Portable Camera-Based Assistive Text and Product Label Reading From Hand-Held Objects for Blind Persons

Chucai Yi, *Student Member, IEEE*, Yingli Tian, *Senior Member, IEEE*, and Aries Arditi

***Abstract*—We propose a camera-based assistive text reading framework to help blind persons read text labels and product packaging from hand-held objects in their daily lives. To isolate the object from cluttered backgrounds or other surrounding objects in the camera view, we first propose an efficient and effective motion-based method to define a region of interest (ROI) in the video by asking the user to shake the object. This method extracts moving object region by a mixture-of-Gaussians-based background subtraction method. In the extracted ROI, text localization and recognition are conducted to acquire text information. To automatically localize the text regions from the object ROI, we propose 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 text regions are then binarized and recognized by off-the-shelf optical character recognition software. The recognized text codes are output to blind users in speech. Performance of the proposed text localization algorithm is quantitatively evaluated on ICDAR-2003 and ICDAR-2011 Robust Reading Datasets. Experimental results demonstrate that our algorithm achieves the state of the arts. The proof-of-concept prototype is also evaluated on a dataset collected using ten blind persons to evaluate the effectiveness of the system's hardware. We explore user interface issues and assess robustness of the algorithm in extracting and reading text from different objects with complex backgrounds.**

***Index Terms*—Assistive devices, blindness, distribution of edge pixels, hand-held objects, optical character recognition (OCR), stroke orientation, text reading, text region localization.**

I. Introduction

OF the 314 million visually impaired people worldwide, 45 million are blind [1]. Even in a developed country like the U.S., the 2008 National Health Interview Survey reported that an estimated 25.2 million adult Americans (over 8%) are blind or visually impaired [2]. This number is increasing rapidly as the baby boomer generation ages. Recent developments in computer vision, digital cameras, and portable computers make it feasible to assist these individuals by developing camera-based products that combine computer vision technology with other existing commercial products such optical character recognition (OCR) systems.

Reading is obviously essential in today's society. Printed text is everywhere in the form of reports, receipts, bank statements, restaurant menus, classroom handouts, product packages, instructions on medicine bottles, etc. And while optical aids, video magnifiers, and screen readers can help blind users and those with low vision to access documents, there are few devices that can provide good access to common hand-held objects such as product packages, and objects printed with text such as prescription medication bottles. The ability of people who are blind or have significant visual impairments to read printed labels and product packages will enhance independent living and foster economic and social self-sufficiency.

Today, there are already a few systems that have some promise for portable use, but they cannot handle product labeling. For example, portable bar code readers designed to help blind people identify different products in an extensive product database can enable users who are blind to access information about these products [22] through speech and braille. But a big limitation is that it is very hard for blind users to find the position of the bar code and to correctly point the bar code reader at the bar code. Some reading-assistive systems such as pen scanners might be employed in these and similar situations. Such systems integrate OCR software to offer the function of scanning and recognition of text and some have integrated voice output. However, these systems are generally designed for and perform best with document images with simple backgrounds, standard fonts, a small range of font sizes, and well-organized characters rather than commercial product boxes with multiple decorative patterns. Most state-of-the-art OCR software cannot directly handle scene images with complex backgrounds.

A number of portable reading assistants have been designed specifically for the visually impaired [12], [16], [20], [21], [23], [24], [27], [31], [32]. *KReader Mobile* runs on a cell phone and allows the user to read mail, receipts, fliers, and many other documents [12]. However, the document to be read must be nearly flat, placed on a clear, dark surface (i.e., a noncluttered background), and contain mostly text. Furthermore, *KReader*

Manuscript received March 23, 2012; revised May 23, 2012, August 17, 2012, November 7, 2012, and January 26, 2013; accepted April 23, 2013. Date of publication May 17, 2013; date of current version April 11, 2014. Recommended by Technical Editor J. Wang. This work was supported in part by the National Institutes of Health under Grant 1R21EY020990, in part by the National Science Foundation under Grant IIS-0957016 and Grant EFRI-1137172, in part by the Federal Highway Administration under Grant DTFH61-12-H-00002, and in part by Microsoft Research.

C. Yi is with the Graduate Center, The City University of New York, New York, NY 10016 USA (e-mail: cyi@gc.cuny.edu).

Y. Tian was with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA. She is now with City College, The City University of New York, New York, NY 10031 USA (e-mail: ytian@ccny.cuny.edu).

A. Arditi was with Lighthouse International, New York, NY 10022 USA, and the IBM T. J. Watson Research Center, New York, NY 10504-1722 USA. He is now with Visibility Metrics LLC, Chappaqua, NY 10514 USA (e-mail: arditi@visibilitymetrics.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2013.2261083

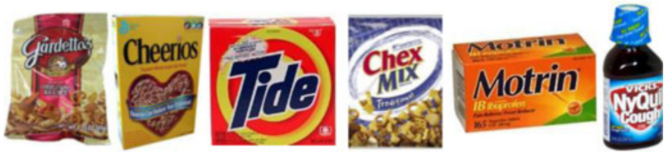


Fig. 1. Examples of printed text from hand-held objects with multiple colors, complex backgrounds, or nonflat surfaces.

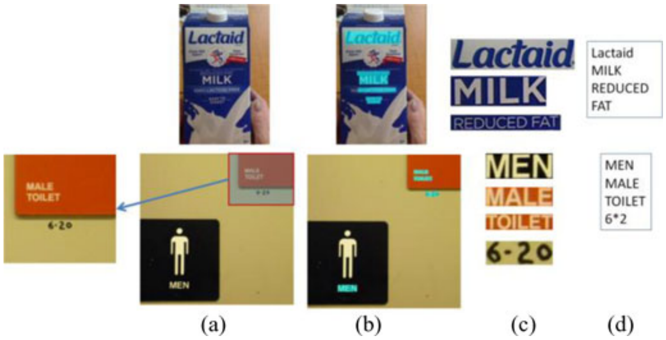


Fig. 2. Two examples of text localization and recognition from camera-captured images. (Top) Milk box. (Bottom) Men bathroom signage. (a) camera-captured images. (b) Localized text regions (marked in blue). (c) Text regions cropped from image. (d) Text codes recognized by OCR. Text at the top-right corner of bottom image is shown in a magnified callout.

Mobile accurately reads black print on a white background, but has problems recognizing colored text or text on a colored background. It cannot read text with complex backgrounds, text printed on cylinders with warped or incomplete images (such as soup cans or medicine bottles). Furthermore, these systems require a blind user to manually localize areas of interest and text regions on the objects in most cases.

Although a number of reading assistants have been designed specifically for the visually impaired, to our knowledge, no existing reading assistant can read text from the kinds of challenging patterns and backgrounds found on many everyday commercial products. As shown in Fig. 1, such text information can appear in multiple scales, fonts, colors, and orientations. To assist blind persons to read text from these kinds of hand-held objects, we have conceived of a camera-based assistive text reading framework to track the object of interest within the camera view and extract print text information from the object. Our proposed algorithm can effectively handle complex background and multiple patterns, and extract text information from both hand-held objects and nearby signage, as shown in Fig. 2.

