Assistive Text Reading from Complex Background for Blind Persons

Chucai Yi[1,2] and Yingli Tian[1,2],

[1] Media Lab, Dept. of Electrical Engeering, The City College of New York, City Univ. of New York, 160 Convent Avenue, New York, NY, USA, 10031
[2] Dept. of Computer Science, The Graduate Center, City Univ. of New York, 365 Fifth Avenue, New York, NY, USA, 10016

cyi@gc.cuny.edu, ytian@ccny.cuny.edu

Abstract. In the paper, we propose a camera-based assistive system for visually impaired or blind persons to read text from signage and objects that are held in the hand. The system is able to read text from complex backgrounds and then communicate this information aurally. To localize text regions in images with complex backgrounds, we design a novel text localization algorithm by learning gradient features of stroke orientations and distributions of edge pixels in an Adaboost model. Text characters in the localized regions are recognized by off-the-shelf optical character recognition (OCR) software and transformed into speech outputs. The performance of the proposed system is evaluated on ICDAR 2003 Robust Reading Dataset. Experimental results demonstrate that our algorithm outperforms previous algorithms on some measures. Our prototype system was further evaluated on a dataset collected by 10 blind persons, with the system effectively reading text from complex backgrounds.

Keywords: blind person; assistive text reading; text region; stroke orientation; distribution of edge pixels; OCR;

1 Introduction

According to the statistics in 2002 [14], more than 161 million persons suffer visual impairment, in which there are 37 million blind persons. It is a challenging task for blind persons to find their way in unfamiliar environments, for example, independently finding the room they are looking for. Many aid systems have been developed to help blind persons avoid obstacles in all kinds of environments [3]. Some indoor positioning systems modeled global layout of a specific zone and used radio wave analysis to locate the persons wearing signal transceivers [13]. Some systems employed Quick Response (QR) codes to guide blind persons to destinations. However, most of these systems require pre-installed devices or pre-marked QR codes. Also, the blind user needs to consider compatibility of different systems. Therefore, the above systems cannot provide blind users these services in environments without pre-installed devices or markers. However, most blind persons can find nearby walls and doors, where text signage is always placed to indicate the

room number and function. Thus blind persons will be well navigated if a system can tell them what the nearby text signage says. Blind persons will also encounter trouble in distinguishing objects when shopping. They can receive limited hints of an object from its shape and material by touch and smell, but miss descriptive labels printed on the object. Some reading-assistive systems, such as voice pen, might be employed in this situation. They integrate OCR software to offer the function of scanning and recognition of text for helping blind persons read print documents and books. However, these systems are generally designed for scanned document images with simple background and well-organized characters rather than packing box with multiple decorative patterns. The OCR software cannot directly handle the scene images with complex backgrounds. Thus these assistive text reading systems usually require manual localization of text regions in a fixed and planar object surface, such as a screen and book.

To more conveniently assist blind persons in reading text from nearby signage or objects held in the hand, we design a camera-based assistive text reading system to extract significant text information from objects with complex backgrounds and multiple text patterns. The tasks of our system are indoor object detection to find out nearby wall, door, elevator or signage, and text extraction to read the involved text information from complex backgrounds. This paper focuses only on the step of text extraction, including 1) text localization to obtain image regions containing text, and 2) text recognition to transform image-based information into text codes [20]. Fig. 1 illustrates two examples of our proposed assistive text reading system. In order to perform text recognition by off-the-shelf OCR software, text regions must be detected and binarized. However, the problem of automatic localization of text regions from camera captured images with complex backgrounds has not been solved. For our application, text in camera captured images is most likely surrounded by various background outliers, and text characters usually appear in multiple scales, fonts, colors, and orientations. In this paper, we propose a novel algorithm of text localization based on gradient features of stroke orientations and distributions of edge pixels.

Fig. 1. Two examples of text localization and recognition from camera captured images. Top: a milk box; Bottom: a male bathroom. From left to right: camera-captured images, localized text regions (marked in cyan), text regions, and text codes recognized by OCR.

