or word. Therefore, the regions which are closed to each other shall assume to be the group with same feature (either text or noise). For each region, locate every region near by it, by searching the area around it by a distance of *J* in horizontal and *K* in vertical as Figure 2. *J* and *K* denote to the average width of a character and average height of a character respectively and they can be obtained by:

$$J = \frac{1}{N}\sum_{i=0}^{N}\left(x^i_{max} - x^i_{min}\right) \quad (4)$$

$$K = \frac{1}{N}\sum_{i=0}^{N}\left(y^i_{max} - y^i_{min}\right) \quad (5)$$

For regions which fall in the nearby range, degree deviation is calculated for both regions before merging them. Let the both regions to be e_1 and e_2 . Calculate how much e_1 deviated from e_2 and indicate it with $tan\theta_{min}$ and $tan\theta_{max}$ where $tan\theta_{max}$ refer to degree of deviation for the minimum x and y coordinate while refer to degree of deviation for the maximum x and y coordinate. Figure 3 illustrates the circumstances. $tan\theta_{min}$ and $tan\theta_{max}$ can be obtained by:

$$\tan\theta_{min} = \frac{|y^1_{min} - y^2_{min}|}{|x^1_{min} - x^2_{min}|} \quad (6)$$

$$\tan\theta_{max} = \frac{|y^1_{max} - y^2_{max}|}{|x^1_{max} - x^2_{max}|} \quad (7)$$

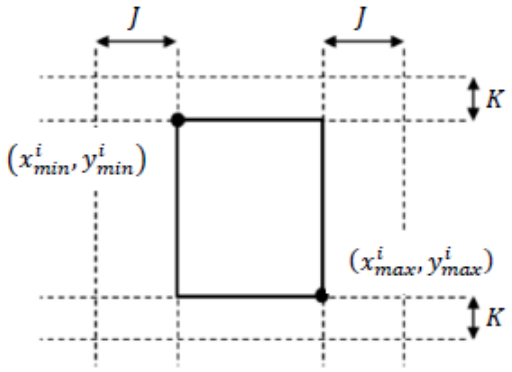


Figure 2 Extended search area region

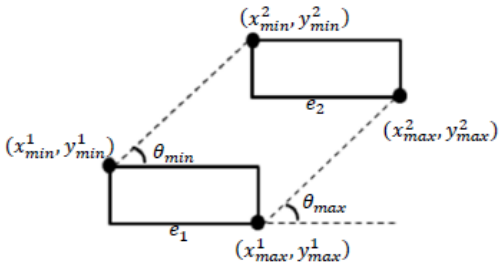


Figure 3 Degree deviation between two regions

Next, find the degree deviation θ_{dev} for both θ_{min} and θ_{max} by using the equation

$$\theta_{dev} = |\theta_{max} - \theta_{min}| \quad (8)$$

If the θ_{dev} is small, i.e. $\theta_{dev} < T_\theta$, both regions will be merged into one by readjusting the maximum and minimum x, y coordinates. T_θ is the maximum limit of deviation allowed. In the proposed system, loose strategy $T_\theta = 9°$ is taken, which equivalence to 10% of maximum possible deviation 90°. After merging, filter out the regions which are more likely to be noise by ignoring the region with either height or width is smaller than three pixels. Result of new edge pixel group groups E' will then be produced. Let $E' = \{e_i' \mid I = 1, 2, \ldots, M\}$ where e_i' refers to remaining regions and M is the total number of region. Next, each e_i' is assumed to contain a mixture of text and noise. Hence, *TBSI* calculation is performed for each e_i' and gets the similarity index, *TBSI* ,for that particular region. The value *TBSI* ,will be used as weight for the threshold of edge detector. Canny edge detection again is re-applied on each region e_i' separately on the original image by using thresholds obtains by:

$$T'_{low} = T_{low} \times (1 + \mu) \times (1 - TBSI_i) \quad (9)$$

$$T'_{high} = 2 \times T'_{low} \quad (10)$$

T_{low} refers to the original lower threshold used at the first stage μ and refers to markup factor for the threshold to reserve the upper limit growth from original threshold. In this research, the threshold is markup by 50% or $\mu = 0.5$. Other regions which do not fall in will be treated as non-text region and will not process. Figure 4 illustrates the circumstances. The final edge image reveals the localized text on the region where edge pixels are clustering.

