Assistive Text Reading from Natural Scene for Blind Persons

Chucai Yi and Yingli Tian

Abstract Text information serves as an understandable and comprehensive indicator, which plays a significant role in navigation and recognition in our daily lives. It is very difficult to access this valuable information for blind or visually impaired persons, in particular, in unfamiliar environments. With the development of computer vision technology and smart mobile applications, many assistive systems are developed to help blind or visually impaired persons in their daily lives. This chapter focuses on the methods of text reading from natural scene as well as their applications to assist people who are visually impaired. With the research work on accessibility for the disabled, the assistive text reading technique for the blind is implemented in mobile platform such as smart phone, tablet, and other wearable device. The popularity and interconnection of mobile devices would provide more low-cost and convenient assistance for blind or visually impaired persons.

1 Introduction

With the development of computer vision technology and smart mobile applications, many assistive systems are developed to help blind or visually impaired persons in their daily lives. This chapter focuses on the methods of text reading from natural scene as well as their applications to assist people who are visually impaired. With the research work on accessibility for the disabled, the assistive text reading technique for the blind is implemented in mobile platform such as smart phone, tablet, and other wearable device. The popularity and interconnection of mobile de-

Chucai Yi
HERE North America, 425 W Randolph St, Chicago, IL 60606
e-mail: gschucai@gmail.com

Yingli Tian
The City College of New York, Convent Avenue at 138th Street, New York, NY 10031
e-mail: ytian@ccny.cuny.edu

vices would provide more low-cost and convenient assistance for blind or visually impaired persons.

Of the 314 million visually impaired persons worldwide, 45 million are blind [1]. In the United States, the 2008 National Health Interview Survey (NHIS) 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. With the help of guide cane and guide dog, visually impaired persons perceive surrounding environments by hearing, smell, or touch, so that they are able to discern objects by shape and material, and avoid obstacles in the way-finding process.

However, it is beyond their capabilities to acquire text information from natural scene. Some office buildings and public facilities do provide blind-assistant signage in Braille. However, in most cases, text information in natural scene is prepared for people with normal vision, in the form of printed fonts at a signage board.

Text information serves as an understandable and comprehensive indicator, which plays a significant role in navigation and recognition in our daily lives. It is very difficult to access this valuable information for blind or visually impaired persons, in particular, in unfamiliar environments. However, recent developments in computer vision, digital cameras, and portable computers make it feasible to develop camera-based assistive products to help them. These blind-assistant systems usually combine computer vision technology with other existing commercial products such OCR, GPS systems.

This chapter is organized as follows. Section 2 introduces the related work on the requirements of blind users and the available effective methods of scene text extraction. Section 3 presents a technical framework of scene text extraction. Section 4 describes two blind-assistant prototype systems of text recognition respectively for hand-held object recognition and indoor navigation. Section 5 introduces blind-assistant system design for accessibility on mobile platform.

2 Related Work

A blind-assistant system should be comfortable to wear, portable, efficient, low-cost, and user friendly. These basic requirements are closely associated with the system design and implementation. Many blind-assistant systems have been developed [3, 43, 44, 45, 46, 47]. In general, a blind-assistant system contains three main components: capture, process, and feedback. More descriptions of the blind-assistant system interface design will be presented later in this chapter.

The *capture* component of a blind-assistant system is to help blind user perceive surrounding objects. For example, white cane can be considered as a simple *capture* component. It perceives surrounding objects by touch, and it is portable and easy to hold. In computer vision based blind-assistant systems, the *capture* component is usually a camera, which can be attached to a wearable device, so that the blind or visually impaired persons can conveniently take it everywhere. To clearly capture surrounding objects in different distances, some systems [21] took multiple cam-

eras with different viewpoints and focuses. In most cases, the cameras are attached to a sunglass [14] or a helmet [45, 56]. Many wearable cameras such as Autographer [59], MeCam [60], Looxcie [61], and GoPro [62] have developed for portable photography or entertainment, but they can also be used in blind assistance. The Google-glass [55] may also be used as a basic device to capture data in blind assistance. Recently, RGBD cameras that are able to capture depth information were often used in blind-assistant navigation [25] or indoor scene indexing [24].

The *process* component of a blind-assistance system is to extract valuable information, which can be provided to blind persons to recognize surrounding objects or find their ways to destination. The image/video data captured by cameras provides large amount of information about the surrounding objects in natural scenes, in which text serves as the most straightforward and informative indicators. Thus in this chapter, as one of the main tasks of the *process* component, we will focus on extracting surrounding text information for blind or visually impaired users. Our research group has developed a series of computer vision-based methods for blind people to recognize signage [13] and object labels [15], recognize objects and clothes patterns [16, 17, 18], independently access and navigate unfamiliar environments [19, 20, 21, 22]. Tian et al. developed a proof-of-concept computer vision-based way finding aid for blind people to independently access unfamiliar indoor environments [42]. We also developed several methods and prototype systems [12, 14] to extract text information from natural scenes. Scene text extraction is usually divided into two steps: detection and recognition. Text detection is to find out image regions containing text characters and strings. Text detection algorithms [31, 49, 50, 51] were mostly involved in color uniformity, gradient distribution, and edge density of text regions. Text recognition is to transform the image-based text information into readable text codes [52, 53, 54]. Text recognition algorithms were mostly based on the design of feature representation for text character recognition, and the combination of vision-based recognition and lexicon-based model for word recognition.

The *feedback* component of a blind-assistant system is to provide the extracted information in an acceptable way to the blind users. The feedback should satisfy several requirements of blind or visually impaired persons who are located in an unfamiliar environment or hold an unfamiliar object. It must be simple, in time, and understandable. A straightforward way of information feedback is indicative speech, which transforms the vision-based information into audio-based information so that the blind or visually impaired persons can hear it. Many systems adopted this scheme [9, 14, 47]. In addition, many sonar-based systems were designed to help blind person avoid obstacles [4, 5, 6, 7, 8]. The ultrasound is transmitted and received to measure the distances and directions of possible obstacles that reflect it, and the blind or visually impaired persons can obtain real-time notifications. However, these systems cannot provide vision-based information like text signage. In addition to acoustical feedback such as audio and speech, some systems designed haptic feedback based on regular vibration of a wearable device. The device used for haptic feedback can be a helmet [56], finger [48], or tongue display unit in the mouth [10, 11, 48, 57].

3 Scene Text Extraction

In this section, we describe the technical framework of scene text extraction that contains two main steps: text detection and text recognition.

3.1 Scene Text Detection

To extract text information from camera-based natural scene images, first, we need to separate the contours or blobs that possibly contain text characters from background outliers. These possible text characters are defined as candidate characters. In our framework, two algorithms are developed to detect candidate characters, which are respectively associated with contours and blobs of text characters.

