A Vision System for Daily Assistive Robots Using Character Information in Daily Environments

Kimitoshi Yamazaki*[1], Tomohiro Nishino*[2], Kotaro Nagahama*[2], Kei Okada*[2], and Masayuki Inaba*[2]

Abstract— **This paper describes a vision system that is for using character information for robots working on daily environments. It is mainly composed of image-based recognition module that detects character candidates by taking particular note of closed contour in an image. In addition, to cope with tiny or distant characters, a camera system with mechanical gaze adjustment is collaborated with the image processing. By combining the detection result with OCR (Optical Character Reader), applications to daily assistance by an autonomous robot were introduced.**

I. Introduction

Characters existing in daily environments provide rich information to human beings, and the same holds for automated machines. That is, many of our behaviors in daily living is influenced by character information. For autonomous robots working in daily environments, the ability to get information from character is also important to do meaningful work. The purpose of this research is to propose a vision system that enables the robot to use character information for effective working. This paper describes a character detection system that is suitable for robots working in living environment. In addition, it is collaborated with character reading techniques, and robotic applications in real world are introduced.

Detecting characters in real world have been one of important issues for understanding scenes. If we consider applications to daily assistive robots, there are several issues that should be solved. Fig. 1 shows an example image that is captured at a living environment. This may be a similar viewpoint as same as a person, and daily assistive robots. We can find a set of furniture and other objects everyday used. As shown in this figure, one of the difficulties for using characters is that we must detect them under the condition of various viewpoints. It means that characters may be observed with small size and skewed shape in an image, so a crucial issue is to design a framework that can cope with variously-sighted characters.

In our research, we applied a method of character string detection based on closed contours, which permits a certain level of sloping and skew of characters. Moreover, we use a camera system that equips convergence control function and electrical zoom lens. Combination of the image processing and gazing control enables a robot to detect various character strings in living environment. In this paper, by combining these results with OCR software, practical robotic applications are introduced.

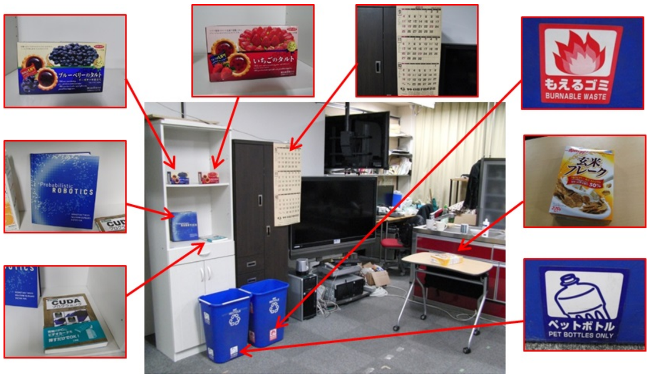


Fig. 1. Characters in a living environment

This paper is organized as follows: section II shows related work, and section III denotes issues and approaches. Section IV explains the method of character string detection. Section V explains function and mechanism for detecting less-visible characters. Section VI shows some experimental results, and section VII presents our conclusion.

II. Related work

A. Robotic applications using meaningful artificial signs

In the field of mobile robotics, there have been researches that utilized meaningful sign information [2]. Miura et al. [11] developed a system that used traffic signs. Image-based recognition method was designed by intensively using built-in functions of a compassable image processing board. The effectiveness of the system was proven through on-road real-time experiments. Klingbeil et al. [6] achieved a task of detecting, localizing, and labeling of elevator buttons. The method is based on machine learning techniques to take advantage of contextual features. They collected a dataset of 150 pictures of elevator panels from more than 60 distinct elevators, and succeeded in correctly localizing and labeling 86.2% of the buttons.

These previous researches focused on the use of sign or character information under the assumption that the region where they exist could be specified relative easily. For instance, traffic signs exist independently on the side of a road, and elevator buttons embedded in elevator panels.

B. Character string detection

There have been many of researches that proposed the methods of character strings detection from natural scene

*[1] Faculty of Engineering, Shinshu University, 4-17-1 Wakasato, Nagano, Nagano, 380-8553, Japan. [2] Graduate School of Systems and Information Engineering, The University of Tokyo, 1-1-1 Hongo, Bunkyo-ku, Tokyo, 305-8573, Japan. yamazaki@shinshu-u.ac.jp

images. The algorithms can be broadly classified under three types [7]: gradient-based [5] [15], color segmentation based [3] [8] and texture-based [1] [10].