In assistive reading systems for blind persons, it is very challenging for users to position the object of interest within the center of the camera's view. As of now, there are still no acceptable solutions. We approach the problem in stages.

To make sure the hand-held object appears in the camera view, we use a camera with sufficiently wide angle to accommodate users with only approximate aim. This may often result in other text objects appearing in the camera's view (for example, while shopping at a supermarket). To extract the hand-held object from the camera image, we develop a motion-based method to obtain a region of interest (ROI) of the object. Then, we perform text recognition only in this ROI.

It is a challenging problem to automatically localize objects and text ROIs from captured images with complex backgrounds, because text in captured images is most likely surrounded by various background outlier "noise," and text characters usually appear in multiple scales, fonts, and colors. For the text orientations, this paper assumes that text strings in scene images keep approximately horizontal alignment. Many algorithms have been developed for localization of text regions in scene images. We divide them into two categories: rule-based and learning-based.

Rule-based algorithms apply pixel-level image processing to extract text information from predefined text layouts such as character size, aspect ratio, edge density, character structure, color uniformity of text string, etc. Phan *et al.* [19] analyzed edge pixel density with the Laplacian operator and employed maximum gradient differences to identify text regions. Shivakumara *et al.* [26] used gradient difference maps and performed global binarization to obtain text regions. Epshtein *et al.* [7] designed stroke width transforms to localize text characters. Nikolaou and Papamarkos [17] applied color reduction to extract text in uniform colors. In [5], color-based text segmentation is performed through a Gaussian mixture model for calculating a confidence value for text regions. This type of algorithm tries to define a universal feature descriptor of text.

Learning-based algorithms, on the other hand, model text structure and extract representative text features to build text classifiers. Chen and Yuille [4] presented five types of Haar-based block patterns to train text classifiers in an Adaboost learning model. Kim *et al.* [11] considered text as a specific texture and analyzed the textural features of characters by a support vector machine (SVM) model. Kumar *et al.* [13] used globally matched wavelet filter responses of text structure as features. Ma *et al.* [15] performed classification of text edges by using histograms of oriented gradients and local binary patterns as local features on the SVM model. Shi *et al.* [25] employed gradient and curvature features to model the grayscale curve for handwritten numeral recognition under a Bayesian discriminant function. In our research group, we have previously developed rule-based algorithms to extract text from scene images [33]–[35]. A survey paper about computer-vision-based assistive technologies to help people with visual impairments can be found in [16].

In solving the task at hand, to extract text information from complex backgrounds with multiple and variable text patterns, we here propose a text localization algorithm that combines rule-based layout analysis and learning-based text classifier training, which define novel feature maps based on stroke orientations and edge distributions. These, in turn, generate representative and discriminative text features to distinguish text characters from background outliers.

II. Framework and Algorithm Overview

This paper presents a prototype system of assistive text reading. As illustrated in Fig. 3, the system framework consists of three functional components: scene capture, data processing, and audio output. The scene capture component collects scenes

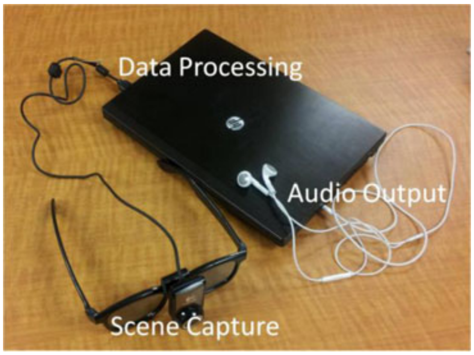


Fig. 3. Snapshot of our demo system, including three functional components for scene capture, data processing, and audio output.

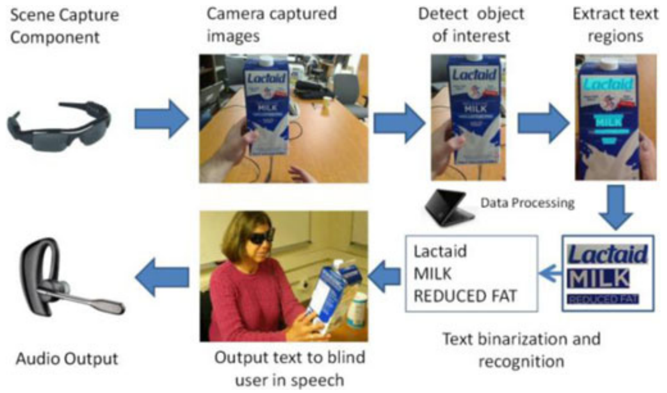


Fig. 4. Flowchart of the proposed framework to read text from hand-held objects for blind users.

containing objects of interest in the form of images or video. In our prototype, it corresponds to a camera attached to a pair of sunglasses. The data processing component is used for deploying our proposed algorithms, including 1) object-of-interest detection to selectively extract the image of the object held by the blind user from the cluttered background or other neutral objects in the camera view; and 2) text localization to obtain image regions containing text, and text recognition to transform image-based text information into readable codes. We use a min-laptop as the processing device in our current prototype system. The audio output component is to inform the blind user of recognized text codes. A Bluetooth earpiece with minimicrophone is employed for speech output.

This simple hardware configuration ensures the portability of the assistive text reading system. Fig. 4 depicts a work flowchart of the prototype system.

A frame sequence V is captured by a camera worn by blind users, containing their hand-held objects and cluttered background. To extract text information from the objects, motion-based object detection is first applied to determine the user's object of interest S by shaking the object while recording video

$$S = \frac{1}{|V|} \sum_i \mathcal{R}\left(V_i, B\right) \tag{1}$$

where V_i denotes the ith frame in the captured sequence, $|V|$ denotes the number of frames, B denotes the estimated background from motion-based object detection, and $\mathcal{R}$ represents the calculated foreground object at each frame. The object of interest is localized by the average of foreground masks (see details in Section III).

Next, our novel proposed text localization algorithm is applied to the object of interest to extract text regions. At first, candidate text regions are generated by layout analysis of color uniformity and horizontal alignment

$$X^C = \text{argmax}_{s \in S} L\left(s\right) \tag{2}$$

where $L\left(\cdot\right)$ denotes the suitability responses of text layout and X^C denotes the candidate text regions from object of interest S. Then, a text classifier is generated from a Cascade-Adaboost learning model, by using stroke orientations and edge distributions of text characters as features (see details in Section IV).

$$X = H\left[X^C\right] = H\left[\text{argmax}_{s \in S} L\left(s\right)\right] \tag{3}$$

where H denotes the Cascade-Adaboost classifier and X denotes the localized text regions.

After text region localization, off-the-shelf OCR is employed to perform text recognition in the localized text regions. The recognized words are transformed into speech for blind users.