1.1 Previous Work in Text Localization

Many algorithms were presented to localize text regions in scene images. We divide them into two categories. The first category are rule-based algorithms that applied pixel level image processing to extract text information by predefined text features such as character size, aspect ratio, edge density, character structure, and color uniformity of text string, etc. Phan et al. [12] modeled edge pixel density by Laplacian operator and maximum gradient difference to calculate text regions. Shivakumara et al. [17] used gradient difference map and global binarization to obtain text regions. Epshtein et al. [4] used the consistency of text stroke width and defined stroke width transform to localize text characters. Nikolaou et al. [10] applied color reduction to extract text in uniform colors. This type of algorithms tried to define a universal feature descriptor of text. In [2], color based text segmentation is performed through a Gaussian mixture model for calculating confidence value of text regions. The second category are learning-based algorithms that apply explicit machine learning models on feature maps of training samples to extract robust text features and build text classifiers. Chen et al. [1] presented 5 types of block patterns on intensity based and gradient based feature maps to train classifiers in Adaboost learning model. Kim et al. [6] considered text as specific texture and analyzed the textural features of characters by support vector machine (SVM) model. Kumar et al. [7] used the responses from Globally Matched Wavelet (GMW) filters of text as features and applied SVM and Fisher classifier for image window classification. Ma et al. [9] performed classification of text edges by using HOG and LBP as local features on the SVM model. Shi et al. [16] used gradient and curvature features to model the gray scale curve for handwritten numeral recognition under a Bayes discriminate function. In this paper, we propose a text localization algorithm by defining novel feature maps based on stroke orientations and edge distributions.

2 System and Algorithm Overview

Our prototype system is equipped with a wearable camera attached to a cap or pair of sunglasses, an audio output device such as Bluetooth or earphones, and a mini-microphone for user speech input. This simple hardware structure ensures the portability of the system. A wearable computer/PDA provides the platform for information processing.

Fig. 2 depicts the main components of the prototype system. Blind users wearing cameras capture signage and objects they are facing. The camera captured images are then processed by our novel proposed text localization algorithm to detect text regions. In this text localization method, the basic processing cells are rectangle image patches with fixed aspect ratio, where features of text are extracted from both stroke orientations and edge distributions. In the training process, a feature matrix from the training set is formed as the input of an Adaboost machine learning algorithm to build a text region classifier. In the testing process, an adjacent character grouping algorithm is first applied on camera captured natural scene images to preliminarily localize the candidate image patches [19]. The classifier learned from

Adaboost algorithm is employed to classify the text or non-text patches, where neighboring text patches are merged into text regions. Then off-the-shelf OCR software is employed to perform text recognition in the localized text regions. The recognized words are transformed into speech for blind users.

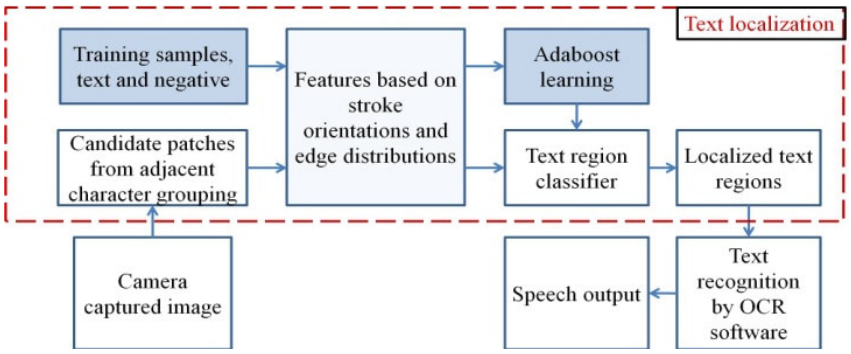

Fig. 2. Flowchart of our system.

The main contributions of this paper include: (1) a novel algorithm of automatic text localization to extract text regions from complex background and multiple text patterns; (2) a camera-based assistive prototype system to aid blind persons reading text from signage in unfamiliar environments and other objects; and (3) a dataset of objects and signage captured by blind persons for assistive text reading system evaluations.

