Robotica (2016) volume 34, pp. 1113–1127. © Cambridge University Press 2014
doi:10.1017/S0263574714002094

Use of character information by autonomous robots based on character string detection in daily environments

Kimitoshi Yamazaki†*, Tomohiro Nishino‡, Kotaro Nagahama‡, Kei Okada‡ and Masayuki Inaba‡

†*Faculty of Engineering, Shinshu University, 4-17-1 Wakasato, Nagano, Nagano, 380-8553, Japan*
‡*Graduate School of Systems and Information Engineering, The University of Tokyo, 7-3–1 Hongo, Bunkyo-ku, Tokyo, 305-8573, Japan*

(Accepted July 11, 2014. First published online: August 15, 2014)

SUMMARY
Characters encountered in daily environments provide valuable information to human beings and potentially to automated machines. This paper describes the use of character information by robots working in daily environments. Our robot system finds and reads character strings by virtue of a recognition module that detects character candidates as closed contours in an image. The closed-contour-based method enables detection of various character strings observed from different viewpoints. To cope with tiny and distant characters, the image processing is collaborated with a camera system having a mechanical gaze adjustment. By combining the detection result with an optical character reader, the autonomous robot can provide daily assistance to humans.

KEYWORDS: Character string detection; Closed contour; Daily assistive robots.

1. Introduction

Objects on which characters are written are commonly encountered in our daily environments and provide considerable information to their viewers. Indeed, most of our daily behaviors is influenced by text. Moreover, autonomous robots working in daily environments extract text character information to perform meaningful work. Therefore, we propose a robot system based on a character detection system. The method is collaborated with character reading techniques, and is tested on robotic applications in real environments.

Detecting real-world characters is important for understanding scenes. The use of images as sensor data has been a research focus for several decades and has been aided by the publication datasets such as ICDAR2011. The results from some of these studies have been combined with an optical character reader (OCR) to provide convenient services such as business card readers.

To implement a character detection method suitable for robotic applications, we must resolve several problems inherent in the living environment. One difficulty is that in real situations, characters are captured from various viewpoints. That is, characters may be diminished and skewed in an image, requiring a framework that can interpret differently viewed characters.

We applied a method of character string detection based on closed contours, which permits a certain level of sloping and skewing of characters. the method is supported by a camera system equipped with convergence control and an electrical zoom lens. Collectively, image processing and gazing control enable the robot to detect various character strings in its real world environment. In this paper, we combine with OCR software, to realize practical robotic applications such as checking freshness dates and separating plastic bottles from other trash.

The reminder of this paper is organized as follows: Section 2 identifies the relevant issues and introduces our approach. Section 3 presents our method of character string detection, and Section 4

* Corresponding author. E-mail: kyamazaki@shinshu-u.ac.jp

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

discusses the interpretation of character strings. Experimental results, and a general discussion are presented in Sections 5 and 6, respectively. Conclusions are presented in Section 7.

2. Issues and Approach

2.1. Extracting character information in daily environments

In daily environments, character string detection faces two major issues:

1. **Detection of sloping character strings:** Because a robot can view text from various perspectives, character strings observed by the robot are not necessarily perfectly aligned, but are likely distorted by rotation along the optical axis and perspective skewing. Not only rotation along optical axis but also perspective skew may occur. In general, rotated and skewed characters are invalid inputs to the OCR software.
2. **Detection of characters captured at a small size:** Small characters are often visualized in daily situation. However, character detection processes cannot detect characters below a certain size. Since small characters are excluded from the candidate pool, most of the characters information is lost from the image.

To input the detected information to an interpretation process, a horizontally-aligned character string must be given to the OCR software. Therefore, the character string detection process must be able to tolerate the abovementioned input conditions. A robust interpretation process is also needed.

2.2. Related work

2.2.1. Robotic applications using character information. Several mobile robotics studies have reported the use of meaningful sign information.[8,10,29] The system of Miura *et al.*[18] interpret traffic signs. Their image-based recognition method is based on the built-in functions of a compassable image processing board and its effectiveness has been validated in on-road experiments. Klingbeil *et al.*[13] designed a system that detects, localizes, and labels elevator buttons. Their method is based on machine learning of contextual features. By accessing a dataset of 150 elevator panels photographed from more than 60 elevators, this system localized and labeled 86.2% of the test buttons.

However, these previous researches extract sign and character information from easily-indentifiable regions. For example, traffic signs are exclusively erected at roadsides, and elevator buttons are embedded in elevator panels.

2.2.2. Character string detection. Other studies have focused on detecting character strings from natural scene images.[14,21,31] Character string detection algorithms can be broadly classified into three types:[14] gradient-based methods,[2,12,25] color segmentation based methods[7,15] and texture-based methods.[4,16]

In our assumed environments, text characters are found on diverse objects such as books, food packages, and plastic bottles. One difficulty is interpreting the many different methods of viewing these characters. Even a straight character string may appear rotated and skewed by the viewing perspective. These situations have been ignored in studies using the horizontally aligned character strings in the ICDAR dataset images.[31–33]

To correct rotation and skew, a system could first detect the background of printed characters.[1] For example, Kabata *et al.*[11] targeted characters on panels. A similar approach was used to identify roadside traffic signs.[26] However, the background detection approaches is inapplicable for our purpose, which is to detect character strings printed on differently manipulated targets.

2.3. Our approach

As mentioned in Section 2.2.1, almost all robotic applications relevant to character interpretation are limiting to mobile robots. Hence, the techniques required for text detection in ordinary object manipulations have not been properly clarified and validated. Because our prerequisite for character reading is more difficult than that in previous robotic studies, we require more general methods. Some specialized character string detection method proposed by previous studies have been mentioned in Section 2.2.2.