Our main contributions embodied in this prototype system are: 1) a novel motion-based algorithm to solve the aiming problem for blind users by their simply shaking the object of interest for a brief period; 2) a novel algorithm of automatic text localization to extract text regions from complex background and multiple text patterns; and 3) a portable camera-based assistive framework to aid blind persons reading text from hand-held objects. Algorithms of the proposed system are evaluated over images captured by blind users using the described techniques.

III. Object Region Detection

To ensure that the hand-held object appears in the camera view, we employ a camera with a reasonably wide angle in our prototype system (since the blind user may not aim accurately). However, this may result in some other extraneous but perhaps text-like objects appearing in the camera view for example, when a user is shopping at a supermarket). To extract the hand-held object of interest from other objects in the camera view, we ask users to shake the hand-held objects containing the text they wish to identify and then employ a motion-based method to localize the objects from cluttered background. Background subtraction (BGS) is a conventional and effective approach to detect moving objects for video surveillance systems with stationary cameras. To detect moving objects in a dynamic scene, many adaptive BGS techniques have been developed.

Stauffer and Grimson [28] modeled each pixel as a mixture of Gaussians and used an approximation to update the model. A mixture of K Gaussians is applied for BGS, where K is from 3 to 5. In this process, the prior weights of K Gaussians are online adjusted based on frame variations. Since background imagery is nearly constant in all frames, a Gaussian always compatible with its subsequent frame pixel distribution is more likely to be the background model. This Gaussian-mixture-model-based method is robust to slow lighting changes, but cannot

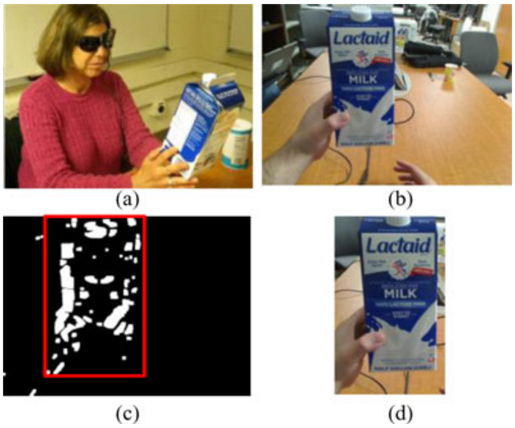


Fig. 5. Localizing the image region of the hand-held object of interest. (a) Capturing images by a camera mounted on a pair of sunglasses. (b) Example of a captured image. (c) Detected moving areas in the image while the user shaking the object (region inside the bounding box). (d) Detected region of the hand-held object for further processing of text recognition.

handle complex foregrounds and quick lighting changes. Tian *et al.* [29] further improved the multiple Gaussian-mixture-based BGS method to better define foreground while remove background objects. First, texture information is employed to remove false positive foreground areas. These areas should be background but are often determined as foreground because of sudden lighting changes. A texture similarity measure is defined to evaluate whether the detected foreground motion is caused by lighting change or moving object. Second, in addition to quick lighting changes, BGS is also influenced by shadows. Many systems use color information to remove the shadow, but this does not work on grayscale videos. To solve this problem, the normalized cross correlation of the intensities is used for shadow removal. The grayscale distribution of a shadow region is very similar to that of the corresponding background region, except is a little darker. Thus, for a pixel in BGS-modeled foreground areas, we calculate the NCC between the current frame and the background frame to evaluate their similarity and remove the influence of shadow.

As shown in Fig. 5, while capturing images of the hand-held object, the blind user first holds the object still, and then lightly shakes the object for 1 or 2 s. Here, we apply the efficient multiple Gaussian-mixture-based BGS method to detect the object region while blind user shakes it. More details of the algorithm can be found in [29]. Once the object of interest is extracted from the camera image, the system is ready to apply our automatic text extraction algorithm.

IV. Automatic Text Extraction

As shown in Fig. 6, we design a learning-based algorithm for automatic localization of text regions in image.

In order to handle complex backgrounds, we propose two novel feature maps to extracts text features based on stroke orientations and edge distributions, respectively. Here, stroke is defined as a uniform region with bounded width and significant extent. These feature maps are combined to build an Adaboost-based text classifier.

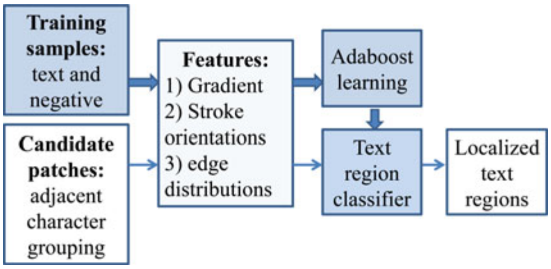

Training samples: text and negative
Candidate patches: adjacent character grouping
Features: 1) Gradient 2) Stroke orientations 3) edge distributions
Adaboost learning
Text region classifier
Localized text regions

Fig. 6. Diagram of the proposed Adaboost-learning-based text region localization algorithm by using stroke orientations and edge distributions.

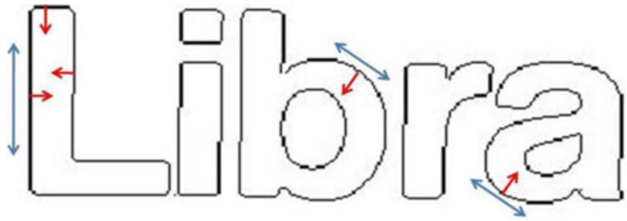


Fig. 7. Sample of text strokes showing relationships between stroke orientations and gradient orientations at pixels of stroke boundaries. Blue arrows denote the stroke orientations at the sections and red arrows denote the gradient orientations at pixels of stroke boundaries.

A. *Text Stroke Orientation*

Text characters consist of strokes with constant or variable orientation as the basic structure. Here, we propose a new type of feature, stroke orientation, 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. 7. To model the text structure by stroke orientations, we propose a new operator to map a gradient feature of strokes to each pixel. It extends the local structure of a stroke boundary into its neighborhood by gradient of orientations. We use it to develop 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 two gradient maps G_x and G_y, respectively. The synthesized gradient map is calculated as $G = \left(G_x^2 + G_y^2\right)^{1/2}$. The Canny edge detector is applied on I to calculate its binary edge map E. For a pixel p_0, we certify whether it is close to a character stroke by setting a circular range as $R(p_0) = \{p | d(p, p_0) \leq r\}$, where $d(.)$ denotes Euclidean distance, and $r = 36$ is the threshold of the circular range to search for edge pixels. We set this threshold because the text patches in our experiments are all normalized into height 48 pixels and width 96 pixels, and the stroke width of text characters in these normalized patches mostly does not exceed 36. If the distance is greater than 36, pixel p_0 would be located at background region far away from text character. In the range, we select the edge pixel p_e with the minimum Euclidean distance from p_0. Then, the pixel p_0 is labeled with gradient orientation at pixel p_e from gradient maps by

$$\begin{aligned} p_e &= \underset{p \in P}{\operatorname{argmin}}\, d(p, p_0) \\ S(p_0) &= \Upsilon\left(\arctan\left(G_y(p_e), G_x(p_e)\right)\right) \end{aligned} \tag{4}$$

