MULTISCALE EDGE-BASED TEXT EXTRACTION FROM COMPLEX IMAGES

Xiaoqing Liu and Jagath Samarabandu

The University of Western Ontario
Department of Electrical & Computer Engineering
London, Ontario, N6A 5B9, Canada
xliu65@uwo.ca, jagath@uwo.ca

ABSTRACT

Text that appears in images contains important and useful information. Detection and extraction of text in images have been used in many applications. In this paper, we propose a multiscale edge-based text extraction algorithm, which can automatically detect and extract text in complex images. The proposed method is a general-purpose text detection and extraction algorithm, which can deal not only with printed document images but also with scene text. It is robust with respect to the font size, style, color, orientation, and alignment of text and can be used in a large variety of application fields, such as mobile robot navigation, vehicle license detection and recognition, object identification, document retrieving, page segmentation, etc.

1. INTRODUCTION

The automatic detection of Region of Interests (ROI) is an active research area in the design of machine vision systems. Text embedded in images contains large quantities of useful semantic information which can be used to fully understand images. Text appears in images either in the form of documents such as scanned CD/book covers or video images. Video text can broadly be classified into two categories: overlay text and scene text. Overlay text refers to those characters generated by graphic titling machines and superimposed on video frames/images, such as video captions, while scene text occurs naturally as a part of scene, such as text in information boards/signs, nameplates, food containers, etc.

Automatic detection and extraction of text in images have been used in many applications. Document text localization can be used in the applications of page segmentation, document retrieving, address block location, etc. Content-based image/video indexing is one of the typical applications of overlay text localization. Scene text extraction can be used in mobile robot navigation to detect text-based landmarks, vehicle license detection/recognition, object identification, etc.

We are looking into algorithms that can perform general-purpose text localization. However, due to the variety of font size, style, orientation, alignment as well as the complexity of the background, designing a robust general algorithm, which can effectively detect and extract text from both types of images, is full of challenges.

Wang et al. [1] proposed a connected-component based method which combines color clustering, a black adjacency graph (BAG), an aligning-and-merging-analysis scheme and a set of heuristic rules together to detect text in the application of sign recognition such as street indicators and billboards. As the author mentioned, uneven reflections result in incomplete character segmentation which increases the false alarm rate in this method. Kim et al. [2] implemented a hierarchical feature combination method to implement text extraction in natural scenes. However, authors admit that this method could not handle large text very well due to the use of local features that represents only local variations of image blocks. Gao et al. [3] developed a three layer hierachical adaptive text detection algorithm for natural scenes. This method has been applied in a prototype Chinese sign translation system which mostly has a horizontal and/or vertical alignment. We proposed a statistics-based method [4] to detect and localize text-based features by calculating the spatial intensity variation. This method is very simple and fast. However, in real scenes, due to uneven illumination, reflections and shadows, an image background may contain areas with high spatial intensity variation that do not contain text. Our experiments showed that this algorithm did not perform well under some situations. This led to the development of a more robust algorithm based on edges, a single scale edge-based text region extraction algorithm [5] for indoor scene images, which is robust with respect to font sizes, styles, color/intensity, orientations, effects of illumination, reflections, shadows, and perspective distortion.

In this paper, we propose a multiscale edge-based text extraction algorithm, a general-purpose method, which can quickly and effectively localize and extract text from both document and indoor/ outdoor scene images.

Financial support for this research is provided by the Natural Sciences and Engineering Research Council (NSERC), IRIS Precarn and the University of Western Ontario.

2. PROPOSED METHOD

The proposed method is based on the fact that edges are a reliable feature of text regardless of color/intensity, layout, orientations, etc. Edge strength, density and the orientation variance are three distinguishing characteristics of text embedded in images, which can be used as main features for detecting text. The proposed method consists of three stages: candidate text region detection, text region localization and character extraction.

2.1. Candidate Text Region Detection

This stage aims to build a feature map by using three important properties of edges: edge strength, density and variance of orientations. The feature map is a gray-scale image with the same size of the input image, where the pixel intensity represents the possibility of text.