In our assuming environments, that is daily environments, characters are existing on various objects; e.g. books, food packages, plastic bottles. One issue is that there are many patterns of differences depending on what we see characters. Because of perspective projection, a straight character string may be rotated and skewed. These situations were not assumed at researches that used horizontally aligned character strings in images included in the ICDAR dataset [18].

One of the feasible approaches to cope with rotation and skew is to firstly detect noteworthy region where characters are printed. Kabata et al. [4] targeted characters on panels. Similar approach was used to find signs existing the side of roads [16]. These researches were performed under the assumption that regions where meaningful information were printed must be specified. However, it is desired that characters are directly detected without region identification if they exist on various types of objects.

III. Issues and Approach

There are two major issues that relate to character string detection:

1) **How to detect sloping character string:**
Because robots can have various point of view, character strings are not limited to be captured with well-aligned condition. Not only rotation along optical axis but also perspective skew may occur. In general, OCR software does not permit such rotated or skewed characters as input.

2) **How to detect characters that are captured with small sizes:**
As shown in Fig.1, characters are often captured with small size. In general, character detection process needs a certain level of large characters in an image. Small characters do not become candidates to read. This means that we have much loss of characters information.

To overcome above issues, we developed a vision system as shown in Fig. 2. Eight series of software modules (A) to (H) depicted on upper side of the figure can be divided depending on the following three technological elements.

One is a set of image processing methods for detecting character string, which are shown as (A) and (B). These are based on closed contour detection that is proposed by Takahashi et al. [17]. It is assumed that a character string consists of characters which lie on a straight line. We extend this method to extract arbitrary rotated or skewed character string. The detail is described in section IV.

The approach using closed contours is restricted to the scenes having large characters. It is insufficient for our target environment as shown in Fig.1. So second function is for detecting candidates of small characters indicated as (C). It is explained at section V.A.

Third is a group of modules that control convergence stereo hardware for gaze adjustment, which are (D), (E) and (F). This system makes it possible to detect tiny characters, to zoom up them, and to get 3D information around them. The details are described in section V.B.

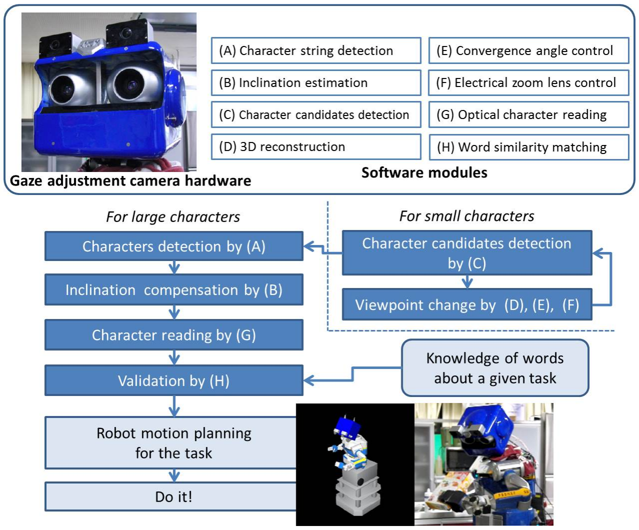

(A) Character string detection
(E) Convergence angle control
(B) Inclination estimation
(F) Electrical zoom lens control
(C) Character candidates detection
(G) Optical character reading
(D) 3D reconstruction
(H) Word similarity matching
Gaze adjustment camera hardware
Software modules
For large characters
For small characters
Characters detection by (A)
Character candidates detection by (C)
Inclination compensation by (B)
Viewpoint change by (D), (E), (F)
Character reading by (G)
Knowledge of words about a given task
Validation by (H)
Robot motion planning for the task
Do it!