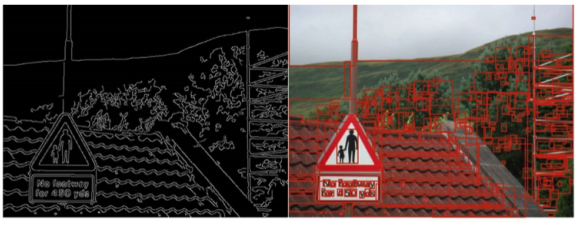
3.1.1 Candidate Character Detection

A. Contours in Edge Map

Candidate characters normally generate regular and closed contours in the edge map of scene image. Thus a straightforward method of detecting candidate characters is to first generate all object contours in a scene image, and then find out the contours probably generated by scene text characters.

We apply Canny edge detector [26] to acquire the edge map of a natural scene image. In low-level image processing, a contour is defined as a set of connected edge pixels. Fig. 1 illustrates the detected contours in a natural scene image. Among these contours, some geometrical constraints are defined to detect the contours of candidate characters.

Both Canny edge detection and object contour generation are computationally efficient. However, without predefined constraints like color uniformity to analyze the blobs, the contours of candidate characters would be mixed with the contours of background objects, and it is difficult to distinguish them. A more effective operator is presented in next section to extract candidate characters.



(a) (b)

Fig. 1 (a) Canny edge map of a scene images. (b) Bounding boxes of object contours in the form of connected edge pixels, obtained from edge map.

B. Maximum Stable Extremal Region

In addition to contours in specific geometrical constraints, candidate characters and their attachment surfaces are usually painted with uniform color. Thus we can extract these candidate characters in the form of blobs, which include not only the contour but also torso information.

Maximum Stable Extremal Region (MSER) operator was proposed in [27], which has been used as a blob detection technique for a long time in computer vision field. MSER is defined based on an extension of the definitions of image and set. Let image I be a mapping such that $I : D \subseteq \mathbb{Z}^2 \rightarrow S$, where D denotes the set of all pixels in the image, and S is a totally ordered set with reflexive, anti-symmetric and transitive properties. It means each pixel in an image is mapped into a value, and each pair of pixels can be in comparison with each other through their respective values. In real applications, this value is defined as pixel gray intensity.

An adjacent relation is defined as A. For two neighboring pixels a_i and a_{i+1}, we have $a_i A a_{i+1}$ if and only if $|I(a_{i+1}) - I(a_i)| \leq Threshold$, where $I(a_i)$ denotes the mapped gray intensity at pixel a_i. Then an MSER region Q is defined as a contiguous subset of D, such that for each $p,q \in Q$, there is a sequences p, a_1, a_2, a_3, $\cdots$, a_n, q, where pAa_1, a_1Aa_2, $\cdots$, a_iAa_{i+1}, $\cdots$, a_nAq. Here, A represents the intensity difference and p, q or a_i is in the form of 2-dimensional vector, representing the x-coordinate and y-coordinate of an image pixel. As shown in Fig. 2, MSER generates connected components of text characters in a scene image, while edge map only gives the contours of text characters. Further, MSER map filters out the foliage thoroughly.

Since MSER cannot confirm the intensity polarity of text and attachment surface, that is, not able to distinguish white-text-in-black-background from black-text-in-

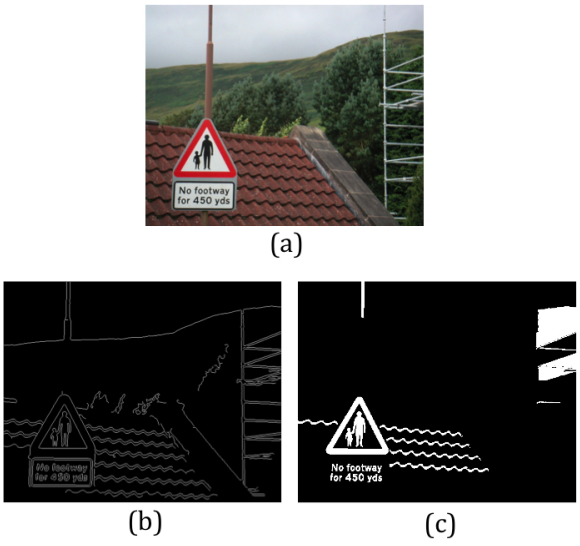


Fig. 2 (a) Canny edge map of a scene images. (b) Bounding boxes of object contours in the form of connected edge pixels, obtained from edge map.

white-background, both text and attachment surface will be extracted as candidate characters in MSER map. However, attachment surface components can be easily removed by defining some geometrical properties. Moreover, MSER has several specific properties compatible with the requirement of extracting candidate characters from natural scene image. Firstly, MSER is invariant to affine transformation of image intensities. Secondly, MSER extraction is very stable since a region is selected only if its support is nearly the same over a range of thresholds. Thirdly, MSER is scale-invariant and is able to extract candidate characters in multiple scales without any pre-processes of original natural scene image.

MSER extraction is efficient enough to satisfy mobile applications, because its time complexity in the worst case is $O(n)$ where n represents the number of pixels in the image. However, in our experiments, MSER blob detection usually takes about 2-3 times the computational time as contour search in edge map.

C. Geometrical Constraints of Candidate Characters

Not all contours in an edge map and not all blobs generated by MSER operator come from text characters, which also compose that from non-text background outliers.

To remove the non-text background outliers from the set of candidate characters, we define a group of geometrical constraints. In these constraints, a candidate characters C, in the form of either contour or blob, is described by several geometrical properties: $height(.)$, $width(.)$, $coorX(.)$, $coorY(.)$, $area(.)$, and $numInner(.)$, which represent height, width, centroid x-coordinates, centroid y-coordinates, area, and the number of inner candidate characters respectively.

We define a group of geometrical constraints based on above measurements to ensure that the preserved candidate characters are real text characters as possible. Since we will further perform text string layout analysis and text structure modeling to remove false positive candidate characters, the constraints defined in this step are not very strict.

$$\begin{aligned} & height(C) > 15 pixels \\ & 0.3 \leq \frac{width(C)}{height(C)} \leq 1.5 \\ & numInner(C) \leq 4 \\ & \frac{1}{10} \cdot ImageWidth \leq coorX(C) \leq \frac{9}{10} \cdot ImageWidth \\ & \frac{1}{10} \cdot ImageHeight \leq coorY(C) \leq \frac{9}{10} \cdot ImageHeight \end{aligned} \tag{1}$$

The involved geometrical constraints are presented in Eq. (1). First of all, the candidate character component cannot be too small, and otherwise we will treat it as background noise. It also means that our whole framework of scene text extraction requires enough resolution of camera-captured scene text image. Second, the aspect

ratio of a character should be located in a reasonable range. Under a threshold of aspect ratio, we might also remove some special text characters like l, but it is very possible to restore this false removal by generating text strings. Third, we define some constraints related to the number of nested candidate character components as presented in [28]. Fourth, we observe that many background outliers obtained from the above partition methods are located at the boundaries of scene images. Thus the candidate character components whose centroids are located at the $1/10$ boundary of the images are not taken in account in further processes.