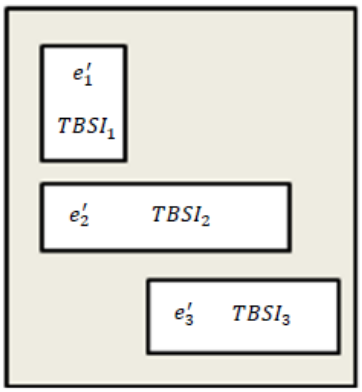


Figure 4 Second Canny Edge Detector Application: Area e_1': Applying Threshold from $TBSI_1$; Area e_2': Applying Threshold from $TBSI_2$; Area e_3': Applying Threshold from $TBSI_3$; Shadowed Area: Ignored

Figure 5 Image with Various TBSI. Top left: TBSI = 0.8039; Top right: TBSI = 0.6078, Bottom left: TBSI = 0.2157; Bottom right: TBSI = 0.0157

3.0 EXPERIMENTAL AND DISCUSSION

The purpose of this experiment is to evaluate the processing speed and localizing strength of experimental algorithms base on images with different text-background color similarity. As the name implies, localizing strength reviews the capability of an algorithm to localize text regardless of any circumstances. Strength of algorithm φ can be calculated by the surface area covered by the function: precision of similarity index -p (τ). Precision P is taken as $P = N_c / N_g$, where N_c is the total number of correct text region, and N_g is the total number of

ground truth text region. The text region is considered localized correctly only if the outcome of localized text region covered at least 80% of text region from ground truth region. Let the similarity index $\tau = TBSI$, where the values range of *TBSI* is from zero to they are part of a sentence, as in

$$\varphi = \int_0^1 p(\tau)\, d\tau \qquad (1)$$

To make it simple, this experiment uses a self-generated image dataset with clear and unique color for each text pixels and background pixels which will eliminate the other uncertainties. All the images will have the same texts and at the same position but difference of color between text pixels and background pixels. To show the feasibility of *TBSI*, all the possible differences of color between text pixels and background pixels are generated which they are made up of a total of 65280 (256 x 255) images. These images are further divided into groups by its *TBSI* value which consists 255of groups. Figure 5 shows some samples of the images. In this experiment, the proposed algorithm (RT algorithm) is compared to the other two algorithms: Stroke-Like Edge (SLE) algorithm [11] and Boundary Clustering (BC) algorithm [12] in terms of localizing strength and speed of processing. All the three algorithms are run by using the same dataset and on the same PC with Intel Core i7 2.00 GHz and 16GB memories. The experimental results are summarized in Table 1 and Figure 6.

Table 1 Strength and speed comparison of SLE algorithm [11], BC algorithm [12], and RT algorithm

Algorithm	Strength	Speed
SLE Algorithm [11]	0.545	0.46s
BC Algorithm [12]	0.560	0.11s
RT Algorithm	0.760	0.13s

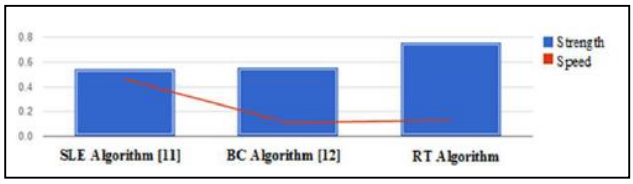


Figure 6 Strength and speed comparison of SLE algorithm [11], BC algorithm [12] and RT algorithm

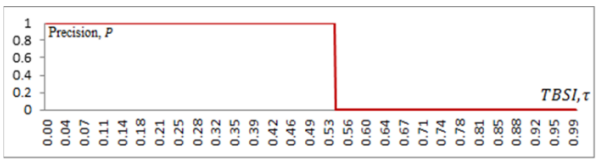


Figure 7 Precision of similarity index function for SLE algorithm [11]

Since the experiment dataset only consists of differences in similarity index, the precision of similarity index function will show a sharp decrease at a certain level of τ (See Figure 7). This simplifies the calculation of localizing strength where it can be obtained by directly taking the value at the sharp decrease point. Processing speed can be obtained by simply averaging the processing time for each image. Localizing strength is normalized to range between zero and one, and is used for judging the ability of an algorithm on localizing image with different *TBSI* (or precisely, image with high (*TBSI*). By observing Fig. 7, it shows that the algorithm always correct (*P = 1*) when the input image having *TBSI* value lower than 0.545, and at the same time, always fail (*P = 0*) when the input image having value higher than 0.545. Hence, the value 0.545 becomes a boundary or limitation of the algorithm and the value can be seen as the localizing strength of the algorithm.