2.1.1. Multi-scale edge detector

In our proposed method, we use magnitude of the second derivative of intensity as a measurement of edge strength as this allows better detection of intensity peaks that normally characterize text in images. The edge density is calculated based on the average edge strength within a window. Considering effectiveness and efficiency, four orientations (0^o, 45^o, 90^o, 135^o) are used to evaluate the variance of orientations, where 0^o denotes horizontal direction, 90^o denotes vertical direction, and 45^o and 135^o are the two diagonal directions, respectively. A convolution operation with a compass operator (as shown in Fig. 1) results in four oriented edge intensity images $E(\theta), (\theta \in \{0, 45, 90, 135\})$, which contain all the properties of edges required in our proposed method. Edge detector is carried out by using a multiscale strategy, where the multiscale images are produced by Gaussian pyramids which successively low-pass filter and down-sample the original image reducing image in both vertical and horizontal directions. In our proposed method, those obtained multiscale images are simultaneously processed by the compass operator as individual inputs.

-1	-1	-1
2	2	2
-1	-1	-1

0° kernel

-1	-1	2
-1	2	-1
2	-1	-1

45° kernel

-1	2	-1
-1	2	-1
-1	2	-1

90° kernel

2	-1	-1
-1	2	-1
-1	-1	2

135° kernel

Fig. 1. Compass operator

2.1.2. Feature map generation

As we mentioned before, regions with text in them will have significantly higher values of average edge density, strength and variance of orientations than those of non-text regions. We exploit these three characteristics to generate a feature map which suppresses the false regions and enhances true candidate text regions. This procedure is described in Eq.1.

$$fmap(i,j) = \bigoplus_{s=0}^{n} \sum_{\theta} N\{ \sum_{x=-c}^{c} \sum_{y=-c}^{c} E(s, \theta, i+x, j+y) \times W(i,j)\} \quad (1)$$

In the above equation, $fmap$ is the output feature map, $\bigoplus$ is an across-scale addition operation, which employs the scale fusion. n is the highest level of scale, which is determined by the resolution (size) of the input image. Generally speaking, the higher the resolution is, the more scales can be used. In our implementation, we use two scales for images with resolution of 640×480. $\theta \in \{0, 45, 90, 135\}$ are different orientation and N is a normalization operation. (i, j) are coordinates of an image pixel. $W(i, j)$ is the weight for pixel (i, j), whose value is determined by the number of edge orientations within a window. The window size is determined by a constant c. Namely, the more orientations a window has, the larger weight the center pixel has. By employing this non linear weight mapping, the proposed method distinguishes text regions from texture-like regions, such as window frames, wall patterns, etc.

2.2. Text Region Localization

Normally, text embedded in an image appears in clusters, i.e., it is arranged compactly. Thus, characteristics of clustering can be used to localize text regions. Since the intensity of the feature map represents the possibility of text, a simple global thresholding can be employed to highlight those with high text possibility regions resulting in a binary image. A morphological dilation operator can easily connect the very close regions together while leaving those whose position are far away to each other isolated. In our proposed method, we use a morphological dilation operator with a 7×7 square structuring element to the previous obtained binary image to get joint areas referred to as text blobs. Two constraints are used to filter out those blobs which do not contain text [5], where the first constraint is used to filter out all the very small isolated blobs whereas the second constraint filters out those blobs whose widths are much smaller than corresponding heights. The retaining blobs are enclosed in boundary boxes. Four pairs of coordinates of the boundary boxes are determined by the maximum and minimum coordinates of the top, bottom, left and right points of the corresponding blobs. In order to avoid missing those character pixels which lie near or outside of the initial boundary, width and height of the boundary box are padded by a small amounts.

2.3. Character Extraction

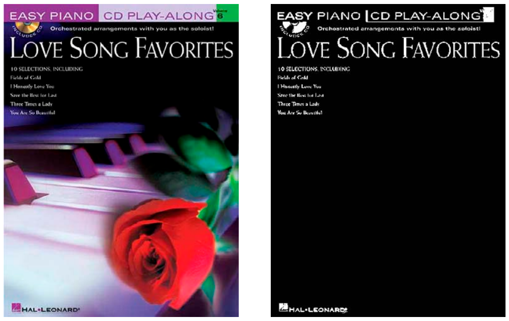
Existing OCR (Optical Character Recognition) engines can only deal with printed characters against clean backgrounds and can not handle characters embedded in shaded, textured or complex backgrounds. The purpose of this stage is to extract accurate binary characters from the localized text regions so that we can use the existing OCR directly for recognition. In our proposed method, we use uniform white character pixels in a pure black ground by using Eq.2.

$$T = \bigcup_{i=1\ldots} \overline{|SUB_i|}_z \qquad (2)$$

In the above equation, T is the text extracted binary output image. $\bigcup$ is an union operation. SUB_i are sub-images of the original image, where i indicates the number of sub-images. Sub-images are extracted according to the obtained boundary boxes in stage two. $\overline{|\cdot|}_z$ is a thresholding algorithm which segments the text regions into white characters in a pure black background.