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

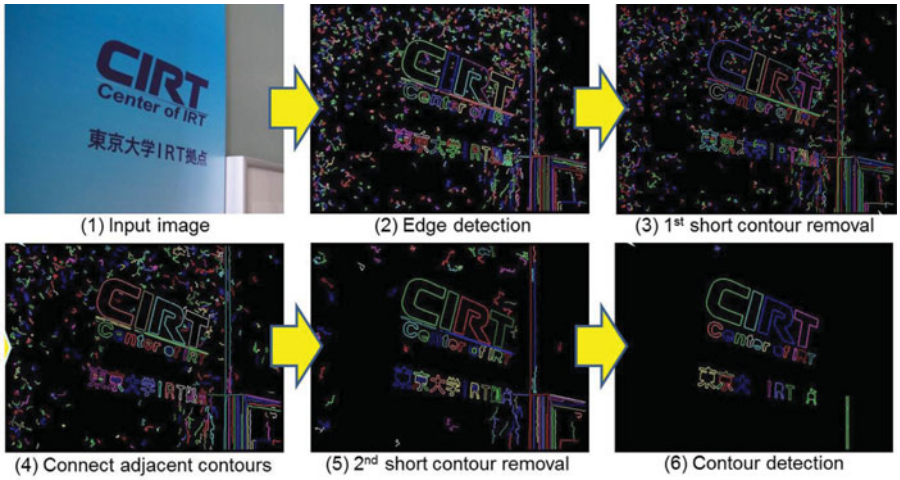


Fig. 1. Closed contour detection. Different contours are represented by different edge colors.

On the other hand, when we aim to develop an assistant robot working in domestic environment, one important point is not only to propose a method in individual technological elements, but also to find good combination of them. A well-designed hardware do not need high performance to a character detection software from an image. In this study, we captured high-resolution character images using a stereo camera with an electric zoom lens.

To achieve our study's aims, we selected and improved an image-based closed-contour method proposed by Takahashi *et al.*[28] In this method, a character string consists of characters located on a straight line. Therefore any region containing regularly arranged closed contours is expected to contain text information. We extend this method to extract an arbitrarily rotated or skewed character string from an image. However, since this approach cannot resolve the closed contours of small characters in an image, it is restricted to scenes with large character, and is therefore insufficient for our target environment. Therefore, we magnify images containing small text by a mechanical gaze adjustment system and image frequency filtering.

Having detected a group of characters, the robot system must interpret them to achieve a given task. However, if our detection method compensates for viewpoint when preparing a character string, it may introduce quantization, rotation, and skew errors. To prevent these errors from causing interpretation failure, we implemented three procedures: (i) providing the robot with words related to a given task, (ii) assuming that a detected character string may have multiple interpretation, and (iii) evaluating the OCR results by a word similarity measure. These procedures are detailed in Section 5.

3. Character String Detection Based on Closed Contours

3.1. Closed contour extraction

Closed contours must be extracted from all existing characters, regardless of whether they are horizontally aligned or rotated and skewed. Our detection method extends the method of Takahashi *et al.*[28] and comprises edge detection, contour detection, and improvement processes (see Fig. 1). First, canny edge detection is applied to an input image, and the edge segments are extracted (Fig. 1, (2)). Short edge segments with lengths less than θ_{l1} are removed (Fig. 1, (3)). On the other hand, disconnected edges are common in detected regions of steeply changing direction. Therefore, an additional connection manipulation between neighboring edges is performed. The endpoints of two edges are connected if they are separated by less than θ_{cn} (Fig. 1, (4)). For example, at stage (3) of the detection, the 'R' in 'CIRT' has been divided into two parts. The disjointed parts were connected and formed a single contour through the re-connection process. Next, short segments of length below θ_{l2} are removed (Fig. 1, (5)). The remaining edges are selected as closed edges if their end-points are separated by less than θ_{cl} (Fig. 1, (6)). The extraction procedure also removes closed contours within other closed contours. However, these removed contours are registered in relation to its external contour.

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

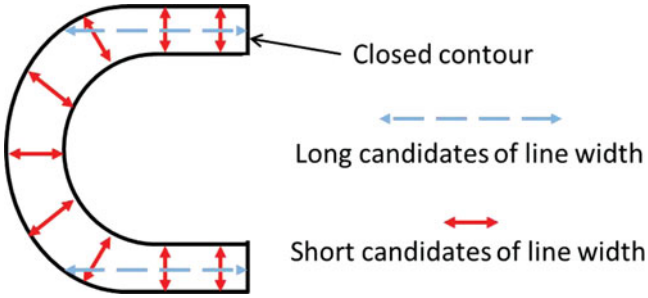


Fig. 2. Candidates of line width.

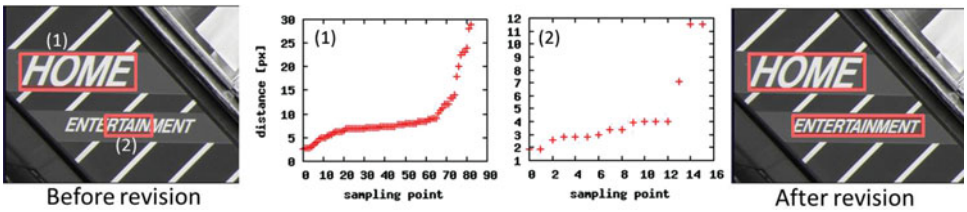


Fig. 3. Left panel: Originally detected character string, central panels: sorted candidates of characters' thickness versus position in the character string; right panel: revised character strings. The region of the lower string is now properly expanded.

3.2. Extraction of character regions

If the characters included in a character string are horizontally aligned, string regions can be extracted by detecting horizontally aligned closed contours.

First, a rectangle circumscribing the closed contours is calculated. This rectangle is slid to the right by its own width in pixels. If the rectangle encloses the center of the circumscribed rectangle of another closed contour, these two closed contours are assumed to be horizontally aligned and included in the same character string. Isolated closed contours are assumed not to be characters. Circumscribed rectangles of each character string are assumed to be string regions. This process extracts the string regions of horizontally aligned closed contours.

3.3. Revision of character string regions based on thickness of characters

Since the proposed method extracts the closed contours of characters, it can calculate the thickness of the characters included in each character string region. To this end, some points on closed contours are first sampled. Next, lines passing through the sampled points and perpendicular to the contours are calculated. The intersection points of these lines and the closed contours are also calculated. The distance between a sampled point and an intersection point is a candidates estimate of the character's thickness. Finally, all candidates of each character string region are sorted.

As shown in Fig. 2, the candidates thickness vary within a single closed contour. However, in most cases, character thickness is determined from the points sampled on the longer side. Therefore, a good estimate of the character thickness is the middle value of the sorted candidates in given character string region.

Once the thickness of characters has been computed, it is used to revise the character string regions. Both closed and unclosed contours lying to the left and right of the character string regions are extracted, and their thickness are calculated. These contours are added to the character string region if their thicknesses are similar to those of other characters in that region.