The constraints in Eq. (1) do not depend on any learning models to decide the parameters as in [29, 31]. Instead, all the involved geometrical constraints are weak conditions, with the preservation of true text characters in higher priority than the removal of false positive background outliers. Therefore, only the obvious background outliers are filtered by the geometrical constraints. The remaining false positive candidate characters will be handled in the extraction of text string.

3.1.2 Text String Detection

A set of candidate character components is created in the form of contours or blobs from an input image. Most candidate characters are not true scene text characters but non-text background objects in uniform color or some portions of an object under uneven illumination. Geometrical constraints as described in last section cannot remove them, so we design more discriminative layout characteristics of scene text from high-level perspective. Text in natural scene mostly appears in the form of words and phrases instead of single characters. It is because words and phrases are more informative text information, while single character usually serves as a sign or symbol. Words and phrases are defined as text strings, and we attempt to find out possible text strings by combining neighboring candidate characters. Therefore, in this chapter, we define a text string as a combination of neighboring candidate characters.

In this section, a method named as adjacent character grouping is presented in [30] to detect text strings among the extracted candidate characters. Text strings in natural scene images usually appear in horizontal alignment and each character in a text string has at least one sibling at adjacent positions. Furthermore, a text character and its siblings in a text string have similar sizes and proper distances. Therefore, the idea of adjacent character grouping is removing the candidate characters that do not have any siblings.

In adjacent character grouping method, the main problem is how to decide whether two candidate characters C_1 and C_2 are sibling characters. According to our observations and statistical analysis of text strings, we define 3 geometrical constraints as follows:

1) Considering the approximate horizontal alignment of text strings in most cases, the centroid of candidate character C_1 should be located between the upper-bound and lower-bound of the other candidate character C_2.

2) Two adjacent characters should not be too far from each other despite the variations of width, so the distance between two connected components should not be greater than T_2 times the width of the wider one.

3) For text strings aligned approximately horizontally, the difference between y-coordinates of the connected component centroids should not be greater than T_3 times the height of the higher one.

In our applications, we set the thresholds $T_1 = 0.5$, $T_2 = 3$, $T_3 = 0.5$. For each candidate character C_i, a sibling set $S(C_i)$ is generated, where $1 \leq i \leq |\mathbb{C}|$ and $|\mathbb{C}|$ represents the number of candidate characters obtained from image partition.

First, an empty sibling set is initialized as $S(C_i) := \phi$. We transverse all candidate characters except C_i itself. If a candidate character C_i' satisfies all above constraints with C_i, we add it into the sibling set as $S(C_i) := S(C_i) \cup \{C_i'\}$. Second, all the sibling sets compose a set of adjacent groups $\Lambda = \{A_i | A_i := S(C_i)\}$, where a sibling set is initialized to be adjacent group A. Third, the set of adjacent groups is iteratively updated by merging the overlapping adjacent groups. An adjacent group is a group of candidate character components that are probably character members of a text string. As Eq. (2), if two adjacent groups A_i and A_j in Λ have intersection, they will be merged into one adjacent group. This merging operation is iteratively repeated until no overlapping adjacent groups exist.

In the resulting set of adjacent groups, each adjacent group A_i is a set of candidate characters in approximate horizontal alignment, which will be regarded as a text string, as shown in Fig.3. Then a bounding box is generated for each adjacent group to represent the region of a localized text string in natural scene image.

$$\begin{aligned} &\forall A_i, A_j \in \Lambda, \quad if \quad A_i \cap A_j \neq \phi, \\ &\qquad\qquad then \quad A_i := A_i \cup A_j \quad and \quad A_j := \phi \end{aligned} \tag{2}$$

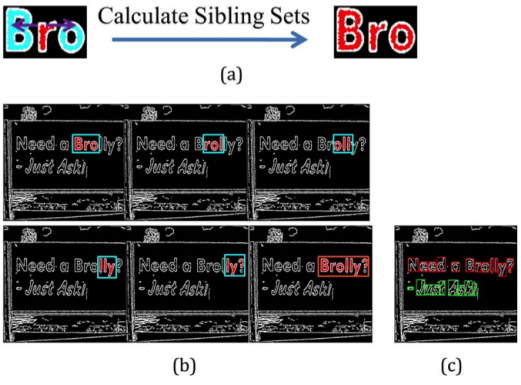


Fig. 3 (a) Sibling group of the connected component r where B comes from the left sibling set and o comes from the right sibling set; (b) Merge the sibling groups into an adjacent character group corresponding to the text string Brolly?; (c) Two detected adjacent character groups marked in red and green respectively [30].

3.2 Scene Text Recognition

The extraction of candidate character and text string is able to efficiently localize most text strings from natural scene image. However, to acquire valuable text information for blind or visually impaired persons, there are still two problems to be solved. First, above steps of text character and string extraction are pixel-based processing and statistic-based parameter setting, and they will bring in many false text strings from the natural scene image with complex background. Second, for true text strings, a method or off-the-shelf system is required to recognize the text information in it, which transforms the image-based text information into readable text codes. To solve these two problems, feature representations related to inner text structure are designed in two different ways.

3.2.1 Text String Classification

In the first problem of scene text recognition, feature representation is proposed to model structural insights of text characters and strings. At first, each text string, localized by above steps in last section, is defined as a sample. It may be a positive sample, which means this region truly contains text information. It may also be a negative sample, which means that this region is generated by background outliers, e. g. some texture similar to text character such as bricks, window grids and foliage, and some objects rendered by specific illumination change, as shown in Fig.4. To distinguish text from non-text outlier, we design text structure-related feature representations by using Haar-like block patterns and feature maps. Fig.5 illustrates the flowchart of the text string classification process.

To extract structural information from these samples, Haar-like filters are designed in the form of block patterns, as shown in Fig.6. Each block pattern consists of white regions and gray regions in specific ratio. It will be resized into the same size as a sample, and used as a mask. Then specific calculation metrics are defined based on these block patterns for extracting structural features.

A simple idea of feature extraction is to apply these block patterns directly to the text string samples, and calculate Haar-like features from intensity values of the image patches. However, unlike face detection [32], the sole intensity values cannot completely represent structure of text strings.

Fig. 4 Some examples of text string samples in the form of image patches.

To model text structure, we design a set of feature maps for the samples, in which the physical meaning of each pixel is transformed from intensity value to some measurements related to text structure. The structure-related measurements are mostly based on gradient, edge density and stroke. The involved feature maps include gradient, stroke width, stroke orientation, and edge density [14].