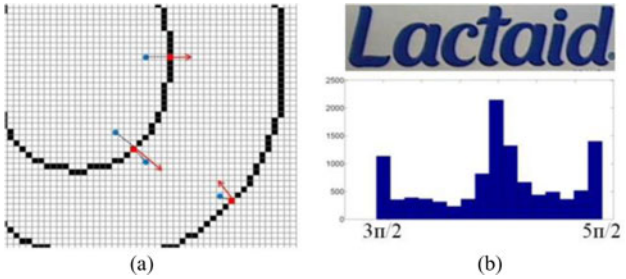


Fig. 8. (a) Example of stroke orientation label. The pixels denoted by blue points are assigned the gradient orientations (red arrows) at their nearest edge pixels, denoted by the red points. (b) 210 ×54 text patch and its 16-bin histogram of quantized stroke orientations.

where $P = \{p | p \in R(p_0), p \text{ is edge pixel}\}$. The stroke orientation calculated from arctan will be in the range $(-\pi/2, \pi/2]$. To distinguish the pixels labeled with stroke orientation 0 and the unlabeled pixels also with value 0, Υ shifts the stroke orientations one period forward into the range $(3\pi/2, 5\pi/2]$, which removes the value 0 from the range of stroke orientations. A stroke orientation map $S(p)$ is output by assigning each pixel the gradient orientation at its nearest edge pixel, as shown in Fig. 8(a). The pixel values in stroke orientation map are then quantized into an N bin histogram in the range $(3\pi/2, 5\pi/2]$ [see Fig. 8(b)]. In the feature extraction stage, strokes with identical or similar orientations are identified to describe the structure of text from one perspective. In the N bin histogram, we group the pixels at every b consecutive bins together to generate a multilayer stroke orientation map, where strokes in different orientations are separated into different layers. Without considering the cyclic shifts of the bins, there are a total of $N - b + 1$ layers.

The range of stroke orientations $(3\pi/2,\ 5\pi/2]$ is quantized into $N = 16$ bins, so each bin corresponds to $\pi/16 = 11.25^\circ$ and $b = 3$ consecutive bins will cover a range of 33.75°. This span value is compatible with most character strokes in scene images, because the stroke orientations are always vertical, horizontal, or approximate $30-40^\circ$ such as "W," "X," and arc components of "P," "D," etc. Since b is set to be 3 and N is set to be 16, 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 such stroke orientation maps.

B. Distribution of Edge Pixels

In an edge map, text characters appear in the form of stroke boundaries. The distribution of edge pixels in stroke boundaries also describes the characteristic structure of text. The most commonly used feature [34], [36] is edge density of text region. But the 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 in its located column. Then, a 3×3 smooth operator w_n is applied to obtain the edge distribution feature map, as (5). In this feature map, pixel value reflects edge density in its located region, and the smoothed map better represents the discriminative inner structure of text characters

$$D(x, y) = \sum_n w_n \cdot N_R(y_n) \cdot N_C(x_n) \qquad (5)$$

where (x_n, y_n) is neighboring pixel of (x, y) and $w_n = 1/9$ denotes the weight value.

C. Adaboost Learning of Text Features

Based on the feature maps of gradient, stroke orientation, and edge distribution, a text classifier is trained from an Adaboost learning model. Image patches with fixed size (height 48 pixels, width 96 pixels) are collected and resized from images taken from the ICDAR-2011 robust reading competition [10] 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 aspect ratio of width w to height h. To train a robust text classifier, we ensure that most positive training samples contain two to four text characters. We build a relationship between the width-to-height aspect ratio and the number of characters of ground truth text regions. It shows that the ground truth regions with two to four text characters have width-to-height ratios between 0.8 and 2.5, while the ones lower than 0.8 mostly have less than two characters and the ones higher than 2.5 mostly have more than four characters. Therefore, if the ratio is $w/h < 0.8$ with too few characters, the region is discarded. If the ratio $w/h \geq 2.5$ corresponding to more than four text characters, we slice this ground truth region into overlapped patches with width-to-height ratio 2:1. If the ratio w/h falls in $[0.8, 2.5)$, we keep it unsliced and scale it to width-to-height ratio 2:1. Then, the samples are normalized into width 96 and height 48 pixels for training. The negative training samples are generated by extracting the image regions containing edge boundaries of nontext objects. These regions also have width-to-height ratio 2:1, and we similarly scale them into width 96 and height 48. In this training set, there are a total of 15 301 positive samples, each containing several text characters, and 35 933 negative samples without containing any text information for learning features of background outliers. Some training examples are shown in Fig. 9.

To train the classifier, we extract 3 gradient maps, 14 stroke orientation maps, and 1 edge distribution map for each training sample. We apply six block patterns [4] on these feature maps of training samples. As shown in Fig. 10, 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

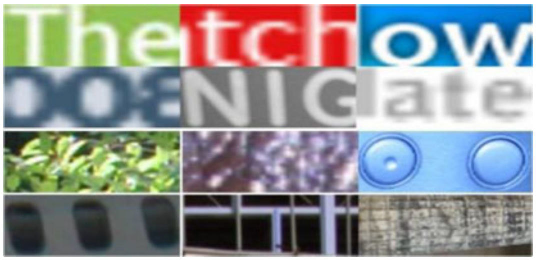


Fig. 9. Examples of training samples with width-to-height ratio 2:1. The first two rows are positive samples and the other two rows are negative samples.

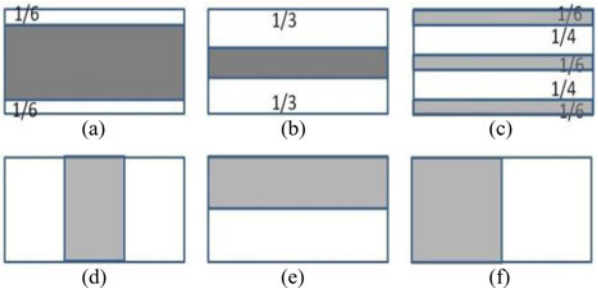


Fig. 10. Block patterns based on [4]. Features are obtained by the absolute value of mean pixel values in white regions minus those in black regions.

regions. For the block patterns with more than two subregions [see Fig. 10(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 six block patterns and two metrics from each feature map. The "integral image" algorithm is used in these calculations [30]. 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 (6). We compute feature vectors for all the 51 234 samples in the training set. By using feature vector $\boldsymbol{f}^i$ of the ith sample as the ith column**,** a feature matrix $\boldsymbol{F}$ is obtained by (7)

$$\boldsymbol{f}^i = \left[f_1^i, f_2^i, \ldots, f_{180}^i\right]^T \tag{6}$$

$$\boldsymbol{F} = \left[\boldsymbol{f}^1, \boldsymbol{f}^2, \ldots, \boldsymbol{f}^t, \ldots, \boldsymbol{f}^{51234}\right]. \tag{7}$$

The $180 \times 51\,234$ feature matrix is used for learning a text classifier in a 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 r, T_r, ρ. The three parameters denote the rth row of feature matrix $(1 \le r \le 180)$, a threshold of the rth 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

$$T_r \in \left\{T | T = f_r^{\min} + \frac{1}{N_T}\left(f_r^{\max} - f_r^{\min}\right) t\right\} \tag{8}$$

where N_T represents the number of thresholds, $f_r^{\min}$ and $f_r^{\max}$ represent the minimum and maximum feature value of the rth row, respectively, 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 $180 \times 2 \times 300 = 108\,000$ weak classifiers denoted as $\mathcal{H}$. When a weak classifier r, ρ, T_r is applied to a sample with corresponding feature vector $\boldsymbol{f} = \left[f_1, \ldots, f_r, \ldots, f_{180}\right]^T$, if $\rho f_r \ge \rho T_r$, it is classified as a positive sample; otherwise, it is classified as a negative sample.