3. EXPERIMENTAL RESULTS AND DISCUSSION

In order to evaluate the performance of the proposed method. We use 75 test images of four types including book covers, object labels, indoor lab nameplates and outdoor information signs, in which text has different font sizes, colors, orientations, alignments, perspective projection under different lighting conditions. Fig. 2 $\sim$ 5 show some of the results.

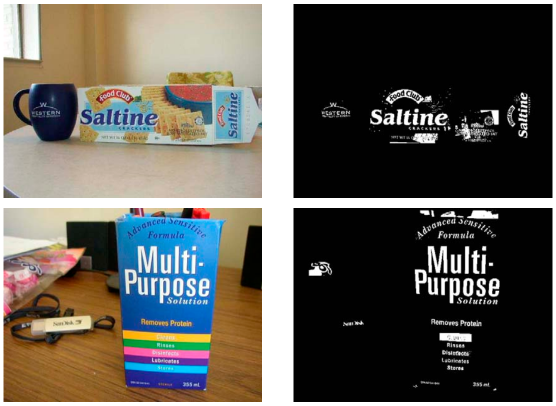


(a) (b)

Fig. 2. Book cover image (a) Original images (b) Extracted text, cover images are collected from google/yahoo web sites

From Fig. 2 $\sim$ 5, we can see that the performance of our proposed method on a wide variety of image set is excellent overall. Therefore, we can conclude that the proposed method is a robust and effective approach to detect text-based features in complex images.

Table 1 shows the performance comparison of our proposed method with several existing methods, where our proposed method shows a clear improvement over existing methods. In this table, the performance statistics of other methods are cited from published work. Considering a 95% confidence interval, it appears that Wang et al. [1], Xi et al. [6] and Gllavata et al. [7] have a similar performance as the proposed method. However, our proposed methods are suitable for more types of images.



(a) (b)

Fig. 3. Object label image with different font sizes, colors and orientational alignments (a) Original images (b) Extracted text

Table 1. Performance Comparison

Method	Image Source No.	Image Type	Precision Rate (%)	Recall Rate (%)
Proposed method	75	Four types	91.8	96.6
Wang et al.[1]	325	Outdoor scene	89.8	92.1
Kim et al. [2]	—	Outdoor scene	63.7	82.8
Agnihotri et al.[8]	293	Text captions	85.8	85.3
Xi et al.[6]	90	Text captions	88.5	94.7
Wolf et al.[9]	60	Text captions	—	93.5
Gao et al.[3]	—	Outdoor scene	—	93.3
Gllavata et al. [7]	326	Text captions	83.9	88.7
Messelodi et al. [10]	100	Document	—	91.2

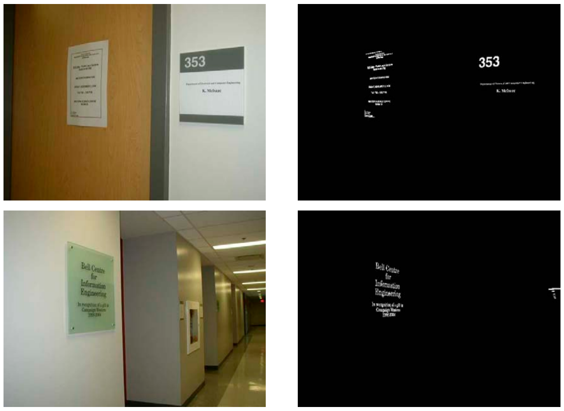
The overall average computation time for 75 test images (with 480 × 640 resolution) using unoptimized matlab codes on a personal laptop with Intel Pentium(R) 1.8GHZ processor and 1.0G RAM is 14.5 seconds ($stddev. = 0.156$), which includes entire run time including image reading, computation as well as image display.