3 Automatic Text Localization

We design a learning based algorithm of automatic text localization. In order to handle complex backgrounds, we propose two novel feature maps to extracts text features based on stroke orientation and edge distribution respectively. Here stroke is defined as a uniform region with bounded width and extensive length. These feature maps are combined to build an Adaboost-based text classifier.

3.1 Text Stroke Orientation

Text characters consist of strokes in different orientations as the basic structure. Here, we propose a new type of features, stroke orientations, to describe the local structure of text characters. From the pixel-level analysis, stroke orientation is perpendicular to the gradient orientations at pixels of stroke boundaries, as shown in Fig. 3. To model the text structure by stroke orientations, we propose a new operator to map gradient feature of strokes to each pixel. It extends local structure of stroke boundary into its neighborhood by gradient orientations. It provides a feature map to analyze global structures of text characters.

Given an image patch I, Sobel operators in horizontal and vertical derivatives are used to calculate 2 gradient maps $G_x$and $G_y$respectively. The synthesized gradient map is calculated as $G = \left(G_x{}^2 + G_y{}^2\right)^{1/2}$. Canny edge detector is applied on I to calculate its binary edge map E. For a pixel p_0, a circular range is set as $R(p_0) = \{p|d(p, p_0) \leq 36\}$, where $d(.)$ is set as Euclidean distance. In this range we find out the edge pixel p_e with the minimum Euclidean distance from p_0. Then the pixel p_0 is labeled with gradient orientation at the pixel p_e from gradient maps by (1), where $P = \{p|p \in R(p_0), p \text{ is edge pixel}\}$ and Υ normalizes stroke orientation into the range $(3\pi/2, 5\pi/2]$, which shifts forward one period from $(-\pi/2, \pi/2]$ to avoid the value 0, because $S(p_0)$ is set as 0 if and only if no edge pixel is found out within the range of pixel p_0.

$$p_e = \underset{p \in P}{\operatorname{argmin}}\, d(p, p_0)$$
$$S(p_0) = \Upsilon\left(\arctan\left(G_y(p_e),\ G_x(p_e)\right)\right) \tag{1}$$

A stroke orientation map $S(p)$ is output by assigning each pixel the gradient orientation at its nearest edge pixel, as shown in Fig. 4(a). The pixel values in stroke orientation map are then quantized into an N bin histogram in the domain $(3\pi/2, 5\pi/2]$ (see Fig. 4(b)). In feature extraction, strokes with identical or similar orientations are employed to describe structure of text from one perspective. In the N bin histogram, we group the pixels at every d consecutive bins together to generate a multi-layer stroke orientation map, where strokes in different orientations are separated into different layers. Without considering the cyclic shifts of the bins, there are totally $N - d + 1$ layers. In our evaluation, d is set to be 3 and N is set to be 16 respectively. Thus each sample generates 14 layers of stroke orientation maps, where text structure is described as gradient features of stroke orientations. We can extract structural features of text from stroke orientation maps.

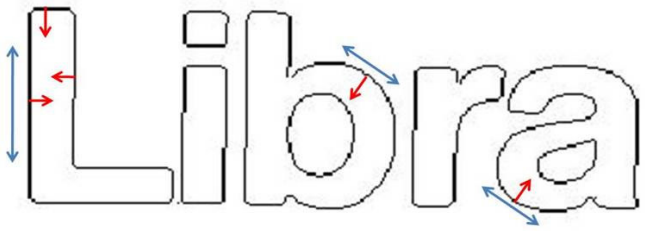

Fig. 3. An example of text strokes and relationship between stroke orientations and gradient orientations at pixels of stroke boundaries. Blue arrows denote the stroke orientations at the sections and red arrows denote the gradient orientations at pixels of stroke boundaries.

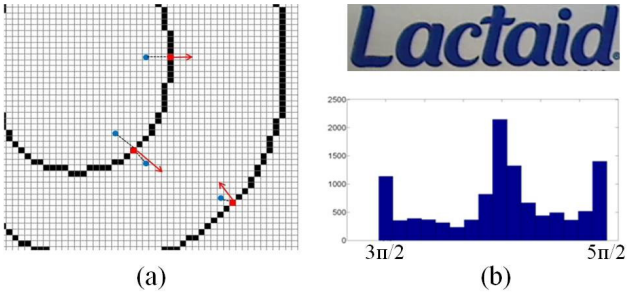

Fig. 4. (a) An example of stroke orientation label. The pixels denoted by blue points are assigned the gradient orientations (red arrows) at their nearest edge pixels, denoted by the red points. (b) A 210 ×54 text patch and its 16-bin histogram of quantized stroke orientations.