The Cascade-Adaboost classifier has proved to be an effective machine learning algorithm in real-time face detection [30]. The training process is divided into several stages. In each stage, a stage-specific Adaboost classifier is learned from a training set, which consists of all positive samples and the negative samples incorrectly classified by previous Adaboost classifiers at this stage. We refer to this as a stage-Adaboost classifier in the following paragraphs.

The learning process based on the Adaboost model [8] at each stage is as follows: 1) The set of m samples $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ is given, where $x_i \in X$ denotes feature vector and $y_i \in \{-1, 1\}$ denotes ground truth. Each sample i is assigned a weight D_i, which is initialized to be $1/m$. 2) In the tth iteration, we select the optimized weak classifier h_t from the set of weak classifiers $\mathcal{H}$, such that $h_t = \text{argmin}_{h \in \mathcal{H}} \sum_{i=1}^m D_i y_i h(x_i)$, and calculate $\varepsilon_t = \sum_{i=1}^m D(i) \cdot (y_i \ne h(x_i))$ and $\alpha_t = 0.5 \ln((1 - \varepsilon_t)/\varepsilon_t)$. 3) Update the sample weights by $D_i := D_i \exp(-y_i h(x_i))$. 4) Start the next iteration from step (2) until all the samples are correctly classified or the maximum number of iterations is reached. 5) The optimized weak classifiers are combined into a stage-Adaboost classifier as $H(x) = \sum_t \alpha_t h_t(x)$.

In the end, all the stage-Adaboost classifiers are cascaded into the final Cascade-Adaboost classifier. When a test image patch is input into the final classifier, it is classified as a text patch if all the cascaded stage-Adaboost classifiers determine it is a positive sample, and otherwise, it is classified as a nontext patch. In the learning process, each stage-Adaboost classifier ensures that 99.5% of positive samples are correctly classified, while 50% of negative samples are correctly classified. Thus, a testing sample with positive ground truth will have a $(0.995)^T$ chance of correct classification, where T represents the total number of stage-Adaboost classifiers.

D. *Text Region Localization*

Text localization is then performed on the camera-based image. The Cascade-Adaboost classifier confirms the existence of text information in an image patch but 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 horizontal text strings containing no less than three character members. Therefore, adjacent character grouping [33] is used to calculate the image patches that contain fragments of text strings. These fragments consist of three or more neighboring edge boundaries that have approximately equal heights and stay in horizontal alignment, as shown in Fig. 11. But not all the satisfied neighboring edge boundaries are text string fragments. Thus, the classifier is applied to the image patches to determine

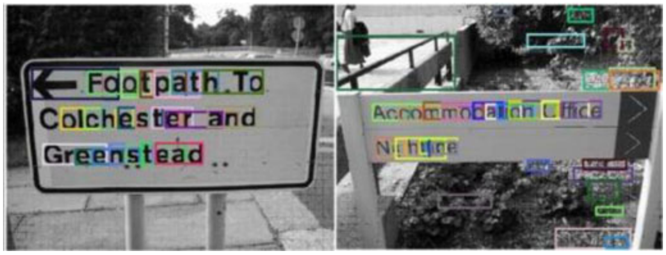


Fig. 11. Adjacent characters are grouped to obtain fragments of text strings, where each fragment is marked by a colored rectangle. The extracted image patches will be processed and input into text classifier.

whether they contain text or not. Finally, overlapped text patches are merged into the text region, which is the minimum rectangle area circumscribing the text patches. The text string fragments inside those patches are assembled into informative words.

V. Text Recognition and Audio Output

Text recognition is performed by off-the-shelf OCR prior to output of 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, our experiments show that OCR generates better performance if text regions are first 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's method [18] to perform binarization of text regions, where margin areas are always considered as background. We test both open- and closed-source solutions that allow the final stage of conversion to letter codes (e.g. OmniPage, Tesseract, ABBYReader).

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

VI. Experiments

A. Datasets

Two datasets are used to evaluate our algorithm. First, the ICDAR Robust Reading Dataset [10], [14] is used to evaluate the proposed text localization algorithm. The ICDAR-2003 dataset 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 three text characters. Thus, 488 images are selected from this dataset as testing images to evaluate our localization algorithm.

To further understand the performance of the prototype system and develop a user-friendly interface, following Human Subjects Institutional Review Board approval, we recruited ten blind persons to collect a dataset of reading text on hand-held objects. The hardware of the prototype system includes a Logitech web camera with autofocus, which is secured to the nose bridge of a pair of sunglasses. The camera is connected to an HP mini laptop by a USB connection. The laptop performs the processing and provides audio output. In order to avoid serious blocking or aural distraction, we would choose a wireless "open" style Bluetooth earpiece for presenting detection results as speech outputs to the blind travelers in a full prototype implementation.

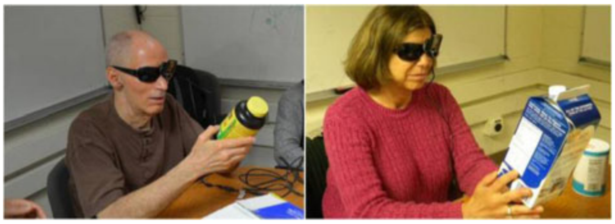


Fig. 12. Examples of blind persons capturing images of the object in their hands.

The blind user wore the camera/sunglasses to capture the image of the objects in his/her hand, as illustrated in Fig. 12. The resolution of the captured image is 960 × 720. There were 14 testing objects for each person, including grocery boxes, medicine bottles, books, etc. They were required keep their head (where the camera is fixed) stationary for a few seconds and subsequently shake the object for an additional couple of seconds to detect the region of object of interest. Each object was then rotated by the user several times to ensure that surfaces with text captions are exposed and captured. We manually extracted 116 captured images and labeled 312 text regions of main titles.