Based on the result table, both SLE algorithm [11] and BC algorithm [12] are having a close value on their strength at around 0.55. Proposed RT algorithm shows its localizing strength with a value of 0.76, corresponding to 0.21 increments or equivalence to the improvement of 38.2% on average. Processing speed of RT algorithm is recorded as average 0.13 second on each image which is a great increase compare to SLE algorithm [11]. However, it is slightly slower than BC algorithm [12] which is mainly due to the repeatedly applying on Canny edge detector.

4.0 CONCLUSION

In this paper, RT algorithm is proposed to localize text in images with various *TBSI* by using multiple adaptive threshold approach. Experiment result which is using the self-generated image dataset clearly shows that RT algorithm can handle the localization efficiency and effectively. However, the proposed algorithm currently has a limitation where it only able to locate text with various color intensity but not been able to differentiate text pixels and noise. Hence, in the future work, the algorithm will enhance to be more sophisticated and able to separate text pixels and noise.

References

[1] Li Sun, Guizhong Liu, Xueming Qian, Danping Guo. 2009. A Novel Text Detection and Localization Method Based on Corner Response. Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on. 28 June– 3 July 2009. 390-393.

[2] Sushma, J., Padmaja, M. 2009. Text Detection in Color Images. Intelligent Agent & Multi-Agent Systems, 2009. IAMA 2009. International Conference on. 22-24 July 2009. 1-6.

[3] Quan Meng, Yonghong Song. 2012. Text Detection in Natural Scenes with Salient Region. Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on. 27-29 March 2012. 384-388.

[4] Emmanouilidis, C., Batsalas, C., Papamarkos, N. 2009. Development and Evaluation of Text Localization Techniques Based on Structural Texture Features and Neural Classifiers. Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on. 26-29 July 2009. 1270- 1274.

[5] Angadi, S. A., Kodabagi, M. M. 2010. Text Region Extraction from Low Resolution Natural Scene Images Using Texture Features. Advance Computing Conference (IACC), 2010 IEEE 2nd International. 19-20 Feb. 2010. 121-128.

[6] Li, Z., Liu, G., Qian, X., Guo, D., Jiang, H. 2011. Effective and Efficient Video Text Extraction Using Key Text Points. *Image Processing, IET*. 5(8): 671-683.

[7] Shivananda, N., Nagabhushan, P. 2009. Separation of Foreground Text from Complex Background in Color Document Images. Advances in Pattern Recognition, 2009. ICAPR '09. Seventh International Conference on. 4-6 Feb. 2009. 306-309.

[8] Dinh, T. N., J. Park, G. Lee. 2010. Text localization using Image Cues and Text Line Information. Image Processing (ICIP), 2010 17th IEEE International Conference on. 26-29 Sept. 2010. 2261-2264.

[9] Lelore, T.; Bouchara, F. 2011. Super-Resolved Binarization of Text Based on the FAIR Algorithm. Document Analysis and Recognition (ICDAR), 2011 International Conference on. 18-21 Sept. 2011.839-843.

[10] Dempster, A. P., Laird, N. M., & Rubin, D. B. 1977. Maximum Likelihood from Incomplete Data via the Em Algorithm. *J. of the Royal Statistical Society B.* 39(1): 1-38.

[11] Liu, X., W. Wang. 2012. Robustly Extracting Captions in Videos Based on Stroke-Like Edges and Spatio-Temporal Analysis. *Multimedia, IEEE Transactions on.* 14(2): 482-489.

[12] Yi, C., Ying Li Tian. 2012. Localizing Text in Scene Images by Boundary Clustering, Stroke Segmentation, and String Fragment Classification. Image Processing. *IEEE Transactions on.* 21(9): 4256-4268.

[13] Liu, J., S. Zhang, H. Li, W. Yang. 2010. A Novel Method For Flash Character Localization. Audio Language and Image Processing (ICALIP), 2010 International Conference on. 23-25 Nov. 2010. 811-814.

[14] Yin, X., Yin, X. C., Hao, H. W., Iqbal, K. 2012. Effective Text Localization in Natural Scene Images with MSER, Geometry-Based Grouping and AdaBoost. Pattern Recognition (ICPR), 2012 21st International Conference on. 11-15 Nov. 2012. 725-728.

[15] Pillai, A. V., Balakrishnan, A. A., Simon, R. A., Johnson, R. C., Padmagireesan, S. 2013. Detection and Localization of Texts from Natural Scene Images Using Scale Space and Morphological Operations. Circuits, Power and Computing Technologies (ICCPCT), 2013 International Conference on. 20-21 March 2013. 880-885.