In feature map of a text string sample, each pixel is transformed from intensity to some measurements related to text structure. Each pixel reflects text structural configuration from a local perspective. By tuning the parameters of generating feature maps under a design scheme, we can obtain multiple feature maps. In our framework, we design 3 gradient maps, 2 stroke width maps, 14 stroke orientation maps, and 1 edge distribution map, to which 6 Haar-like block patterns are applied for calculating feature values. Each combination of a feature map, a block pattern and a calculation scheme [12] is developed into a weak classifier in Adaboost learning model. By using the localized regions in above text detection steps as training samples, Adaboost learning model selects an optimized subset of weak classifiers and weighted combine them to effectively classify text from non-text outlier. This classifier is actually an optimized combination of a subset of weak classifiers.

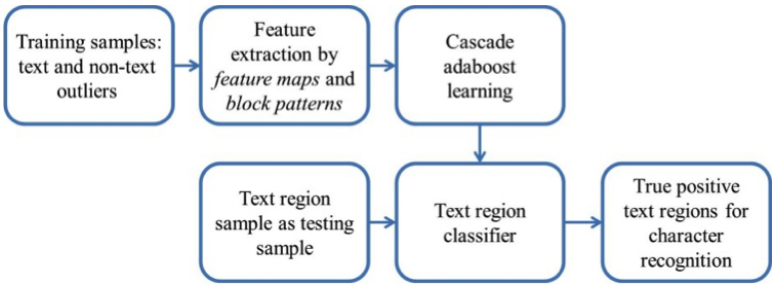


Fig. 5 Diagram of the proposed Adaboost learning based text string classification algorithm.

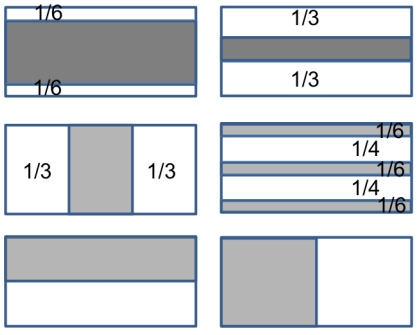


Fig. 6 Some examples of Haar-like block patterns to extract text structural features. Features are obtained by the absolute value of mean pixel values in white regions minus those in black regions.

3.2.2 Scene Text Character Recognition

In the second problem of scene text recognition is to transform image-based text information into readable text codes. A straightforward way is to apply off-the-shelf optical character recognition (OCR) software [33, 34, 35] to the text strings. However, most OCR systems are designed for scan documents or hand-written recognition, and they are not robust to background interference and various text patterns. Thus we also propose a feature representation for recognizing total of 62 categories of scene text characters (STCs), which include 10 digits (0-9) and 26 English letters in both upper (A-Z) and lower cases (a-z).

The most significant role in STC recognition is to work out a multi-class classifier to predict the category of a given STC. In our system, Chars74K [41] dataset is adopted to train this multi-class classifier. Fig.7 illustrates some examples of STCs cropped from text strings. We observe that the STCs have irregular patterns and similar structure to each other.

A feature representation is designed to model the representative structure of each of the 62 STC categories and the discriminative structure between STC categories. Each STC sample is mapped into its feature representation in the form of a vector.

Fig. 7 Some examples of STCs cropped from text strings. Most STCs have similar structure to another counterpart.

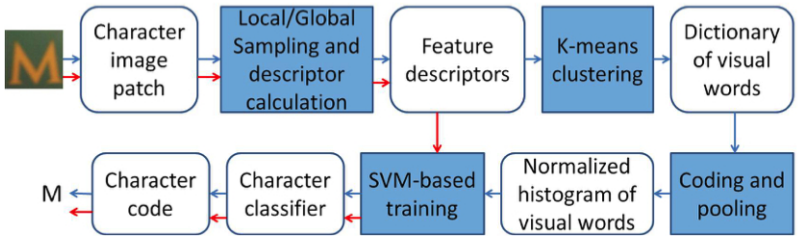


Fig. 8 Some examples of STCs cropped from text strings. Most STCs have similar structure to another counterpart.

Then it is input into SVM learning model to obtain the multi-class classifier. Fig.8 demonstrates the whole process of STC recognition.

First, low-level features are extracted from STC image patches to describe appearance and structure of STCs from all 62 STC categories. Through performance evaluations of 6 state-of-the-art low-level feature descriptors [37], our framework selects Histogram of Oriented Gradient [36] descriptor applied to the key points sampled from STC image patches.

Second, key-point sampling and feature coding/pooling play a significant role. We made a comparative study of several methods of key-point sampling and schemes of feature coding/pooling [37]. In dense sampling, soft-assignment coding and max pooling scheme obtain the best performance. However, the global sampling obtains even better performance, which uses the whole character patch as a key-point neighborhood window to extract features. In global sampling, key point detection, coding and pooling process are all skipped to largely reduce information loss.

Third, STC recognition depends on SVM-based training and testing over the STC samples. While the learning process in text string classification is to select the representative combinations of feature maps, Haar-like block patterns and calculation schemes to distinguish text from non-text, the learning process in STC prediction treats the feature representation vector of an STC sample as a point in feature space, which describes the STC structure. Thus we would adopt SVM learning model [38] to generate hyper-planes in feature space as STC classifier, rather than the Adaboost algorithm to select optimized combinations of the weak classifiers. In the SVM-based learning process, we adopt multiple SVM kernels, including Linear Kernel and χ^2 Kernel, to evaluate the feature representations of STC structure. In recent work, deep learning framework demonstrates better performance in scene text recognition. But the implementation of the multi-layer convolutional neutral network depends on GPU computational units, which are usually not available for wearable mobile devices in blind-assistant systems.

4 Blind-Assistant Applications of Scene Text Extraction

Many blind assistant reading systems are developed to help visually impaired people reading object bar code or documents through some wearable devices. 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.

To assist blind or visually impaired people to read text from hand-held objects, a camera-based text reading prototype is developed to track the object of interest within the camera view and extract print text information from the object label. Our framework of scene text extraction can effectively handle complex background and multiple text patterns, and obtain text information from both hand-held objects and nearby signage, as shown in Fig.9. Two corresponding blind-assistant applications are developed on the basis of scene text extraction.

To assist blind or visually impaired people to read text from hand-held objects, a camera-based text reading prototype is developed to track the object of interest within the camera view and extract print text information from the object label. Our framework of scene text extraction can effectively handle complex background and multiple text patterns, and obtain text information from both hand-held objects and nearby signage, as shown in Fig.9. Two corresponding blind-assistant applications are developed on the basis of scene text extraction.

4.1 Reading Text Labels for Hand-held Object Recognition

In most assistive reading systems, users have to position the object of interest within the center of the cameras view. To ensure the hand-held object be captured within the camera view, we use a wide-angle camera to accommodate users with only approximate aim. However, this wide-angle camera will also capture many other text objects (for example while shopping at a supermarket). To extract the hand-held object from the camera image, we develop a motion-based scheme to acquire a region of interest (ROI) of the object by asking the blind user shakes the object for a couple of seconds. This scheme is based on background subtraction-based motion detection [42]. Then we perform scene text extraction from only this ROI, including detecting text strings and recognizing text codes. In the end, the recognized text codes are output to blind users in audio or speech. To present how our prototype system works, a flowchart is presented in Fig.10.