3.2 Distributions of Edge Pixels

In an edge map, text character appears in the form of stroke boundaries. Distribution of edge pixels in stroke boundaries also describes the characteristic structure of text. The most commonly used feature is edge density of text region. But edge density measure does not give any spatial information of edge pixels. It is generally used for distinguishing text regions from relatively clean background regions. To model text structure by spatial distribution of edge pixels, we propose an operator to map each pixel of an image patch into the number of edge pixels in its cross neighborhood. At first, edge detection is performed to obtain an edge map, and the number of edge pixels in each row y and each column x is calculated as N_R (y) and N_C (x). Then each pixel is labeled with the product value of the number of edge pixels in its located row and that in its located column. Based on this transform, the feature map of edge distribution is calculated by assigning each pixel weighed sum of the neighborhood centered at it, as (2). In the feature map of edge distribution, pixel value reflects edge density of its located region.

$$D(x,y) = \sum_{n} w_n \cdot N_R(y_n) \cdot N_C(x_n) \tag{2}$$

where (x_n, y_n) is neighboring pixel of (x, y) and w_n denotes the weight value.

3.3 Adaboost Learning of Features of Text

Based on the feature maps of gradient, stroke orientation and edge distribution, a classifier of text is trained from Adaboost learning model. Image patches with fixed size (height 48 pixels, width 96 pixels) are collected from images of ICDAR 2011 robust reading competition [21] to generate a training set for learning features of text. We generate positive training samples by scaling or slicing the ground truth text regions, according to the ratio of width w to height h. If the ratio is $w/h < 0.8$, the

region is discarded. If the ratio w/h falls in $[0.8,2.5)$, the ground truth region is scaled to a window of width-to-height ratio $2:1$. If the ratio is $w/h \geq 2.5$, we slice this ground truth region into overlapped training samples with width-to-height ratio 2:1. Then they are scaled into width 96 and height 48 pixels. The negative training samples are generated by extracting the image regions containing edge boundaries of non-text objects. These regions also have width-to-height ratio 2:1, and then we scale them into width 96 and height 48. In this training set, there are total 15301 positive samples and each contains several text characters with compatible accommodation of image patch, and 35933 negative samples without containing any text information for learning features of background outliers. Some training examples are shown in Fig. 5.

Fig. 5. Examples of training samples with width-to-height ratio 2:1. The first two rows present positive samples and the remaining two rows present negative samples.

To train the classifier, we extract 3 gradient maps, 14 stroke orientation maps, and 1 edge distribution map for each training sample. We apply 6 block patterns [1] on these feature maps of training samples. As shown in Fig. 6, these block patterns are involved in the gradient distributions of text in horizontal and vertical directions. We normalize the block pattern into the same size (height 48 pixels, width 96 pixels) as training samples and derive a feature response f of a training sample by calculating the absolute difference between the sum of pixel values in white regions and the sum of pixel values in black regions. For the block patterns with more than 2 sub-regions (see Fig. 6(a-d)), the other metric of feature response is the absolute difference between the mean of pixel values in white regions and the mean of pixel values in black regions. Thus we obtain $6 + (6 - 2) = 10$ feature values through the 6 block patterns and 2 metrics from each feature map. The "integral image" algorithm is used in these calculations [18]. From the 18 feature maps (3 gradient maps, 14 stroke orientation maps, and 1 edge distribution map), a training sample can generate a feature vector of 180 dimensions as (3). Then we compute feature vectors for all the 51234 samples in training set. By using the feature vector $\boldsymbol{f}^i$ of the *i*-th sample as the *i*-th column**,** a feature matrix $\boldsymbol{F}$ is obtained by (4).