B. Evaluations of Text Region Localization

Text classification based on the Cascade-Adaboost classifier plays an important role in text region localization. To evaluate the effectiveness of the text classifier, we first performed a group of experiments on the dataset of sample patches, in which the patches containing text are positive samples and those without text are negative samples. These patches are cropped from natural scene images in ICDAR-2003 and ICDAR-2011 Robust Reading Datasets.

Each patch was assigned a prediction score by the text classifier; a higher score indicates a higher probability of text information. We define the true positive rate as the ratio of correct positive predictions to the total number of positive samples. Similarly, the false positive rate is the ratio of correct positive predictions to the total number of positive predictions. Fig. 13 plots the variation of true positive against false positive rates. This curve indicates that our text classifier is biased toward negative (i.e., no text) responses because the false positive rate stays near zero until the true positive rate approximately rises to 0.7. This characteristic is compatible with the design of our blind-assistive framework, in which it is useful to filter out extraneous background outliers and keep a conservative standard for what constitutes text.

Next, the text region localization algorithm was performed on the scene images of ICDAR-2003 Robust Reading Dataset

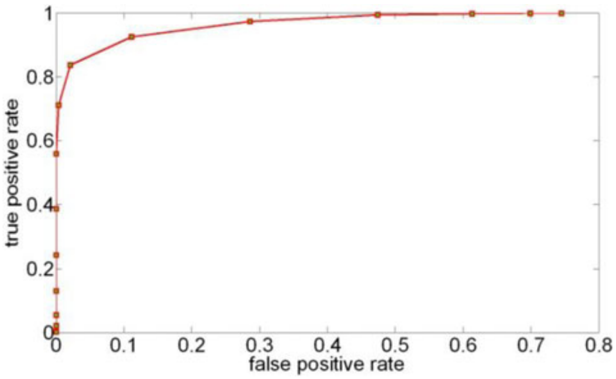


Fig. 13. Curve of classification performance, where horizontal axis denotes false positive rate and vertical axis denotes true positive rate.

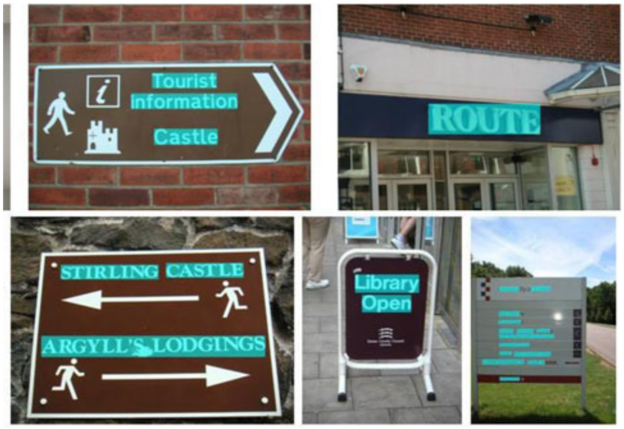


Fig. 14. Some example results of text localization on the ICDAR-2003 robust reading dataset, and the localized text regions are marked in blue. It shows that our algorithm can localize multiple text labels in indoor and outdoor environments.

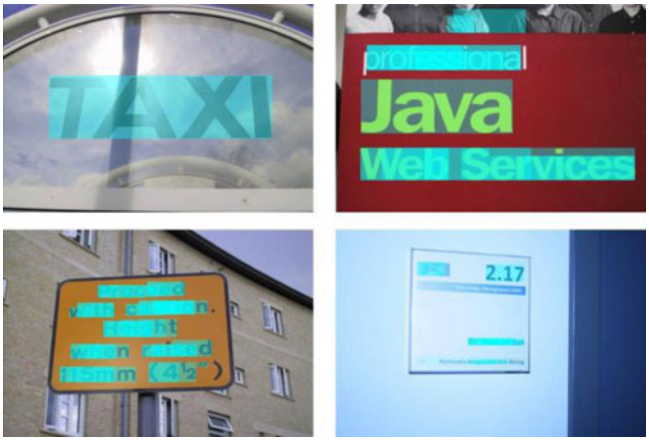


Fig. 15. Some example results of text localization on the ICDAR-2011 robust reading dataset, and the localized text regions are marked in blue. Our algorithm can localize multiple text labels in indoor and outdoor environments.

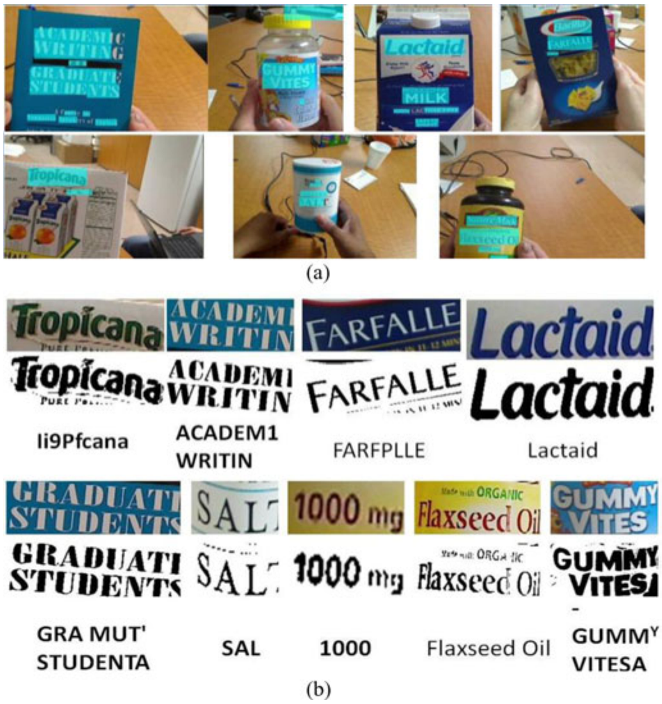


Fig. 16. (a) Some results of text localization on the user-captured dataset, where localized text regions are marked in blue. (b) Two groups of enlarged text regions, binarized text regions, and word recognition results from top to down.

to identify image regions containing text information. Figs. 14, 15, and 16(a) depict some results showing the localization of text regions, marked by blue rectangular boxes. To analyze the accuracy of the localized text regions, we compare them with ground truth text regions and characterize the results with measures we call *precision*, *recall*, and *f-measure.* For a pair of text regions, match score is estimated by the ratio between the intersection area and the mean area of the union of the two regions. Each localized (ground truth) text region generates a maximum match score from its best matched ground truth (localized) text region. *Precision* is the ratio of total match score to 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. The f-measure combines *precision* and *recall* as a harmonic sum and is defined by (9), where α represents the relative weight between the two metrics. According to the standard evaluation methods in [14], we set $\alpha = 0.5$

$$f = 1 \Big/ \left(\frac{\alpha}{p} + \frac{(1-\alpha)}{r} \right). \tag{9}$$

The evaluation results are calculated from average measures on all testing images, which have precision 0.69, recall 0.56, and f-measure 0.60. To evaluate the proposed features of text based on stroke orientations and edge distributions, we can make a comparison with Alex Chen's algorithm [4], [14] 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 are precision 0.60, recall 0.60, and f-measure 0.58 (see Table I). Thus, our proposed feature maps of stroke orientation and edge distribution perform better on precision and f-measure.