A prototype system of scene text extraction is designed and implemented in PC platform and Mobile platform [42, 39]. This system consists of three main components: scene capture, data processing, and audio output. The scene capture com-

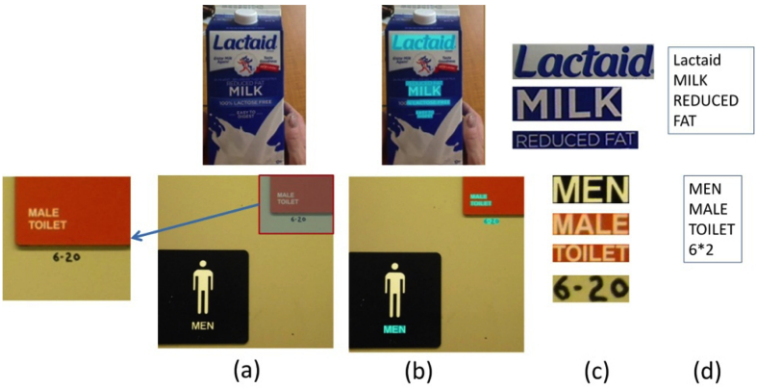


Fig. 9 Two examples of text extraction by the prototype system from camera-captured images. Top: a milk box; Bottom: a men bathroom signage. (a) camera-captured images; (b) localized text strings (marked in blue); (c) text strings cropped from image; (d) text codes recognized by OCR [14].

ponent collects surrounding scenes or objects and generates high-quality image frames. The data processing component is used for deploying our proposed framework of scene text extraction. The audio output component is to inform the blind user of recognized text codes. This simple hardware configuration proves the portability of the assistive text reading system. The prototype system has been used to assist blind or visually impaired people to recognize hand-held object as described, as shown in Fig.11.

To evaluate the performance of hand-held object recognition system, following the Human Subjects Institutional Review Board approval, we recruited 10 blind persons to collect a dataset of reading text on hand-held objects. The blind user wore a camera attached on sunglasses to capture images of the objects in his/her hand, as illustrated in Fig.11. 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 to keep head (where the camera is fixed) stationary for a few seconds and subsequently shake the object for an additional couple of seconds to allow our system detect the object of interest based on the motion. Then the user rotated each object several times to ensure the main text on the object are exposed and captured. We manually extracted 116 captured images and labeled 312 text regions of object labels.

In our evaluations, a region is correctly detected if the ratio of the overlapping area of a detected text region and its ground truth region is no less than 3/4. Ex-

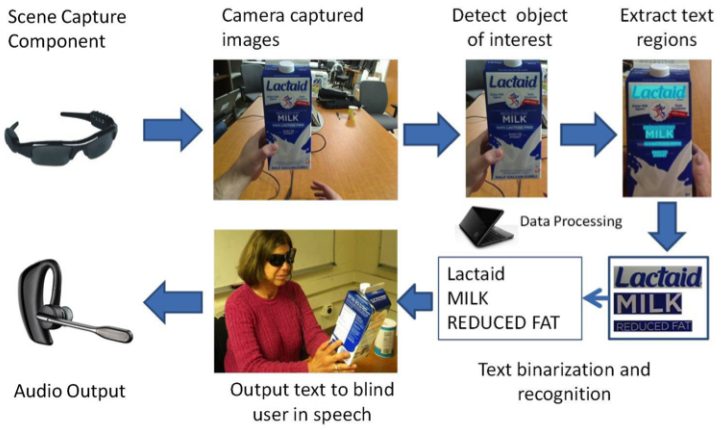


Fig. 10 Flowchart of prototype system to read text from hand-held objects for blind users [14].

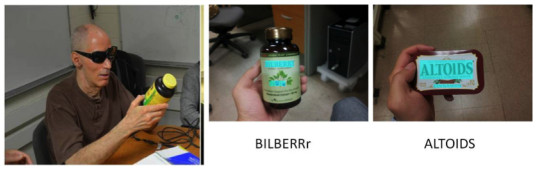


Fig. 11 Prototype system assists blind user read text information from hand-held objects, including the detected text strings in cyan and the recognized text codes.

periments demonstrate that 225 of the 312 ground truth text regions are correctly detected by our localization algorithm. Some examples of extracted scene text from hand-held objects are illustrated in Fig.12, proving that our proposed framework is suitable for real applications. To further improve the accuracy of text detection and recognition, the practical system would restrict the range of possible recognized words by a prior dictionary of common words that are frequently printed in hand-held objects. The extracted text results are output by audio only if it has close edit distance to some word in the dictionary.

Currently, the system efficiency mainly depends on the efficiency of scene text extraction in each image or video frame. However, through the design of parallel processing for text extraction and device input/output, the efficiency of this assistant reading system can be further improved. That is, speech output of recognized text in the current frame and localization of text strings in the next image are performed simultaneously.

4.2 Reading Text Signage for Indoor Navigation

A blind-assistant prototype system is designed for hand-held object recognition in last section. It can be further extended to indoor navigation, by extracting indicative information from surrounding text signage in indoor environment. In most cases, indoor navigation is to guide blind users to a targeted destination such as an office, a restroom, or an elevator entrance. All of them have doors by a signage with a room name or a room number. The people with normal vision can refer a floor plan map to find their ways, but blind or visually impaired people cannot acquire this

Fig. 12 (a) Some results of text detection on the blind 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 bottom [14].

information. Thus our proposed prototype system can perceive their current location and generate a proper path from current location to their destination. The hardware of this prototype system is similar to the system of hand-held object recognition in last section, including a wearable camera, a process unit, and audio output device. However, the system implements indoor navigation by adding the door detection.

In indoor environments, doors and elevators serve as important landmarks and transition points for way finding. They also provide entrance and exit information. Thus, an effective door detection method plays an important role in indoor navigation. We develop the vision-based door detection method [40] to localize doors for blind users. This method depends on a very general geometric door model, describing the general and stable features of a door frameedges and corners, as shown in Fig.13. Our method can handle complex background objects and distinguish doors from other door-like shapes such as bookshelves. After detecting doors, scene text extraction is performed within the door region or its immediately neighboring region to obtain text information related to room names and room numbers, as shown in Fig.14. Both the localization and navigation processes are based on accurate scene text extraction. Fortunately, the indoor environment mostly does not contain too much background interferences, and the text signage has relatively fixed pattern, e.g., room number contains only digits in print format, and restroom is usually marked by MEN, WOMEN, or RESTROOM. Thus the proposed scene text extraction will adapt its parameters to this indoor navigation application.

By using the extracted information from text signage, blind or visually impaired person can better perceive his/her current location and surrounding environment. Furthermore, most buildings have floor plan maps as tourist guide. A floor plan map contains room numbers and relative locations of the offices, restrooms, and elevator entrances. The data of floor plan map can be combined with the extracted text information to figure out blind-assistant navigation prototype in unfamiliar buildings.