The graphs in the panels of Fig. 3 plot the sorted candidates of characters' thickness against the horizontal character string. The gradient of the sorted candidates is small around the string center; hence, the character thickness is properly selected within this region. The right panel of Fig. 3 shows the result of revising the character string region. This process correctly includes all closed and unclosed contours in the character string region.

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

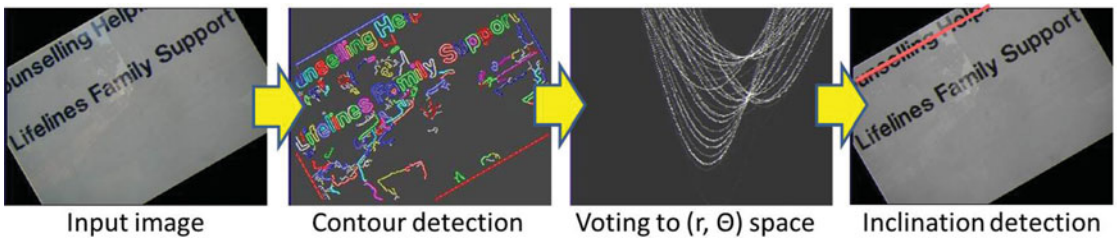


Fig. 4. Inclination estimation based on the Hough transform.

3.4. Inclined string detection based on the Hough Transform[22]

The above-described method assumes that a character string is captured along a horizontal line. The character detection likely fails when the character string is inclined. However, character strings are frequently inclined in real text or the objects containing text are oriented at arbitrary direction with respect to the camera. To accommodate such variability, we estimate the inclination angle from a detected line of characters. This method is an application of the Hough transform. Simple geometrical feature such as line segments and circles are identified by the common Hough transform. Because a character string is assumed to comprise characters arranged in a line, it can be detected by the same principle. The method detects line passing through the centers of contours in a character string.

A group of straight lines intersecting at a point can be represented as a sine curve in (r, θ) space. Now we consider two points A and B in (x, y) space. If a sine curve in (r, θ) space, representing the group of straight lines that cross point A in (x, y) space, intersects with a sine curve derived from point B, the crossing point indicates a line connecting point A and B. In our case, the center of a closed contour is regarded as a point, and each point draws a sine curve in (r, θ) space. The (r, θ) space is divided into lattice arrangement, and grid units containing sine curves acquire a vote. If a certain grid unit collects many votes, a reliable straight line exists in the corresponding (x, y) space. The voting process, imposes a weighting factor based on the completeness of the closed contour; that is, the distance between both end-points of the contour. If the contour is entirely closed, the weighting factor is large. Otherwise, the vote value is modified the distance d between the tips of two edges.

$$val = \frac{1}{d^2 + 1} \tag{1}$$

This line detection enables character strings to be identified. In our implementation, the width of each grid unit in the (r, θ) space is $\pi/60$ radians. The lattice covers an area of $-\pi \leq \theta \leq \pi$.

Figure 4 shows one result of the inclination estimation. First, closed contours are extracted from the input image, and voted to the (r, θ) space on their positions (x, y). The red line in the rightmost panel is the line generated from the voting result. Thus, we can calculate the line passing through the characters in a character string. Since the slope of this line should be equal to that of the character string, the inclined character string can be extracted by detecting the neighboring closed contours along the sloping angle. Examples of character string detection are show in Fig. 5. Black characters were printed on a white plate, and the plate was rotated. In the upper panels, the block was rotated (from left to right) by 40 [deg], 50 [deg], 60 [deg] along the horizontal axis; the central panels show rotations by 50 [deg], 60 [deg], 70 [deg] about the vertical axis; and the bottom panels show 30 [deg], 60 [deg], 90 [deg] rotations along the optical axis. The proposed inclination method accurately detected large characters with resolvable closed contours.

3.5. Image frequency filtering for detecting character candidates

The above character string detection is not suitable for detecting tiny characters whose contours are difficult to obtain. However, such characters should not be ignored. Instead, we extract them by applying a spatial frequency filter This filter is inspired by Ryu's approach,[27] which uses the Fourier transform.

Noting that tiny characters in images aggregate in the high frequency region, we apply a two dimensional discrete Fourier transform. Let $I(x, y)$ be an image of width w and height h. The Fourier

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

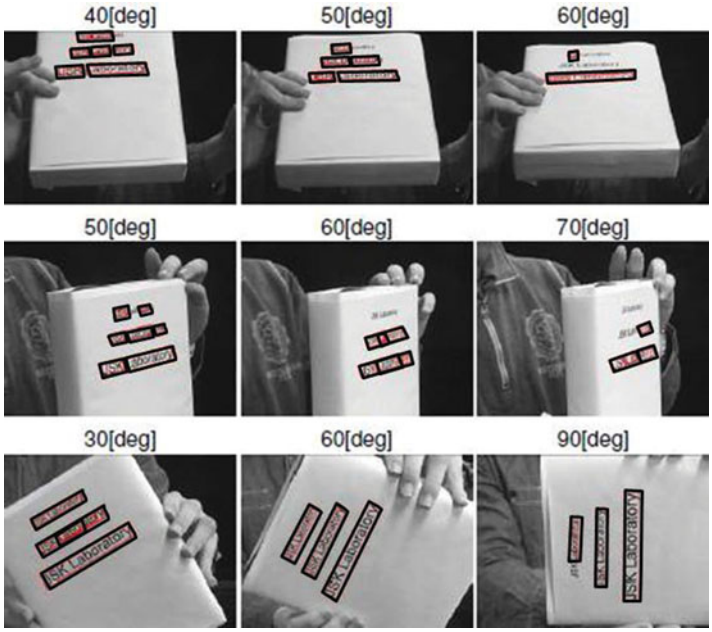


Fig. 5. Results of inclination estimation.