Our proposed text localization algorithm participated in the ICDAR-2011 Robust Reading Competition, and we won the

TABLE I
PERFORMANCE COMPARISON BETWEEN OUR ALGORITHM AND THE ALGORITHMS PRESENTED IN [14] ON ROBUST READING DATASET. OUR (DS) INDICATES OUR METHOD APPLIED ON THE DOWNSAMPLED IMAGES

Method	***precision***	***recall***	***f-measure***
Our	**0.69**	**0.56**	**0.60**
HinnerkBeck	0.62	0.67	0.62
AlexChen	0.60	0.60	0.58
Ashida	0.55	0.46	0.50
HWDavid	0.44	0.46	0.45

TABLE II
RESULTS OF ICDAR-2011 ROBUST READING COMPETITION ON SCENE TEXT LOCALIZATION (%) [10]

Method	***precision***	***recall***	***f-measure***
Kim	62.47	82.98	71.28
Yi (Our)	**58.09**	**67.22**	**62.32**
TH-TextLoc	57.68	66.97	61.98
Neumann	52.54	68.93	59.63
TDM IACS	53.52	63.52	58.09
LIP6-Retin	50.07	62.97	55.78
KAIST AIPR	44.57	59.67	51.03

Our proposed framework won second place.

second place (see Table II). In this competition, the same evaluation measures are employed to evaluate and compare the submitted detection results. Some examples of localized text regions are presented in Fig. 15 using blue boxes. To improve the performance of blind-assistive technology applications, we adjusted the parameters of text layout analysis to adapt to the hand-held object images.

C. Prototype System Evaluation

The automatic ROI detection and text localization algorithms were independently evaluated as unit tests to ensure effectiveness and robustness of the whole system. We subsequently evaluated this prototype system of assistive text reading using images of hand-held objects captured by ten blind users in person.

Two calibrations were applied to prepare for the system test. First, we instructed blind users to place hand-held object within the camera view. Since it is difficult for blind users to aim their held objects, we employed a camera with a reasonably wide angle. In future systems, we will add finger point detection and tracking to adaptively instruct blind users to aim the object. Second, in an applicable blind-assistive system, a text localization algorithm might prefer higher recall by sacrificing some precision. We adjusted the parameters of our text localization algorithm and obtained another group of evaluation results, as precision 0.48, recall 0.72, f-measure 0.51. The higher recall ensures a lower miss (false negative) rate. To filter out false positive localizations, we could further employ some postprocessing algorithm based on scene text recognition or lexical analysis. This work will be carried out in future work.

Next, we evaluated the user-captured dataset of object text. The dataset was manually annotated by labeling the regions of the object of interests and the text regions inside the object of interest regions. In our algorithm evaluation, we defined a region as correctly detected if the ratio of the overlap area of a detected region and its ground truth region is no less than 3/4. Experiments showed that 225 of the 312 ground truth text regions were hit by our localization algorithm. By using the same evaluation measures as above experiments, we obtained precision 0.52, recall 0.62, and f-measure 0.52 on this dataset. The precision is lower than that on the Robust Reading Dataset. The images in the user-captured dataset have lower resolutions and more compact distribution of text information, so they generate low-quality edge maps and text boundaries, which result in improper spatial layouts and text structure features.

OCR is applied to the localized text regions for character and word recognition. Fig. 16 shows some examples of text localization and recognition of our proposed framework. We note that the recognition algorithm might not correctly and completely output the words inside localized regions. Additional spelling correction is likely required to output accurate text information. Our text reading system spends 1.87 s on average reading text from a camera-based image. The system efficiency can and will be improved by parallel processing of text extraction and device input/output, i.e., speech output of recognized text and localization of text regions in the next image are performed simultaneously.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have described a prototype system to read printed text on hand-held objects for assisting blind persons. In order to solve the common aiming problem for blind users, we have proposed a motion-based method to detect the object of interest, while the blind user simply shakes the object for a couple of seconds. This method can effectively distinguish the object of interest from background or other objects in the camera view. To extract text regions from complex backgrounds, we have proposed a novel text localization algorithm based on models of stroke orientation and edge distributions. The corresponding feature maps estimate the global structural feature of text at every pixel. Block patterns project the proposed feature maps of an image patch into a feature vector. Adjacent character grouping is performed to calculate candidates of text patches prepared for text classification. An Adaboost learning model is employed to localize text in camera-based images. Off-the-shelf OCR is used to perform word recognition on the localized text regions and transform into audio output for blind users.

Our future work will extend our localization algorithm to process text strings with characters fewer than three and to design more robust block patterns for text feature extraction. We will also extend our algorithm to handle nonhorizontal text strings. Furthermore, we will address the significant human interface issues associated with reading text by blind users.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and insightful suggestions that improved the quality of this manuscript.

REFERENCES

[1] World Health Organization. (2009). 10 facts about blindness and visual impairment [Online]. Available: www.who.int/features/factfiles/blindness/blindness_facts/en/index.html

[2] Advance Data Reports from the National Health Interview Survey (2008). [Online]. Available: http://www.cdc.gov/nchs/nhis/nhis_ad.htm

[3] International Workshop on Camera-Based Document Analysis and Recognition (CBDAR 2005, 2007, 2009, 2011). [Online]. Available: http://www.m.cs.osakafu-u.ac.jp/cbdar2011/

[4] X. Chen and A. L. Yuille, “Detecting and reading text in natural scenes,” in *Proc. Comput. Vision Pattern Recognit.*, 2004, vol. 2, pp. II-366–II-373.

[5] X. Chen, J. Yang, J. Zhang, and A. Waibel, “Automatic detection and recognition of signs from natural scenes,” *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 87–99, Jan. 2004.

[6] D. Dakopoulos and N. G. Bourbakis, “Wearable obstacle avoidance electronic travel aids for blind: A survey,” *IEEE Trans. Syst., Man, Cybern.*, vol. 40, no. 1, pp. 25–35, Jan. 2010.

[7] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *Proc. Comput. Vision Pattern Recognit.*, 2010, pp. 2963–2970.

[8] Y. Freund and R. Schapire, “Experiments with a new boosting algorithm,” in *Proc. Int. Conf. Machine Learning*, 1996, pp. 148–156.

[9] N. Giudice and G. Legge, “Blind navigation and the role of technology,” in *The Engineering Handbook of Smart Technology for Aging, Disability, and Independence*, A. A. Helal, M. Mokhtari, and B. Abdulrazak, Eds. Hoboken, NJ, USA: Wiley, 2008.

[10] A. Shahab, F. Shafait, and A. Dengel, “ICDAR 2011 robust reading competition: ICDAR Robust Reading Competition Challenge 2: Reading text in scene images,” in *Proc. Int. Conf. Document Anal. Recognit.*, 2011, pp. 1491–1496.