A prototype design of floor plan based way-finding system can be found in [21]. A floor plan map is first parsed into a graph, in which a room is defined as a node (see Fig.15). Each pair of nodes is connected, and an available path of way finding is defined for each connection. For example, in Fig.15 (c), the yellow line corre-

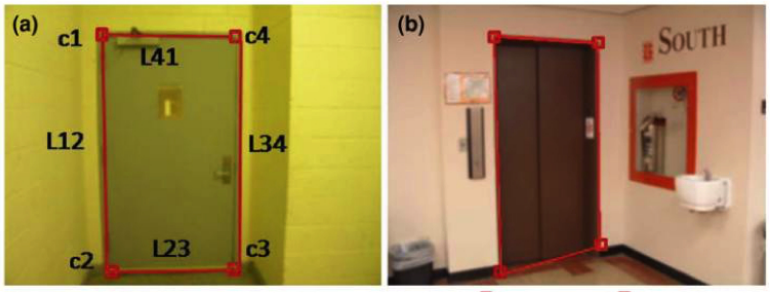


Fig. 13 (a) Edges and corners are used for door detection. (b) Door detection under cluttered background.

sponds to a proper path marked in yellow in Fig.15 (b) from room 632 to room 623. According to the length and the number of turning corners of the path, a cost value is assigned to its corresponding connection in the graph. In this weighted graph, the current location of a blind user is regarded as a starting point while his/her destination is regarded as an ending point. In the navigation process, our system finds out a path with minimum cost value and then generates the corresponding path to destination based on the floor plan map.

5 Blind-Assistant System Interface Design

The interface design always plays a significant role in the development a blind-assistant system. A well-developed system should provide safe, comfortable, and efficient services that are compatible with the daily life of blind or visually impaired persons. Our research group invited 10 visually impaired persons to survey user interface preference [23]. These 10 persons are all well-educated, employed (or retired after employment), and familiar with blind-assistant technology. Through a questionnaire, we collect their advices and requirements of blind-assistant systems. The associated problems and solutions of interface design will be presented in this section according to the three main components of blind-assistant system as described in Section 2, which are *capture*, *process* and *feedback*.

The *capture* component of a compute vision based blind-assistant system is normally a wearable camera, which is attached to a sunglass, helmet, kneepad or wrist. These wearable cameras should satisfy specific requirements of blind or visually impaired persons. First, they should be easy and comfortable to put on and take off. Although the wearable devices are light and compatible with human face or body, almost all the users will choose to take them off if not necessary. Second, they

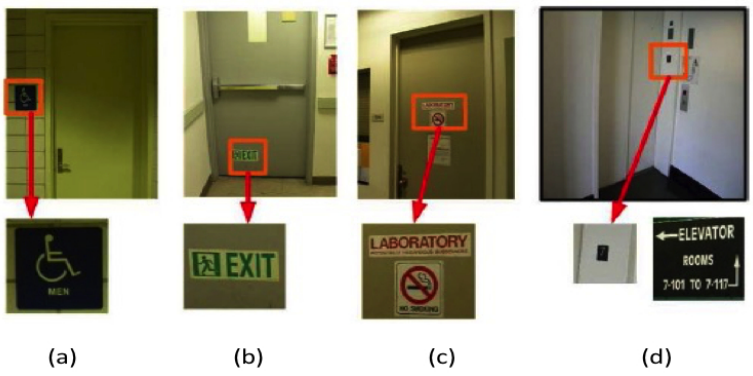


Fig. 14 Indoor objects (top row) and their associated text information (bottom row): (a) a bathroom, (b) an exit, (c) a lab room, (d) an elevator. Text information (bottom row) can be extracted to help blind or visually impaired persons find their ways [19].

should be easy to control. Based on our survey about this issue, most blind or visually impaired persons prefer button control rather than voice recognition, because the latter one is not reliable in noisy situations. Third, they should be able to capture relatively high-quality images or videos for information retrieval. Thus the camera focus should be adaptive to most indoor environments. It is difficult for blind or visually impaired persons to stand still for waiting for the calibration process because they cannot know the quality of the image. In addition, our group designed a method of selecting high-quality frames from blind captured videos [58].

The *process* component of a blind-assistant system is the process unit of the technical framework and algorithm implementation. Although it is not directly related to the user interface design, it is very important to reduce the computational complexity of the technical algorithms and optimize the codes to ensure the efficiency of the whole system for real-time processing as well as to reduce the power consumption of the processing unit. For example, the framework should be able to save and search

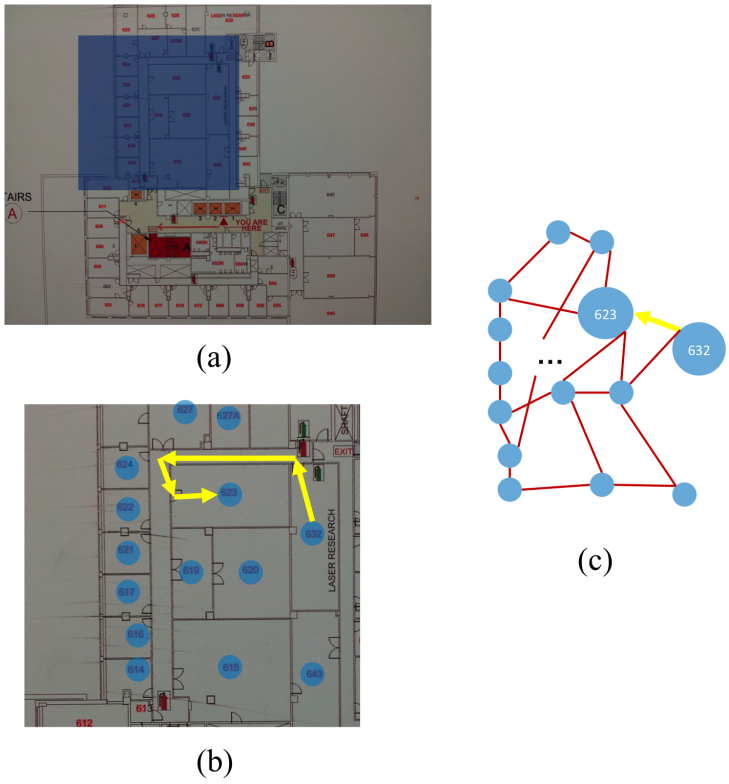


Fig. 15 (a) An example of a floor plan of our building, where the blue shaded region will be analyzed. (b) Each room number is regarded as a node, and the path from room 632 to room 623 is marked in yellow. (c) The abstract graph of the floor plan map, where the yellow line indicates a connection from node 632 to node 623, corresponding the yellow path in (b).

the historical data of blind-assistant recognition or navigation in specific buildings or scenes. Since a system belongs to one specific blind user, he/she would be able to directly obtain previous results when entering the same building or a scene again. Also the framework should be developed into a query-based system, rather than a notification-based system. It means that the system will not generate continuous notifications but keep sleep until the user wakes it. When the blind or visually impaired persons did not need the help of the system, the notifications would become useless noise.