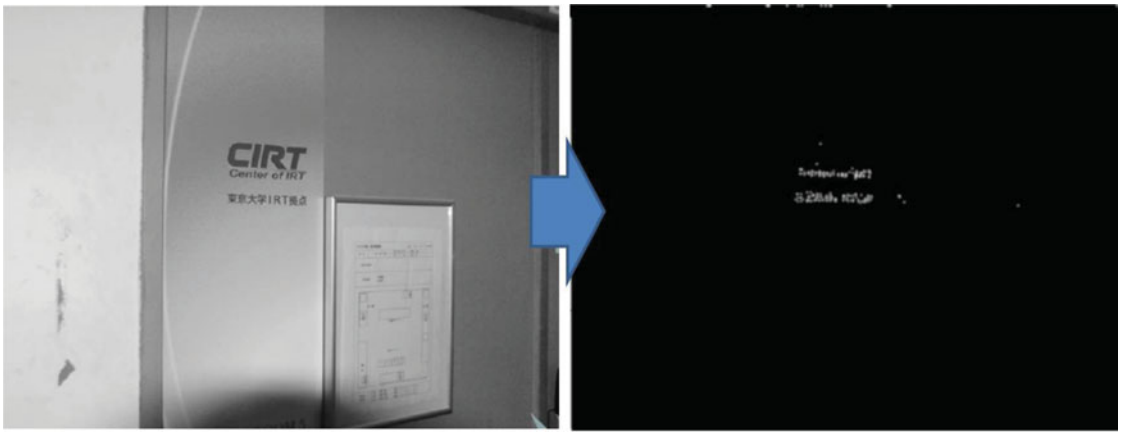


Fig. 6. Detection of character candidates by image frequency filtering. The text region is highlighted by groups of white pixels.

transform of the image is given by

$$F(u, v) = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I(x, y) exp(-2\pi j(uxw + vyh)). \tag{2}$$

Applying the two-dimensional discrete Fourier transform to $\hat{F}(u, v)$, the particular frequencies extracted from $F(u, v)$, we obtain the spatial frequency filter

$$\hat{I}(x, y) = \frac{1}{wh} \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} \hat{F}(u, v) exp(2\pi(uxw + vyh)). \tag{3}$$

An example of the filtering is shown in Fig. 6. These results are considered as the character candidates, and are targeted in close-up captures by the gaze adjustment vision system, described in Section 5.

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

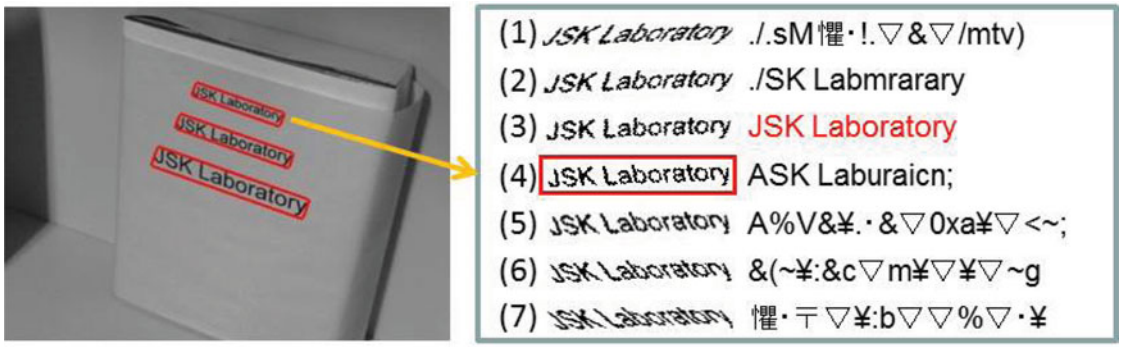


Fig. 7. A set of affine transformations and read results of a character string.

4. Interpretation of a Character String

4.1. Issues on character reading and our approach

The character string detection method presented in Section 3 outputs a set of horizontally-aligned character strings; thus, these strings satisfy the requirements of common OCR.

However, merely joining the strings is insufficient for the following reasons.

- If the aligned character string resides in a skewed image region, the potential for large quantization error is high.
- If an image rotation angle estimated by the Hough transformation contains even a small error, the success rate of character reading by OCR decreases.

To overcome these issues, we interpret the character information as outlined below:

- The robot is provided with several keywords directly related to its target task in advance. For example, when assigning a bottle-trashing task to the robot, we supply relevant words such as "plastic-bottle" and "PET".
- Character reading errors are avoided by two countermeasures; one is applied before OCR, whereas the other is applied after OCR. The first countermeasure adds multi-hypothesized image skews into the target character string region, and the second permits an incomplete character string in the word interpretation method.

These approaches reduce the failure of character string interpretation. The second item is detailed in the next subsection.

4.2. Robust character string interpretation

Although the proposed character string detection compensates for inclination, the inclination may not be completely modified. In addition, the interpretation may fail because of quantization error. Therefore, dozens of affine transformations are added to the original character string, and the most appropriate interpretation is finally adopted.

An example is shown in Fig. 7. The character string numbered (4) is the originally detected. Character strings shown in right column show OCR results. In string (4), although the inclination is almost modified by the detection method, the OCR output contains misreading errors. On the other hand, one of six character strings that were slightly skewed by affine transformation outputs a correct result. If the character string matches a registered character string during a given task, the robot can use that result to plan its next behavior and eventually complete the task.

The above countermeasures do not eliminate character interpretation errors. To cope with such error, we impose a character string similarity criterion called BLEU[24] in the evaluation procedure. BLEU is based on n-gram correspondence, where their n-gram representation divides a word into n series of groups. BLEU estimates the word immediately succeeding the present word. It has been used for the estimation of a word just after a present word, and so on.

For example, the word "image" is divided into the following n-grams:

- 1-gram : i,m,a,g,e,

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

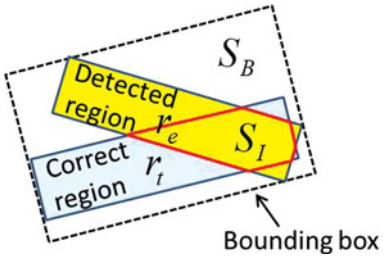


Fig. 8. Method of evaluating the accuracy of character string detection.

- 2-gram : im,ma,ag,ge,
- 3-gram : ima,mag,age.

The accuracy p that specifies the similarity between two character strings S and S_r is calculated from the number of n-grams. For the two strings $S =$ "image" and $S_r =$ "magic," the accuracy values are given by:

- p1 = 4/5,
- p2 = 2/4 = 1/2,
- p3 = 1/3.

Because the similarity value are higher for short words than that for long words, we impose a penalty BP, defined as

$$BP = \left\{ \begin{array}{ll} 1 & (c > r) \\ e^{1-r/c} & (c \leq r) \end{array} \right\}. \tag{4}$$

where c and r are the lengths of the input and reference character strings, respectively. The BLEU is then calculated as follows:

$$BLEU = BP exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{5}$$

where w_n is the weight of each n, here set to $w_n = 1/N$. To ensure a reliable result, N should be large. In our experiments, we selected the maximum N that unsure nonzero P_n.