$$\boldsymbol{f}^i = \left[f_1^i, f_2^i, \dots, f_{180}^i\right]^T \tag{3}$$

$$\boldsymbol{F} = [\boldsymbol{f}^1, \boldsymbol{f}^2, \dots, \boldsymbol{f}^t, \dots, \boldsymbol{f}^{51234}] \tag{4}$$

The 180×51234 feature matrix is used for learning a text classifier in cascade Adaboost model. A row of the feature matrix records feature responses of a certain block pattern and a certain feature map on all training samples. In the process of Adaboost learning, weak classifier is defined as $\langle r, T_r, \rho \rangle$. The three parameters denote the r-th row of feature matrix $(1 \le r \le 180)$, a threshold of the r-th row T_r, and polarity of the threshold $\rho \in \{-1,1\}$. In each row r, linearly spaced threshold values are sampled in the domain of its feature values by (5).

$$T_r \in \left\{ T \middle| T = f_r^{min} + \frac{1}{N_T}(f_r^{max} - f_r^{min})t \right\} \tag{5}$$

where N_T represents the number of thresholds, f_r^{min} and f_r^{max} represent the minimum and maximum feature value of the r-th row, and t is an integer ranging from 1 to N_T. We set $N_T = 300$ in the learning process. Thus there are in total $180 \times 2 \times 300 = 108000$ weak classifiers. When a weak classifier $\langle r, \rho, T_r \rangle$ is applied on a sample with corresponding feature vector $\boldsymbol{f} = [f_1, \dots, f_r, \dots, f_{180}]^T$, if $\rho f_r \ge \rho T_r$, it is classified as positive samples, otherwise it is classified as negative samples.

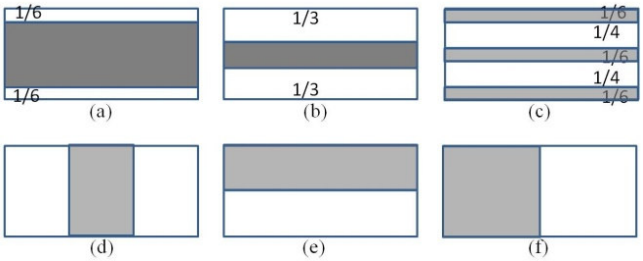

Fig. 6. Block patterns based on [1]. Features are obtained by the absolute value of sum (or mean) of pixel values in white regions minus sum (or mean) of pixel values in black regions.

Cascade Adaboost classifiers proved to be an effective machine learning algorithm in real-time face detection [18]. The training process is divided into several stages. In each stage, based on the feature matrix of all positive samples and the negative samples that are incorrectly classified in previous stages, Adaboost model [5] performs an iterative selection of weak classifiers. The selected weak classifiers are integrated into a strong classifier by weighted combination. The iteration of a stage stops when 99.5% of positive samples are correctly classified while 50% of negative samples are correctly classified by the current strong classifier. The strong classifiers from all stages are cascaded into the final classifier. When a testing image patch is input into the final classifier, it is classified as text patches if all the cascaded strong classifiers determine it is a positive sample, otherwise it is classified as a non-text patch.

3.4 Text Region Localization

Text localization is then performed on the camera captured image. Cascade Adaboost classifier cannot handle the whole image, so heuristic layout analysis is performed to extract candidate image patches prepared for text classification. Text information in the image usually appears in the form of text strings containing no less than three character members. Therefore adjacent character grouping [19] is used to calculate the image patches that possibly contain fragments of text strings. These fragments consist of three or more neighboring edge boundaries which have approximately equal heights and stay in horizontal alignment, as shown in Fig. 7. But not all the satisfied neighboring edge boundaries are text string fragments. Thus the classifier is applied to the image patches to determine whether they contain text or not. Finally, overlapped text patches are merged into a text region, which is the minimum rectangle area circumscribing the text patches. The text string fragments inside those patches are assembled into informative words.

Fig. 7. (a) Character 'e' have adjacent siblings 'p' on the left and 'n' on the right. (b) Adjacent characters are grouped together to obtain two fragments of text strings. (c) Candidate image patches after scaling and slicing, prepared for classification.