4. CONCLUSION

In this paper, we present an effective and robust general-purpose text detection and extraction algorithm, which can automatically detect and extract text from complex background images. Our main future work involves using a suitable existing OCR technique to recognize the extracted text. The contributions of the proposed method are:

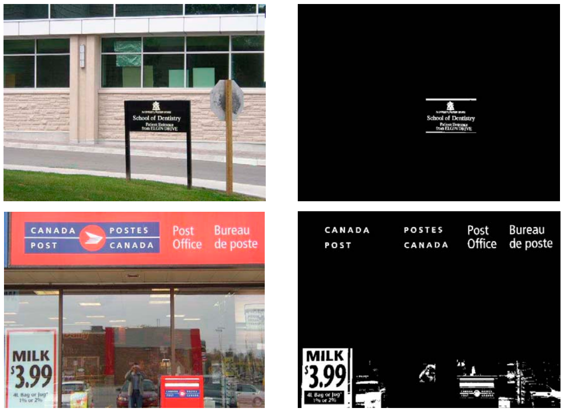
(a) can handle both printed document and scene text images.

(b) Not sensitive to image color/intensity, robust with respect to font, sizes, orientations, alignment, uneven illumination, perspective and reflection effects.



(a) (b)

Fig. 4. Indoor nameplate images with different font sizes, perspective distortion, colors and strong reflections (a) Original images (b) Extracted text



(a) (b)

Fig. 5. Outdoor sign image (a) Original images (b) Extracted text

(c) Unlike commonly used connected component based methods which analyze every single character, the proposed method only analyzes text blocks. Therefore, it is computationally efficient, which is essential for real-time applications.

(d) Distinguishes text regions from texture-like regions, such as window frames, wall patterns, etc., by using the variance of edge orientations.

(e) Binary output can be directly be used as an input to an existing OCR engine for character recognition without any further processing.

5. REFERENCES

[1] Kongqiao Wang and Jari A. Kangas, “Character location in scene images from digital camera,” *Pattern Recognition*, vol. 36, no. 10, pp. 2287–2299, 2003.

[2] K. C. Kim, H. R. Byun, Y. J. Song, Y. M. Choi, S. Y. Chi, K. K. Kim, and Y. K. Chung, “Scene text extraction in natural scene images using hierarchical feature combining and verification,” in *Pattern Recognition, 2004*, Aug. 2004, vol. 2 of *ICPR 2004. Proceedings of the 17th International Conference on*, pp. 679–682.

[3] Jiang Gao and Jie Yang, “an adaptive algorithm fot text detection from natural scenes,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001*, 2001, Proceedings of the 2001 IEEE Computer Society Conference on, pp. II–84–II–89.

[4] X. Liu and J. Samarabandu, “A simple and fast text localization algorithm for indoor mobile robot navigation,” in *Proc. of the SPIE- IS&T Electronic Imaging 2005*, San Jose, California, USA, Jan. 2005, vol. SPIE vol. 5672, pp. 139–150.

[5] X. Liu and J. Samarabandu, “An edge-based text region extraction algorithm for indoor mobile robot navigation,” in *Proc. of the IEEE International Conference on Mechatronics and Automation (ICMA 2005)*, Niagara Falls, Canada, July 2005, pp. 701–706.

[6] Jie Xi, Xian Sheng Hua, Xiang Rong Chen, Liu Wenyin, and Hong Jiang Zhang, “A video text detection and recognition system,” in *Multimedia and Expo, 2001. ICME 2001*, 2001, IEEE International Conference on, pp. 873–876.

[7] J. Gllavata, R. Ewerth, and B. Freisleben, “A robust algorithm for text detection in images,” in *Image and Signal Processing and Analysis, 2003. ISPA 2003*, 2003, Proceedings of the 3rd International Symposium on, pp. 611–616.

[8] L. Agnihotri and N. Dimitrova, “Text detection for video analysis,” in *Content-Based Access of Image and Video Libraries, 1999. (CBAIVL ’99)*, 1999, Proceedings. IEEE Workshop on, pp. 109–113.

[9] C. Wolf, J. M. Jolion, and F. Chassaing, “Text localization, enhancement and binarization in multimedia documents,” in *Pattern Recognition, 2002*, Aug. 2002, vol. 2 of *Proceedings. 16th International Conference on*, pp. 1037–1040.

[10] S. Messelodi and C. M. Modena, “Automatic identification and skew estimation of text lines in real scene images,” *Pattern Recognition*, vol. 32, no. 5, pp. 791–810, 1999.