5. Evaluation and Experiments

5.1. Performance evaluation of character string detection using ICDAR dataset

The proposed method of character string detection was evaluated on the dataset used for the ICDAR 2003 robust reading competitions.[36] This dataset continues to be used as an evaluation index in character detection studies.[9,17,23] We resized all images to maintain their aspect ratio within a resolution of 320 × 240, although many images in database have higher resolution. The evaluation method was the same as that in the Text Locating Competition at ICDAR 2003 and 2005[31,32] and is schematized in Fig. 8. The size of the overlapped area $S_B(r_e, r_t)$ between the detected bounding box r_e and the correct bounding box r_t is defined as follows:

$$\begin{aligned} m_p(r_e, r_t) &= \frac{S_I(r_e, r_t)}{S_B(r_e, r_t)} \\ m(r, R) &= \max m_p(r, r') | r' \in R. \end{aligned} \tag{6}$$

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

Table I. Character detection results with ICDAR image dataset.

	Precision	Recall	f-measure
TrialTrain	0.478194	0.385912	0.403682
TrialTest	0.373111	0.308109	0.321485



Fig. 9. Examples of character string detection in ICDAR2003 dataset.

From these, we can calculate the precision p, recall r, and f-measure f as follows:

$$p = \frac{\Sigma_{r_e \in E} m(r_e, T)}{|E|},$$
$$r = \frac{\Sigma_{r_t \in E} m(r_t, T)}{|T|}, \tag{7}$$
$$f = \frac{2pr}{p + r},$$

where E and T denote a set of detected and correct characters, respectively. In this experiment, α was set to 0.5.

Table I shows evaluation result of character string detection using the ICDAR 2003 dataset. The dataset was divided into 250 images for training, and 249 images for testing. Some detection examples are shown in Fig. 9. In some of the ICDAR images, the character strings were partially obscured by illumination reflectance. In some cases, the correct contour was difficult to detect because of the bright background region. However, our method, which uses the thickness of characters as a compensatory measure, as explained in Section 3.3, yielded feasible results. Our closed contour-based method could identify both handwriting and various printed font such as bold and mincho.

5.2. Performance evaluation using natural images

The character strings in ICDAR dataset images are all horizontally aligned. However, robots working in real environments should detect and read inclined and skewed character strings viewed from different angles. To investigate whether our method can evaluate such strings, we collected 107 images of our laboratory. After identifying the closed contours, character candidates were detected and character strings were generated. The evaluation method is described in ref. [32]. The following result obtained: $precision = 0.44$, $recall = 0.46$, $f - measure = 0.43$. As shown in Fig. 10, various character strings printed on objects were properly detected despite the inclination and skew of the captured images.

5.3. Experimental camera system with mechanical gaze adjustment

Our camera system[19] is photographed and schematized in Fig. 11. The stereo camera comprises two FCB-IX11A camera modules made by SONY Co., Ltd. The modules are installed with electric

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press



Fig. 10. Examples of character string detection on office environments.

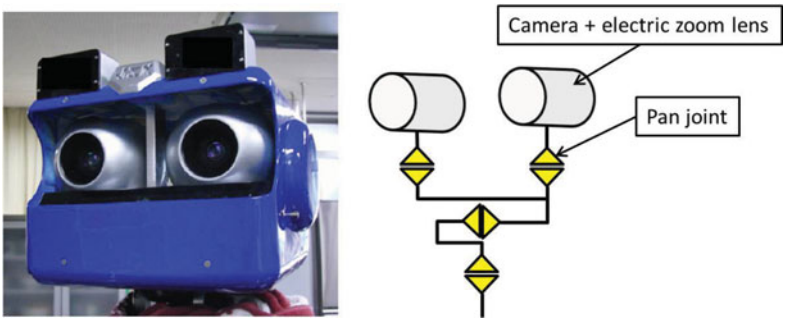


Fig. 11. Mechanism of gaze adjustment camera system.

zoom lenses that change the optical zoom factor from 1.0 to 9.5 times. Combined with a 0.7× wide conversion lens (VCB-HG0730X; SONY Co., Ltd), their view angle is controllable from 65.7 to 6.9 [deg]. For VGA-sized images (640 × 480 pixels) and maximum zoom factor (9.5 ×), resolution of approximately 20/13 vision (USA unit) is attained. Each zoom camera can independently move around its pan joint. The maximum accuracy of the joint angle of the pan axis is 0.00125 [deg/count], sufficiently smaller than the view angle per pixel at the maximum zoom factor, which is 6.9/640 = 0.01078 [deg/pixel] at VGA resolution.

For our purposes, the images captured by the camera system should suit the immediate requirements. Essentially, identifying character string candidates require a wide-angle view and an enlarged view is needed for verifying each candidate. In the former case, the 3D positions of the candidates are useful for detecting the viewpoint of the succeeding enlarged images. For this purpose, we engage two cameras to realize convergence eye movement, and capture target characters in the centers of the enlarged images. The size is adjusted as required by the electric zoom lenses.

Once the 3D positions of character candidates are obtained, the image can be enlarged. Enlargement requires both the direction of the camera and the zoom ratio. To determine the next zoom ratio, we denote the present zoom ratio by z, and the width and height of an image by w_i and h_i, respectively. Denoting the width and height of a bounding box enclosing the target character string w_b and h_b, respectively, the next zoom ratio $z_0$is calculated as

$$z_0 = z \times min\left(\frac{w_i}{w_o}, \frac{h_i}{h_o}\right). \tag{8}$$

5.4. Settings for robotic applications

The frequency value of the filter described in Section 3.5 was set within the range $100/640 \leq \sqrt{x^2 + (y \times 480/640)^2} \leq 250/640$, where x [cyc/px] and y are the horizontal and vertical frequency,

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

respectively. The size of the Gaussian kernel for canny edge detection was 1.0, and the hysteresis thresholds were set to 0.5 and 0.9.

After detection of closed contours, whether each contour was included in a character region was determined by the following conditions:

$$1/1.3 < W_c/W_s < 1.3, \tag{9}$$

$$|y_s - y_c| < h_s/3, \tag{10}$$

$$0 < x_{sl} - x_c < 2h_s \quad or \quad 0 < x_c - x_{sr} < 2h_s, \tag{11}$$

$$h_c/h_s < 1.1, \tag{12}$$

where x_{sl} and x_{sr} are the leftmost and rightmost x coordinates of the character region, respectively. y_s is a y coordinate and h_s is the height. W_s is the estimated width of the character region. The central positions of the contour, height, and width are denoted x_c, y_c, h_c and W_c, respectively.