4 Text Recognition and Audio Output

Text recognition is performed by off-the-shelf OCR to output the informative words from the localized text regions. A text region labels the minimum rectangular area for the accommodation of characters inside it, so the border of the text region contacts the edge boundary of the text character. However, experiments show that OCR generates better performance if text regions are assigned proper margin areas and binarized to segment text characters from background. Thus each localized text region is enlarged by enhancing the height and width by 10 pixels respectively, and then we use Otsu' method [11] to perform binarization of text regions, where margin areas are always considered as background.

We evaluate two OCR engines, *Tesseract* and *Nuance OmniPage*, on the localized text regions. *OmniPage* shows better performance in most cases, but it is commercial software without open source codes. *Tesseract* is an open-source OCR engine that can be more conveniently integrated into our system.

The recognized text codes are recorded in script files. Then we use Microsoft Speech SDK to load these files and display the audio output of text information. Blind users can adjust speech rate, volume and tone according to their requirements.

5 Experiments

5.1 Datasets

Two datasets are used in our experiments. First, the ICDAR 2003 Robust Reading Dataset is used to evaluate the proposed localization algorithm separately. It contains 509 natural scene images in total. Most images contain indoor or outdoor text signage. The image resolutions range from 640×480 to 1600×1200. Since layout analysis based on adjacent character grouping can only handle text strings with three or more character members, we omit the images containing only ground truth text regions of less than 3 text characters. Thus 488 images are selected from this dataset as testing images to evaluate our localization algorithm.

To evaluate the whole system and develop a user friendly interface, we recruit 10 blind persons to build a dataset of reading text on hand-held objects. They wear a camera attached to a pair of sunglasses and capture the image of the objects in his/her hand, as shown in Fig. 8. The resolution of captured image is 960×720. There are 14 testing objects for each person, including grocery boxes, medicine bottles, books, etc. They are required to rotate each object several times to ensure that surfaces with text captions are captured. These objects are exposed to background outliers and illumination changes. We extract 116 captured images and label 312 text regions of main titles manually.

Fig. 8. Blind persons are capturing images of the object in their hands.

5.2 Results and Discussions

A localization algorithm is performed on the scene images of Robust Reading Dataset to calculate image regions containing text information. Fig. 9 and Fig. 10(a) depict some results of localized text regions, marked by cyan rectangle boxes. To analyze

the accuracy of the localized text regions, we compare them with ground truth text regions by the measures *precision*, *recall* and *f-measure*. For a pair of text regions, match sore is estimated by the ratio between the intersection area and the united mean area of the two regions. Each localized (ground truth) text region generates maximum match score from its best matched ground truth (localized) text region. *Precision* is the ratio between the total match score and the total number of localized regions. It estimates the false positive localized regions. *Recall* is the ratio between the total match score and the total number of ground truth regions. It estimates the missing text regions. *f*-measure combines *precision* and *recall* by harmonic sum. The evaluation results are calculated from average measures on all testing images, which are precision 0.69, recall 0.56, and *f*-measure 0.60. The results are comparable to previous algorithms as shown in Table I. Average processing time on original image resolution is 10.36s. To improve the computation speed, we downsample the testing images into lower resolutions while ensuring that the degradation does not significantly influence the performance. Both the width and the height of downsampled testing image do not exceed 920. Then we repeat the evaluation and obtain precision 0.68, recall 0.54, *f*-measure 0.58, and average process time 1.54s.