[11] K. Kim, K. Jung, and J. Kim, “Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1631–1639, Dec. 2003.

[12] *KReader Mobile User Guide*, knfb Reading Technology Inc. (2008). [Online]. Available: http://www.knfbReading.com

[13] 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 Image Process.*, vol. 16, no. 8, pp. 2117–2128, Aug. 2007.

[14] S. M. Lucas, “ICDAR 2005 text locating competition results,” in *Proc. Int. Conf. Document Anal. Recognit.*, 2005, vol. 1, pp. 80–84.

[15] L. Ma, C. Wang, and B. Xiao, “Text detection in natural images based on multi-scale edge detection and classification,” in *Proc. Int. Congr. Image Signal Process.*, 2010, vol. 4, pp. 1961–1965.

[16] R. Manduchi and J. Coughlan, “(Computer) vision without sight,” *Commun. ACM*, vol. 55, no. 1, pp. 96–104, 2012.

[17] N. Nikolaou and N. Papamarkos, “Color reduction for complex document images,” *Int. J. Imaging Syst. Technol.*, vol. 19, pp. 14–26, 2009.

[18] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst., Man Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[19] T. Phan, P. Shivakumara, and C. L. Tan, “A Laplacian method for video text detection,” in *Proc. Int. Conf. Document Anal. Recognit.*, 2009, pp. 66–70.

[20] *The Portset Reader*, TVI Technologies for the Visually Impaired Inc., Hauppauge, NY, USA. (2012). [Online]. Available: http://www.tvi-web.com/products/porsetreader.html

[21] L. Ran, S. Helal, and S. Moore, “Drishti: An integrated indoor/outdoor blind navigation system and service,” in *Proc. 2nd IEEE Annu. Conf. Pervasive Comput. Commun.*, 2004, pp. 23–40.

[22] ScanTalker, Bar code scanning application to help Blind Identify over one million products. (2006). [Online]. Available: http://www.freedomscientific.com/fs_news/PressRoom/en/2006/ScanTalker2-Announcement_3–30–2006.asp

[23] H. Shen, K. Chan, J. Coughlan, and J. Brabyn, “A mobile phone system to find crosswalks for visually impaired pedestrians,” *Technol. Disability*, vol. 20, no. 3, pp. 217–224, 2008.

[24] H. Shen and J. Coughlan, “Grouping using factor graphs: An approach for finding text with a camera phone,” in *Proc. Workshop Graph-based Representations Pattern Recognit.*, 2007, pp. 394–403.

[25] M. Shi, Y. Fujisawab, T. Wakabayashia, and F. Kimura, “Handwritten numeral recognition using gradient and curvature of gray scale image,” *Pattern Recognit.*, vol. 35, no. 10, pp. 2051–2059, 2002.

[26] P. Shivakumara, T. Phan, and C. L. Tan, “A gradient difference based technique for video text detection,” in *Proc. Int. Conf. Document Anal. Recognit.*, 2009, pp. 66–70.

[27] S. Shoval, J. Borenstein, and Y. Koren, “Auditory guidance with the Navbelt: A computerized travel for the blind,” *IEEE Trans. Syst., Man, Cybern. C. Appl. Rev.*, vol. 28, no. 3, pp. 459–467, Aug. 1998.

[28] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” presented at the IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit., Fort Collins, CO, USA, 1999.

[29] Y. Tian, M. Lu, and A. Hampapur, “Robust and efficient foreground analysis for real-time video surveillance,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit*, 2005, pp. 1182–1187.

[30] P. Viola and M. J. Jones, “Robust real-time face detection,” *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[31] X. Yang, Y. Tian, C. Yi, and A. Arditi, “Context-based indoor object detection as an aid to blind persons accessing unfamiliar environments,” in *Proc. ACM Multimedia*, 2010, pp. 1087–1090.

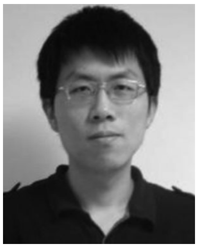
[32] X. Yang, S. Yuan, and Y. Tian, “Recognizing clothes patterns for blind people by confidence margin based feature combination,” in *Proc. ACM Multimedia*, 2011, pp. 1097–1100.

[33] C. Yi and Y. Tian, “Text string detection from natural scenes by structure based partition and grouping,” *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2594–2605, Sep. 2011.

[34] C. Yi and Y. Tian, “Assistive text reading from complex background for blind persons,” in *Proc. Int. Workshop Camera-Based Document Anal. Recognit.*, 2011, vol. LNCS-7139, pp. 15–28.

[35] C. Yi and Y. Tian, “Text detection in natural scene images by stroke gabor words,” in *Proc. Int. Conf. Document Anal. Recognit.*, 2011, pp. 177–181.

[36] J. Zhang and R. Kasturi, “Extraction of text objects in video documents: recent progress,” in *Proc. IAPR Workshop Document Anal. Syst.*, 2008, pp. 5–17.



Chucai Yi (S’12) received the B.S. and M.S. degrees from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2007 and 2009, respectively. Since 2009, he has been working toward the Ph.D. degree in computer science at the Graduate Center, The City University of New York, New York, NY, USA.

His research focuses on text detection and recognition in natural scene images. His research interests include object recognition, image processing, and machine learning. His current work is to develop new computer vision algorithms and systems to help people with severe vision impairment to independently find doors, rooms, elevators, stairs, bathrooms, and other building amenities in unfamiliar indoor environments.

Yingli Tian (M’99–SM’01) received the B.S. and M.S. degrees from Tianjin University, Tianjin, China, in 1987 and 1990, respectively, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 1996.

After holding a faculty position at the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, Beijing, China, she joined Carnegie Mellon University in 1998, where she was a Postdoctoral Fellow of the Robotics Institute. She then worked as a Research Staff Member at the IBM T. J. Watson Research Center from 2001 to 2008. She is currently a Professor in the Department of Electrical Engineering and the Department of Computer Science at the Graduate Center, The City University of New York, New York, NY, USA. Her current research focuses on a wide range of computer vision problems from event detection and analysis, assistive technology, to human identification, facial expression analysis, and video surveillance.

Aries Arditi received the B.A. degree from Connecticut College, New London, CT, USA, in 1973 (philosophy and psychology), and the M.A. and Ph.D. degrees from New York University, New York, NY, USA, in 1979 (experimental psychology).

After postdoctoral fellowships in neuroscience at Northwestern University, Evanston, IL, USA, and New York University, he has spent most of his 30-year career at Lighthouse International, New York, NY, USA, including two years as a Research Staff Member at IBM T.J. Watson Research Center, doing psychophysical vision research with an emphasis on low vision and blindness accessibility. He is currently a Principal Scientist with Visibility Metrics LLC, Chappaqua, NY, USA, a research consulting firm, and the President of the Mars Perceptrix Corporation, developer of innovative tests of vision function. His current research focuses on human interface issues associated with low vision and blindness, and on psychophysical and usability issues associated with visual prostheses.