Two closely spaced character regions r_i and r_j are merged if they satisfy both of the following conditions.

Condition I:

$$S_{I_{ij}}/S_{r_i} >= 0.5, \tag{13}$$

$$S_{I_{ij}}/S_{r_j} >= 0.5. \tag{14}$$

Condition II:

$$y_{0i} <= y_{cj} <= y_{1i}, \tag{15}$$

$$y_{0j} <= y_{ci} <= y_{1j}, \tag{16}$$

$$|x_{ci} - x_{cj}| - (w_i + w_j)/2 < W_I/10, \tag{17}$$

$$0.7 < (y_{1i} - y_{0i})/(y_{1j} - y_{0j}) < 1/0.7. \tag{18}$$

Here, S_{r_k} is the area of r_k, and $S_{I_{ij}}$ is the area overlapped by r_i and r_j. y_{0k} is the top y coordinate of the r_k, y_{ck} denotes the center, and y_{1k} is the bottom coordinate. w_k and W_I are the widths of the line of the character line and image, respectively.

Character string interpretation was enabled by connecting tesseract-ocr[34] and NHocr[35] to our character detection software. These OCR engines recognize English and Japanese text, respectively.

5.5. Application 1: Freshness date confirmation

One piece of important information provided by characters is the freshness date. As an application of our method, we combined reading this information with a robot's manipulation skill.

In this experiment, a food package was grasped by a robot. The robot first sought the freshness date on the top surface of the package (see upper left panel of Fig. 12). Detection was implemented in wide-angle view, and the detected region was enlarged by the zooming function and input to the OCR. The character string denoting "freshness date" in Japanese was pre-supplied, and was compared with the OCR results using BLEU. The character string with the highest similarity to the known string was selected. Next, assuming that the specific date would be displayed near the freshness date, a number character string was searched and decoded.

Examples of the detection are shown in the right panels of Fig. 12. Because this experiment was counted in January 2012, the robot automatically judged that 2012.11 presented no problem but that 2011.11 exceeded the data and the food is not fit for consumption.

5.6. Application 2: Separation of trash

To properly dispose garbage, we need to locate the proper trash can. In many cases, we can get the information by sign of character strings printed on trash boxes.

As a second application that of our method, we directed the robot to find a proper trash can and dispose garbage into it. As shown in Fig. 14, there were two trash cans in our laboratory, one labeled

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

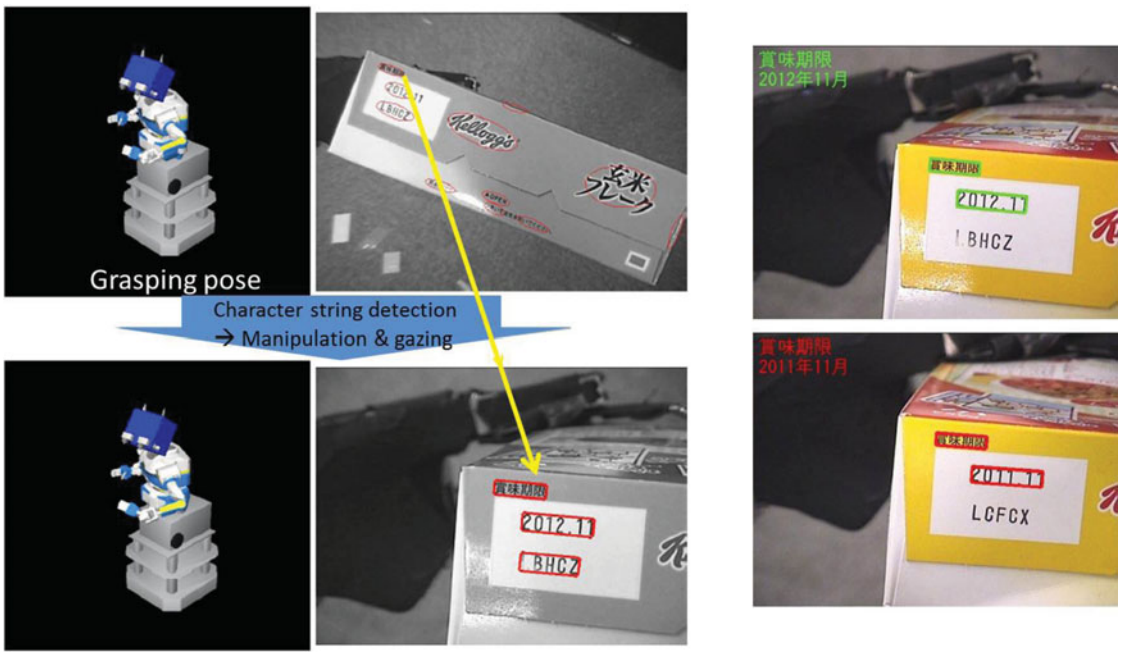


Fig. 12. Freshness date detection and reading.

"combustible material," the other labeled "plastic bottle." Text was displayed in both Japanese and English. A plastic bottle was placed on a table near the start position of the robot. The robot was instructed to find and grasp the plastic bottle, identify and read the character strings, and deposit the bottle into the correct trash can.

This application was rendered difficult by the limited viewpoint of the robot. Because the text was printed at a relatively low height on the trash cans, the strings in the images captured by the robot were significantly skewed by the perspective. This experiment demonstrated the usefulness of out method described in Section 3. As shown in the central image of Fig. 13, the robot first detected the candidate character strings in wide-angle view. Next, the robot acquired a high resolution image and focused on the candidate strings. From this image, the character string was detected, aligned, and input to OCR. The OCR output was used to select the correct trash can.

Although problems might arise from the false reading results in the OCR output, there were alleviated by BLEU. The robot disposed the plastic bottle in the correct trash can. A time sequence of the experiment is shown in Fig. 14.

6. Discussion

This study demonstrated the use of character information by an autonomous robot. In real robot applications (Sections 5.5 and 5.6), conventional character detection techniques alone are insufficient. Additional techniques are needed to implement the unique situations encountered by life-sized robots. The difficulty of character detection and reading can be alleviated by purpose-built robot hardware. Here, we succeeded in combining hardware and software modules into highly functional unified system. On the other hand, the performance of each module requires improvement, especially in character string detection. In this section, we discuss future developments and related work that may assist in improving our system.