Fig. 2. Configuration of the vision system

IV. Character String Detection Based on Closed Contour

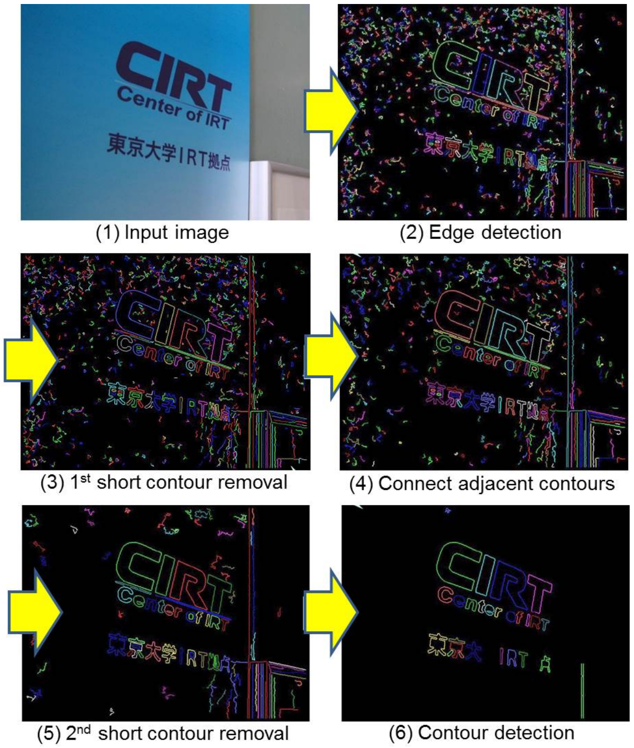
A. Closed contour extraction

All of characters containing the property that they have closed contours. This property is not limited to only horizontally aligned characters, but rotated and skewed characters.

The detection method that is an extension of the method [17] is composed of edge detection, contour detection and improvement processes as shown in Fig. 3. First, canny edge detection is applied to an input image (Fig. 3, (2)). From this result, short edge segments whose length is less than θ_{l1} are removed (Fig. 3, (3)). On the other hand, edge re-connection between neighborhoods is added because edges that have steeply changes of its direction are often disconnected in the above detection process. Endpoints of two edges whose distance is less than θ_{cn} are connected (Fig. 3, (4)). For instance, 'R' in 'CIRT' have been divided into two parts at (3). However, they were connected and formed one contour through this re-connection. After that, short segments whose length is less than θ_{l2} are again removed (Fig. 3, (5)). Remaining edges are selected as closed edges if the distances between two end-points is less than θ_{cl} (Fig. 3, (6)).

B. Character string region extraction

Character strings are extracted by determining the linearity of their alignments. First, a circumscribed rectangle is calculated for each closed contour. Next, the rectangle is slid to right by some pixels until it is moved as much as the width of the rectangle. If the rectangle includes the center of the circumscribed rectangle of another closed contour, these two closed contours are assumed to be aligned horizontally and to be included in the same character string.


(1) Input image

(2) Edge detection

(3) 1st short contour removal

(4) Connect adjacent contours

(5) 2nd short contour removal

(6) Contour detection

Fig. 3. Closed contour detection. The difference of edge color means that they are difference contours.

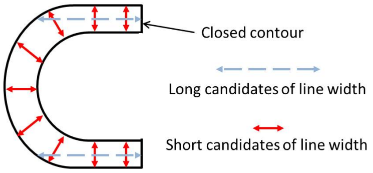


Fig. 4. Candidates of line width

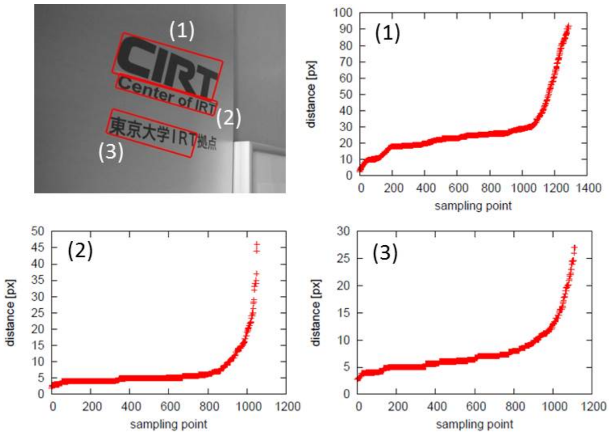


Fig. 5. Upper left picture: originally detected character string, graphs (1) to (3) show sorted candidates of characters' thickness

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

Since closed contours are extracted from characters, thickness of characters can be calculated. First, a point on a closed contour is sampled, a vertical line from the point is calculated, Next, the intersection of the line and a closed contour is also calculated. The distance between the sampled point and the intersection point is one of the candidates of the character' s thickness. This intersection calculation is performed for many sampled points, the results are arranged in descending order. Fig. 4 shows an example that short and long thickness candidates exist in a closed contour.