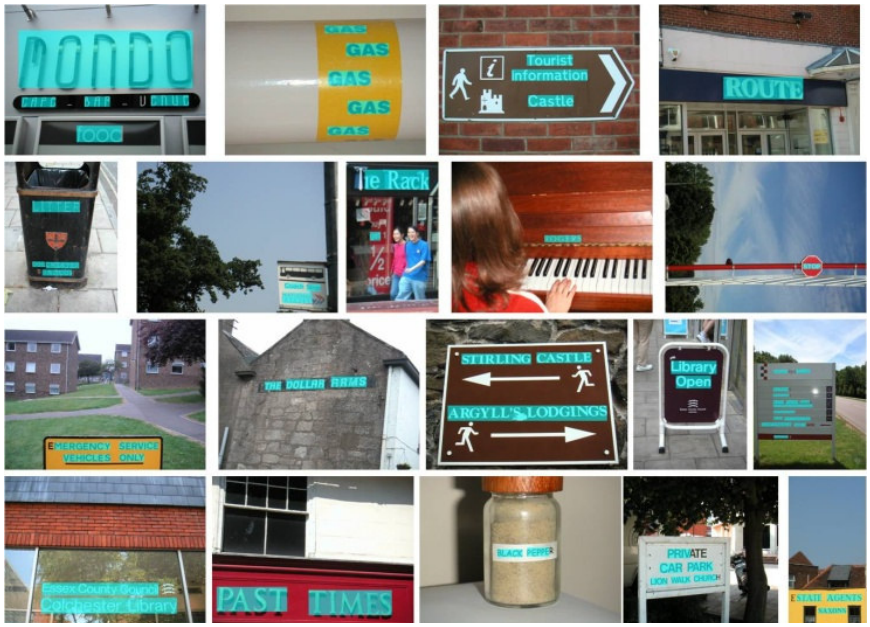

Fig. 9. Some example results of text localization on the robust reading dataset, and the localized text regions are marked in cyan.

To evaluate the proposed features of text based on stroke orientations and edge distributions, we can make a comparison with Alex Chen's algorithm [1, 8] because it applies similar block patterns and a similar learning model, but with different feature maps, which are generated from intensities, gradients and joint histograms of intensity

and gradient. The evaluation results of Chen's algorithm on the same dataset is precision 0.60, recall 0.60, and *f*-measure 0.58 (Table 1). This demonstrates that our proposed feature maps of stroke orientation and edge distribution give better performance on precision and *f*-measure.

Table 1. The performance comparison between our algorithm and the algorithms presented in [8] on Robust Reading Dataset.

Method	Precision	Recall	*f*	time/s
Ours	0.69	0.56	0.60	10.36
Ours(downsample)	0.68	0.54	0.58	1.54
HinnerkBecker	0.62	0.67	0.62	14.4
AlexChen	0.60	0.60	0.58	0.35
Ashida	0.55	0.46	0.50	8.7
HWDavid	0.44	0.46	0.45	0.3

Fig. 10. The top two rows present some results of text localization on the blind-captured dataset, where localized text regions are marked in cyan. The bottom rows show two groups of enlarged text regions, binarized text regions and word recognition results from top to down.

Further, our system is evaluated on the blind-captured dataset of object text. We define that a ground truth region is hit if its three-quarter is covered by localized regions. Experiments show that 225 of the 312 ground truth text regions are hit by our localization algorithm. By using the same evaluation measures as above experiments, we obtain precision 0.52, recall 0.62, and *f*-measure 0.52 on this dataset. The precision is much lower than that on Robust Reading Dataset. We infer that the images in blind-captured dataset of object text have lower resolutions and more compact distributions of text information. Then OCR is applied on the localized regions for character and word recognition rather than the whole images. Fig. 10 shows some examples of text localization and word recognition in the system. Recognition algorithm might not correctly and completely output the words inside localized regions. Additional spelling correction is required to output accurate text information. It takes 1.87 seconds on average in reading text from the normalized blind-captured images with resolution 640×480. In real applications, text extraction and device input/output can be processed in parallel, that is, speech output of recognized text while localization of text regions in the next image.

6 Conclusion

In this paper, we have developed a novel text localization algorithm and an assistive text reading prototype system for blind persons. Our system can extract text information from nearby text signage or object captions under complex backgrounds. Text localization and recognition are significant components of our system. To localize text, models of stroke orientation and edge distribution are proposed for extracting features of text. The corresponding feature maps estimate the global structural feature of text at every pixel. Block patterns are defined to project the proposed feature maps of an image patch into a feature vector. An Adaboost learning model is employed to train classifiers of text based on the feature vectors of training samples. To localize text in camera captured images, adjacent character grouping is performed to calculate candidates of text patches prepared for text classification. The Adaboost-based text classifier is applied to obtain the text regions. Off-the-shelf OCR is used to perform word recognition in the localized text regions and transform into audio output for blind users.