Because our character string detection is based on closed contours, we could construct a robust algorithm that calculate the thickness of characters from two parallel edges extracted by a canny operator. A similar approach was adopted by Epsthein *et al.*,[6] who proposed the stroke width transform. Epsthein's method has been extended and used in state-of-the-art character detection methods.[3,20] Although these methods can potentially process more character types than our closed contour based method they have been validated only on ICDAR 2005 and 2011 datasets,[32,33] which exclude largely inclined characters. We expect that our closed contour method could be successfully

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

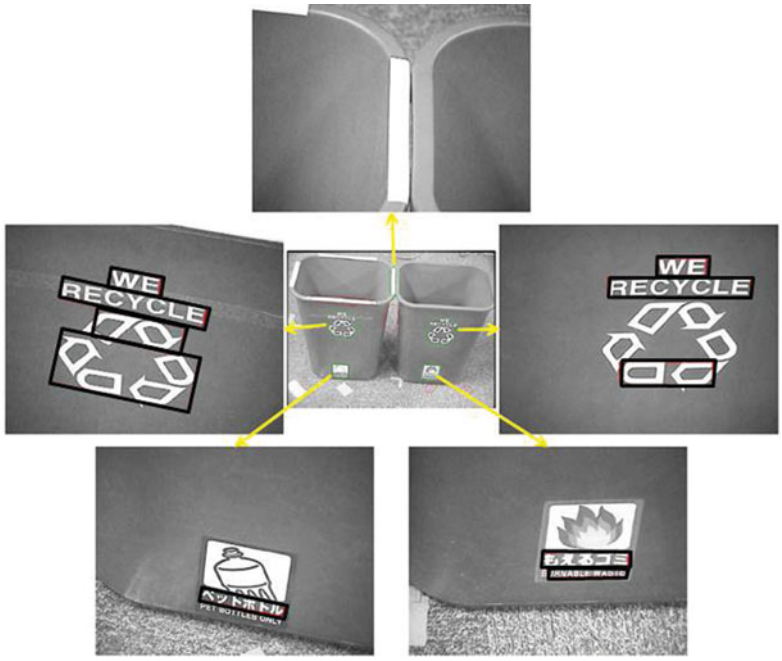


Fig. 13. Character string candidates extracted from text on trash cans.

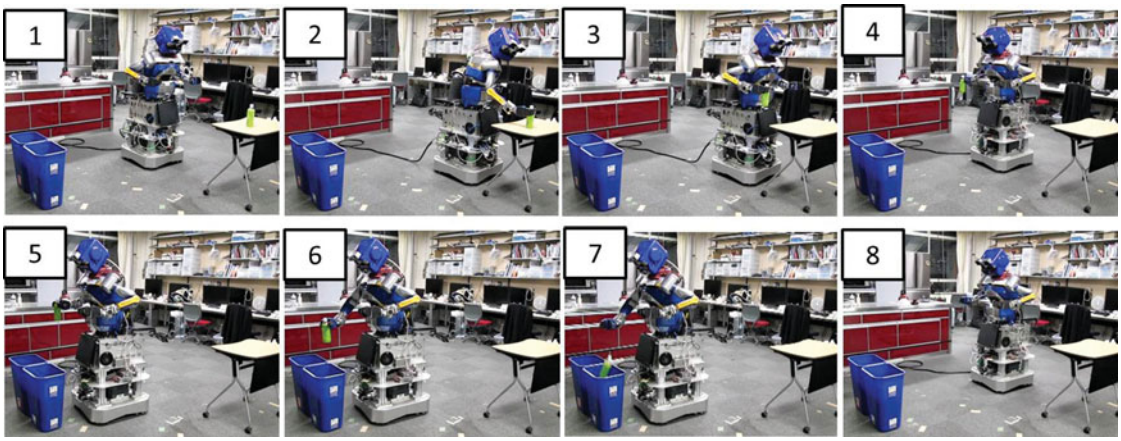


Fig. 14. Time-sequence of plastic bottle detection and trashing.

replaced by Epstheins *et al.*'s method because their method is compatible with our inclination compensation using the Hough transform, which independently detects each character.

All the above methods, including our contour-based character detection, are restricted to large character sizes in images. However, as shown in Fig. 6, robots encounter many instances of small size fonts. We overcame this problem by frequency filtering, which is easily implemented, but which returns an undesirable number of false positives. Although incorrect results were removed by the character-reading process, a preferable option is to reduce the number of false positives in the filtering. This could be achieved by a learning-based approach. Zhao *et al.*[30] proposed discriminative dictionaries consisting of edges extracted by wavelet transform. Other approaches are an AdaBoost-based approach,[9] and unsupervised learning using sets of small image patches.[5] Because these methods are insensitive to character thickness, they may allow detection of relatively small characters. As such, they might be suitable alternatives to our frequency filtering process that is rather sensitive to high-frequency noises.

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

7. Conclusions

In this paper we reported the use of character information by robots operating in daily environments, where visualized text varies in size, inclination and skew. We developed visual functions and a robotic hardware system that targets such diverse text and validated its applicability. A visual function method based on closed contours enables the detection of various character strings observed from different viewpoints. To cope with tiny and distant characters, the image processing is collaborated with a camera system with mechanical gaze adjustment. By combining the detection result with OCR, we accomplished routine reading-based tasks by an autonomous robot.

At present, our method is limited to character strings printed in planar surfaces. In future work, we will extend the method to various surfaces of different curvatures such as cylinders, and thereby implement and evaluate other daily assistance tasks.