About the *feedback* component of a blind-assistant system, firstly, the feedback should be simple, so that the users are able to obtain the most informative feedback within the shortest time. For example, in object recognition, it should adopt only two or three common-use words to describe an object or its main characteristic. Secondly, the feedback should be in time, that is, neither too early nor too late, so that the users are able to make decisions at the reasonable time window. Thirdly, the feedback should be understandable. For example, it is improper to say a door is located 3 meters in front with 10-degree deviation to the right, because the blind or visually impaired persons cannot measure the distance and orientations in an unfamiliar environment. It would be much better to navigation them to be close enough to the object and then tell them use their hands or white canes to touch it. The blind or visually impaired persons prefer speech communication with the system rather than the haptic feedback such as vibration. In addition, the feedback from users should be included in the *process* component.

6 Conclusions and Discussions

In this chapter, we focus on assistive text reading from natural scenes since text signage plays an important role in blind-assistant recognition and navigation applications. However, scene text extraction is still an open research topic to be addressed. It is a challenging task to extract text information from natural scene images for several reasons. Firstly, the frequency of occurrence of text information in natural scene image is usually very low, and text information is always buried under all kinds of non-text outliers in cluttered background of natural scenes. Thus background removal plays a significant role in text detection. Secondly, even though image regions containing text characters are detected from complex background, current optical character recognition (OCR) systems do not work well on the recognition of scene text, because they are mostly designed for scan documents. More effective feature representations and more robust models are required to improve the performance of scene text recognition. Unlike the text in scan documents, scene text usually appears in multiple colors, fonts, sizes and orientations.

In Section 2, we have reviewed several computer vision-based blind assistant applications, including the technical framework, user interface design, and prototype systems. We described a framework of scene text extraction in Section 3. First edge-based contour and MSER-based connected components are extracted as candidate

characters while removing large amount of non-text background outliers, and then text string alignment is applied to filter out the false positive candidate characters. Next, feature representations are designed to describe text structure, on the basis of gradient distribution, stroke width and orientation, edge density, and color uniformity, to remove false text strings. At last, feature representations are designed to recognize each text character of the text strings, on the basis of HOG descriptor.

The proposed framework of scene text extraction is involved in two blind-assistant applications in Section 4, hand-held object recognition and indoor navigation. In these applications, scene text extraction is transplanted into mobile platforms, and combined with other techniques. In a practical blind-assistant system, the user interface design is very important as well as the algorithm framework. In our design, a blind-assistant system consists of three components, which are capture, process and feedback. According to the survey of blind or visually impaired persons who are familiar with blind-assistant technology, we summarize the requirements of the three components respectively in Section 5.

In future, we will further improve the accuracy of scene text extraction algorithm, making it adaptive to more complex environments for more reliable practical application. Also we will make the algorithms more compatible with mobile applications. Furthermore, more interactions with blind or visually impaired people will be performed to better understand their requirements and design more robust and user friendly blind-assistant interface.

Acknowledgement

This work was supported in part by NSF grants EFRI-1137172, IIP-1343402, and FHWA grant DTFH61-12-H-00002.

References

1. 10 facts about blindness and visual impairment, World Health Organization: Blindness and visual impairment, 2009.
2. Advance Data Reports from the National Health Interview Survey, 2008. http://www.cdc.gov/nchs/nhis/nhis_ad.htm
3. D. Dakopoulos and N. G. Bourbakis, Wearable Obstacle Avoidance Electronic Travel Aids for Blind: A Survey. IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 40, 2535. 2010.
4. M. Bousbia-Salah, A. Redjati, M. Fezari, M. Bettayeb. An Ultrasonic Navigation System for Blind People, IEEE International Conference on Signal Processing and Communications (ICSPC), 2007; pp. 1003-1006.
5. G. Kao, FM sonar modeling for navigation, Technical Report, Department of Engineering Science, University of Oxford. 1996.
6. R. Kuc, A sonar aid to enhance spatial perception of the blind: engineering design and evaluation, IEEE Transactions on Biomedical Engineering, Vol. 49 (10), 2002; pp. 11731180.

7. B. Laurent, and T. Christian. A sonar system modeled after spatial hearing and echo locating bats for blind mobility aid, International Journal of Physical Sciences, Vol. 2 (4), April, 2007; pp. 104-111.
8. C. Morland, and D. Mountain. Design of a sonar system for visually impaired humans, The 14th International Conference on Auditory Display, June, 2008.
9. Seeing with Sound The vOICe: http//www.seeingwithdound.com/.
10. BrainPort lets you see with your tongue, might actually make it to market. http://www.engadget.com/2009/08/14/ brainport-lets-you-see-with-your-tongue-might-actually-make-it/.
11. D. R. Chebat, C. Rainville, R. Kupers, and M. Ptito, Tactilevisual acuity of the tongue in early blind individuals, NeuroReport, vol. 18, no. 18, pp. 19011904, Dec. 2007.
12. C. Yi and Y. Tian, Assistive Text Reading from Complex Background for Blind Persons, The 4th International Workshop on Camera-Based Document Analysis and Recognition (CBDAR), 2011.
13. S. Wang, C. Yi, and Y. Tian, Signage Detection and Recognition for Blind Per-sons to Access Unfamiliar Environments, Journal of Computer Vision and Image Processing, Vol. 2, No. 2, 2012.
14. C. Yi, Y. Tian, and A. Arditi, Portable Camera-based Assistive Text and Product Label Reading from Hand-held Objects for Blind Persons, IEEE/ASME Transactions on Mechatronics, Vol. 19, No. 3, pp808-817, June 2014. http://dx.doi.org/10.1109/TMECH.2013.2261083
15. Z. Ye, C. Yi and Y. Tian, Reading Labels of Cylinder Objects for Blind Persons, IEEE International Conference on Multimedia and Expo (ICME), 2013.
16. S. Yuan, Y. Tian, and A. Arditi, Clothing Matching for Visually Impaired Persons, Technology and Disability, Vol. 23, 2011.
17. F. Hasanuzzaman, X. Yang, and Y. Tian, Robust and Effective Component-based Banknote Recognition for the Blind, IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews, Jan. 2012.
18. X. Yang, S. Yuan, and Y. Tian, "Assistive Clothing Pattern Recognition for Visually Impaired People," IEEE Transactions on Human-Machine Systems, Vol. 44, No. 2, pp234-243, April, 2014.
19. Y. Tian, X. Yang, C. Yi, and A. Arditi. Toward a Computer Vision-based Wayfinding Aid for Blind Persons to Access Unfamiliar Indoor Environments, Machine Vision and Applications, 2012.
20. H. Pan, C. Yi and Y. Tian, A Primary Travelling Assistant System of Bus Detection and Recognition for Visually Impaired People, IEEE Workshop on Multimodal and Alternative Perception for Visually Impaired People (MAP4VIP), in conjunction with ICME 2013.
21. S. Joseph, X. Zhang, I. Dryanovski, J. Xiao, C. Yi, and Y. Tian, "Semantic Indoor Navigation with a Blind-user Oriented Augmented Reality, IEEE International Conference on Systems, Man, and Cybernetics, 2013.
22. S. Wang, H. Pan, C. Zhang, and Y. Tian, RGB-D Image-Based Detection of Stairs, Pedestrian Crosswalks and Traffic Signs, Journal of Visual Communication and Image Representation (JVCIR), Vol. 25, pp263-272, 2014. http://dx.doi.org/10.1016/j.jvcir.2013.11.005
23. A. Arditi and Y. Tian, User Interface Preferences in the Design of a Camera-Based Navigation and Wayfinding Aid, Journal of Visual Impairment and Blindness, Vol. 107, Number 2, pp118-129, March-April, 2013.
24. Z. Wang, H. Liu, X. Wang, Y. Qian, Segment and Label Indoor Scene Based on RGB-D for the Visually Impaired, Multimedia Modeling, Lecture Notes in Computer Science Volume 8325, pp 449-460, 2014.
25. Y. H. Lee and G. Medioni, A RGB-D camera Based Navigation for the Visually Impaired, RGB-D: Advanced Reasoning with Depth Camera Workshop, June 2011.
26. J. Canny, "A computational approach to edge detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, pp. 679-698.
27. J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in British Machine Vision Conference, 2002, pp. 384-396.