Our future work will focus on extending our localization algorithm to process text strings with less than 3 characters and to design more robust block patterns for text feature extraction. We will also extend our system to extract non-horizontal text strings. Furthermore, we will address the significant human interface issues associated with reading region selection by blind users.

ACKNOWLEDGEMENT

This work was supported in part by NIH Grant 1R21EY020990, NSF Grant IIS-0957016 and EFRI-1137172.

References

1. X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," In *CVPR*, Vol. 2, pp. II-366 – II-373, 2004.
2. X. Chen, J. Yang, J. Zhang and A. Waibel, "Automatic detection and recognition of signs from natural scenes," In *IEEE Transactions on image processing*, Vol. 13, No. 1, pp. 87-99, 2004.
3. D. Dakopoulos and N. G. Bourbakis, "Wearable obstacle avoidance electronic travel aids for blind: a survey," In *IEEE Transactions on systems, man, and cybernetics*, Vol. 40, No. 1, pp. 25-35, 2010
4. B. Epshtein, E. Ofek and Y. Wexler, "Detecting text in natural scenes with stroke width transform," In *CVPR*, pp. 2963-2970, 2010.
5. Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," In *Int. Conf. on Machine Learning*, pp.148–156, 1996.
6. K. I. Kim, K. Jung, and J. 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*, 2003.
7. S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Joshi, "Text Extraction and Document Image Segmentation Using Matched Wavelets and MRF Model," *IEEE Trans on Image Processing*, Vol. 16, No. 8, pp. 2117-2128, 2007.
8. S. M. Lucas, "ICDAR 2005 text locating competition results," *Proceedings of the ICDAR*, Vol. 1, pp 80–84, 2005.
9. L. Ma, C. Wang, B. Xiao, "Text detection in natural images based on multi-scale edge detection and classification," In *the Int. Congress on Image and Signal Processing (CISP)*, 2010.
10. N. Nikolaou and N. Papamarkos, "Color Reduction for Complex Document Images," *International Journal of Imaging Systems and Technology*, Vol.19, pp.14-26, 2009.
11. N. Otsu, "A threshold selection method from gray-level histograms," In *IEEE Tran.s on system, man and cybernetics*, pp. 62-66, 1979.
12. T. Phan, P. Shivakumara and C. L. Tan, "A Laplacian Method for Video Text Detection," In *Proceedings of ICDAR*, pp.66-70, 2009.
13. L. Ran, S. Helal, and S. Moore, "Drishti: an integrated indoor/outdoor blind navigation system and service," In *Pervasive computing and communications,* pp. 23-40, 2004.
14. S. Resnikoff, D. Pascolini, D. Etya'ale, I. Kocur, R. Pararajasegaram, G. P. Pokharel, et al, "Global data on visual impairment in the year 2002." In *Bulletin of the World Health Organization,* 844- 851, 2004.
15. H. Schneiderman and T. Kanade, "A statistical method for 3D object dection applied to faces and cars," In *CVPR* 2000.
16. M. Shi, Y. Fujisawab, T. Wakabayashia and F. Kimura, "Handwritten numeral recognition using gradient and curvature of gray scale image," In *Pattern Recognition*, Vol. 35, Issue 10, pp. 2051-2059, 2002.
17. P. Shivakumara, T. Phan, and C. L. Tan, "A gradient difference based technique for video text detection," *The 10th ICDAR*, pp.66-70, 2009.
18. P. Viola and M. J. Jones, "Robust real-time face detection," In *IJCV* 57(2), 137–154, 2004.
19. C. Yi and Y. Tian, "Text string detection from natural scenes by structure based partition and grouping," In *IEEE Transactions on Image Processing*, 2011.
20. J. Zhang and R. Kasturi, "Extraction of Text Objects in Video Documents: Recent Progress," In *IAPR Workshop on Document Analysis Systems*, 2008.
21. ICDAR 2011 Robust Reading Competition: http://robustreading. opendfki.de/