References

1. C. Case, B. Suresh, A. Coates and A. Y. Ng, "Autonomous Sign Reading for Semantic Mapping," *Proceedings of International Conference on Robotics and Automation* (2011) pp. 3297–3303.
2. M. Cai, J. Song and M. R. Lyu, "A New Approach for Video Text Detection," *Proceedings of ICIP*, vol. 1 (2002) pp. 117–120.
3. H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk and B. Girod, "Robust Text Detection in Natural Images with Edge-Enhanced Maximally Stable Extremal Regions," *Proceedings of the International Conference on Image Processing* (2011) pp. 2609–2612.
4. P. Clark and M. Mirmehdi, "Finding Text Regions Using Localized Measures," *Proceedings of the 11th British Machine Vision Conference* (2000) pp. 675–684.
5. A. Coates *et al.*, "Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning," *Proceedings of the International Conference on Document Analysis and Recognition* (2011) pp. 440–445.
6. B. Epsthein, E. Ofek and Y. Wexler, "Detecting Text in Natural Scenes with Stroke Width Transform Stroke Width Transform," *Proceedings of the International Conference on Computer Vision and Pattern Recognition* (2010) pp. 2963–2970.
7. K. Jung, K. I. Kim and JungHyun Han, "Text Extraction in Real Scene Images on Planar Planes," *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02)*, vol. 3 (2002) pp. 469–472.
8. E. D. Haritaoglu and I. Haritaoglu, "Real Time Image Enhancement and Segmentation for Sign/Text Detection," *Proceedings of the International Conference on Image Processing* (2003) pp. 14–17.
9. S. M. Hanif and L. Prevost: "Text Detection and Localization in Complex Scene Images Using Constrained Adaboost Algorithm," *Proceedings of 10th International Conference on Document Analysis and Recognition* (2009) pp. 1–5.
10. B. Ibraheem, A. Hamdy and N. Darwish: "Textual signs reading for indoor semantic map construction," *Proc. Int'l Conf. Comput. Appl.*, **53**(10), 36–43 (2012).
11. T. Kabata, H. Watabe and T. Kawaoka, "Extraction method of the character string area for signboard understanding," SIG Technical Reports, No. 26, pp. 159–166 (2007).
12. S. Kim, D. Kim, Y. Ryu and G. Kim, "A Robust License-Plate Extraction Method under Complex Image Conditions," *Proceedings of International Conference on Pattern Recognition*, vol. 3 (2002) 216–219.
13. E. Klingbeil, B. Carpenter, O. Russakovsky and A. Y. Ng, "Autonomous Operation of Novel Elevators for Robot Navigation," *Proceedings of the IEEE International Conference on Robotics and Automation* (2010) pp. 751–758.
14. J. Liang, D. Doermann and H. Li, "Camera-based analysis of text and documents: a survey," *Int. J. Doc. Anal. Recognit.* **7**(2), 84–104 (2005).
15. R. Lienhart and F. Stuber, "Automatic Text Recognition in Digital Videos," *Proceedings of SPIE*, vol. 2666 (1996) pp. 180–188.
16. A. Miene, T. Hermes and G. Ioannidis, "Extracting Textual Inserts from Digitalvideos," *Proceedings of ICDAR* (2001) pp. 1079–1083.
17. R. Minetto, N. Thome, M. Cord, J. Fabrizio and B. Marcotegui, "Snoopertext: A Multiresolution System for Text Detection in Complex Visual Scenes," *Proceeedings of 2010 17th IEEE International Conference on Image Processing (ICIP)* (2010) pp. 3861–3864.
18. J. Miura, Tsuyoshi K and Y. Shirai, "An Active Vision System for Real-Time Traffic Sign Recognition," *Proceedings of IEEE International Conference on Intelligent Transportation System* (2000).
19. K. Nagahama, T. Nishino, M. Kojima, K. Yamazaki, K. Okada and M. Inaba, "End Point Tracking for a Moving Object with Several Attention Regions by Composite Vision System," *Proceedings of IEEE International Conference on Mechatronics and Automation* (2011) pp. 590–596.
20. L. Neumann and J. Matas: "Scene Text Localization and Recognition with Oriented Stroke Detection," *Proceedings of the International Conference on Computer Vision* (2013) pp. 97–104.
21. P. G. Nguyen and H. J. Andersen, "A Fast Sign Localization System Using Discriminative Color Invariant Segmentation," *Proceedings of the International Conference on Pattern Recognition and Computer Vision* (2008).

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press

22. T. Nishino, K. Yamazaki, K. Okada and M. Inaba: "Extraction of Character String Regions from Scenery Images Based on Contours and Thickness of Characters," *Proceedings on IAPR Conference on Machien Vision Applications* (2011) pp. 307–310.
23. Y.-F. Pan, X. Hou and C.-L. Liu, "Text Localization in Natural Scene Images Based on Conditional Random Field," *Proceedings of the 10th International Conference on Document Analysis and Recognition* (2009) pp. 6–10.
24. K. Papineni, S. Roukos, T. Ward and W. Zhu, "Bleu: A Method for Automatic Evaluation of Machine Translation," *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (2002) pp. 311–318.
25. T. Q. Phan, P. Shivakumara, B. Su and C. L. Tan, "Gradient Vector Flow-Based Method for Video Character Segmentation," *Proceedings of the 2011 International Conference on Document Analysis and Recognition* (2011) pp. 1024–1028.
26. A. V. Reina, R. J. Sastre, S. L. Arroyo and P. G. Jimenez: "Adaptive Traffic Road Sign Panels Text Extraction," *Proceedings of the 5th WSEAS International Conference on Signal Processing, Robotics and Automation* (2006) pp. 295–300.
27. E. Ryu, N. Onishi, T. Yamamura and N. Sugie, "Extraction of character series regions in scenes," *Trans. Inst. Electron. Inf. Commun. Eng.* **10**(3), 23–26 (1999).
28. H. Takahashi, K. Kasai, D. Kim and M. Nakajima, "Extraction of text regions from scenery images using multiple features," *ITE Techn. Rep.* **25**(11), 39–44 (2001).
29. A. J. Trevor, J. Rogers III, C. Nieto-Granda and H. I. Christensen, "Feature-based Mapping with Grounded Landmark and Place Labels," Workshop on Grounding Human-Robot Dialog for Spatial Tasks. Robotics, Science and Systems (2011).
30. M. Zhao, S. Li and J. Kwok: "Text detection in images using sparse representation with discriminative dictionaries," *J. Image Vis. Comput.* **28**(12), 1590–1599 (2010).
31. S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong and R. Young, "Icdar 2003 Robust Reading Competitions," *ICDAR'03* (2003) pp. 682–687.
32. S. M. Lucas, "Icdar 2005 Text Locating Competition Results," *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, vol. 1 (2005), pp. 80–84.
33. D. Karatzas, "ICDAR 2011 Robust Reading Competition - Challenge 1: Reading Text in Born-Digital Images," *Proc. of Document Analysis and Recognition* (2011) pp. 1485–1490.
34. tesseract-ocr http://code.google.com/p/tesseract-ocr/.
35. NHocr http://sourceforge.jp/projects/nhocr/.
36. The ICDAR 2003 competitions http://algoval.essex.ac.uk/icdar/Competitions.html.

https://doi.org/10.1017/S0263574714002094 Published online by Cambridge University Press