28. T. Kasar, J. Kumar, and A. Ramakrishnan, "Font and background color independent text binarization," in Camera-based Documentation Analysis and Recognition, 2007, pp. pp. 3-9.
29. L. Neumann and J. Matas, "A method for text localization and detection," in Asian Conference on Computer Vision, 2010.
30. C. Yi and Y. Tian, Text String Detection from Natural Scenes by Structure-based Partition and Grouping, IEEE Transactions on Image Processing, Vol. 20, Issue 9, 2011.
31. C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting text of arbitrary orientations in natural images," in IEEE Conference on Computer Vision and Pattern Recognition, 2012.
32. P. Viola and M. Jones, "Robust real-time face detection," International Journal of Computer Vision, 2004.
33. Nuance. Nuance Omnipage. http://www.nuance.com/for-business/by-product/omnipage/index.htm
34. Abbyy. http://finereader.abbyy.com/
35. R. Smith, "An overview of the Tesseract OCR engine," in International Conference on Document Analysis and Recognition, 2007.
36. N. Dalal and B. Triggs, "Histogram of Oriented Gradients for Human Detection," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005.
37. C. Yi, X. Yang and Y. Tian, Feature Representations for Scene Text Character Recognition: A Comparative Study. International Conference on Document Analysis and Recognition, 2013.
38. Christopher J. C. Burges, "A tutorial on support vector machine for pattern recognition," Data Mining and Knowledge Discovery, pp. 121-167, 1998.
39. C. Yi and Y. Tian, Scene Text Recognition in Mobile Applications by Character Descriptor and Structure Configuration. IEEE Transactions on Image Processing, Vol. 23, No. 7, pp2972-2982, July 2014.
40. X. Yang and Y. Tian, "Robust Door Detection in Unfamiliar Environments by Combining Edge and Corner Features," in IEEE Conference on Computer Vision and Pattern Recognition Workshop on Computer Vision Applications for Visual Impaired, 2010.
41. T. de-Campos, B. Babu, and M. Varma, "Character recognition in natural images," in International Conference on Computer Vision Theory and Applications, 2009.
42. Y. Tian, A. Senior, and M. Lu, Robust and Efficient Foreground Analysis in Complex Surveillance Videos, Machine Vision and Applications, Volume 23, Issue 5, Page 967-983, 2012.
43. C. Yi, R. Flores, R. Chincha, and Y. Tian, Finding Objects for Assisting Blind People, Network Modeling Analysis in Health Informatics and Bioinformatics, July 2013, Volume 2, Issue 2, pp 71-79.
44. B. Schauerte, M. Martinez, A. Constantinescu, and R. Stiefelhagen. An assistive vision system for the blind that helps find lost things. In ICCHP, 2012
45. S. Caperna, C. Cheng, et al. A navigation and object location device for the blind. In Techical Report, University of Maryland, 2009.
46. A. Hub, J. Diepstraten, T. Ertl. Design and development of an indoor navigation and object identification system for the blind. In Proc. ACM SIGACCESS Conf. Computer and Accessibility, 2004.
47. J. Bigham, C. Jayant, A. Miller, B. White and T. Yeh. VizWiz: LocateIt enabling blind people to locate objects in their environments. In Proc. CVPR Workshop Computer vision applications for the visually impaired, 2010.
48. R. Velazquez. Wearble assistive devices for the blind. Chapter 17 in A. Lay-Ekuakille and S.C. Mukhopadhyay (Eds.), Wearable and Autonomous Biomedical Devices and Systems for Smart Envrionment: Issues and Characterization, LNEE 75, Springer, pp. 331-349, 2010.
49. B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in Natural scene with stroke width transform," in IEEE Conference on Computer Vision and Pattern Recognition, 2010.
50. N. Nikolaou and N. Papamarkos, "Color reduction for complex document images," International Journal of Imaging Systems and Technology, vol. 19, pp. 14-26, 2009.
51. T. Phan, P. Shivakumara, and C. Tan, "A laplacian method for video text detection," in International Conference on Document Analysis and Recognition, 2009, pp. 66-70.

52. C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang. "Scene text recognition using part-based tree-structured character detection," in IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2961-2968.
53. J. Weinman, E. Learned-Miller, and A. Hanson, "Scene text recognition using similarity and a lexicon with sparse belief propagation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 10, pp. 1733-1746, 2009.
54. A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," in IEEE Conference on Computer Vision and Pattern Recognition, 2011.
55. Google glass (2014) https://www.google.com/glass/start
56. S. Mann, J. Huang, R. Janzen, R. Lo, V. Rampersad, A. Chen, and T. Doha. Blind naviation with a wearable range camera and vibrotactile helmet. In ACM-MM, 2011.
57. W. Khoo, J. Knapp, F. Palmer, T. Ro, and Z. Zhu. Designing and testing wearable range-vibrotactile devices. In Journal of Assistive Technologies, 2013.
58. L. Tian, C. Yi, and Y. Tian, "Detecting good quality frames in videos captured by a wearable camera for blind navigation," in IEEE Conference on Bioinformatics and Biomedicine, 2013, pp. 334-337.
59. Autographer. http://www.autographer.com/#home
60. MeCam. http://www.mecam.me/
61. Looxcie. http://www.looxcie.com/
62. GoPro. http://www.gopro.com