After the process of extraction of the thickness, character string regions are revised based on it. Both closed and unclosed contours that lie left or right of character string regions are extracted, and each thickness is calculated. These contours are added to the character string region if they have the similar thickness to that of characters in that region.

Graphs presented in Fig. 5 show the sorted candidates of characters' thickness against to three character strings. These show that the gradient of the sorted candidates is low around the middle, so thickness of characters can be detected safely by selecting the middle. By this process, both closed and unclosed contours are able to be included in character string region properly.

D. Inclined string detection based on Hough Transform [13]

The method described above subsections assumes that a character string is captured at horizontal line. This may prevent getting right result when character strings are inclined. However, character strings may practically be written with inclined line, or daily objects which have target strings will be placed with arbitrary direction from a camera. To cope with such various conditions, we take an approach to estimate the inclination angle based on finding a line of characters. The method complies with Hough transform. With this method, we detect lines which pass the center of contours in character string.

One point in (x, y) space can be represented as a sine curve in (r, θ) space. A center of closed contour is regarded as a point, and each point translated as a sine curve into (r, θ) space. If there are crossing points in the space after this translation, it indicates that there are a reliable straight line exists in the (x, y) space.

The (r, θ) space is divided into lattice cells, and cells which include sine curves acquire a vote with considering a weighting factor according to the completeness of closed contour. That is, if the contour is entirely closed, large value is voted. Otherwise, the voted value is adaptively defined depending on the distance d that is defined from one tip of an edge to another.

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

V. Function and Mechanism for Detecting Less-visible Characters

A. Frequency filtering for detecting character candidates

For finding more characters, a vision system should find the placements of characters from an image captured at wide

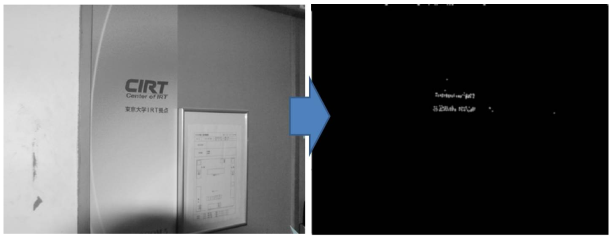


Fig. 6. Detection of character candidates by means of image frequency filtering. A group of white pixels show the region.

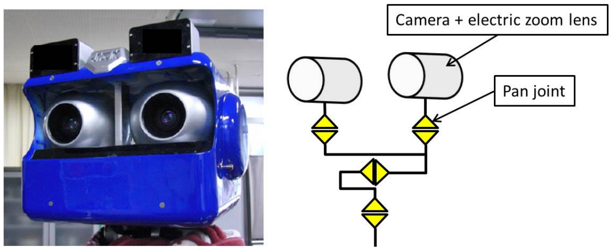


Fig. 7. The mechanism of a gaze adjustment camera system

angle of view. Fourier transform is used for the purpose.

Based on the consideration that small size characters in images should aggregate at high frequency region, we apply two dimensional discrete Fourier transform (DFT). Let be an image $I(x, y)$, whose width is w and height is h. Fourier transform of this image is performed by the following equation:

$$F(u, v) = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I(x, y) exp(-2\pi j(\frac{ux}{w} + \frac{vy}{h})). \qquad (2)$$

Two dimensional discrete Fourier transform is applied to $\hat{F}(u, v)$ that is particular frequencies extracted from $F(u, v)$, we can get spatial frequency filter,

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

This filter is used to guess the region having characters that are too small to detect closed contours. Fig. 6 shows an example. These filtering results that are considered as characters are transferred to a confirmation process by using gaze adjustment camera system described in next subsection.

B. A Camera System with Mechanical Gaze Adjustment and Its Control Policy

Fig. 7 shows our camera system [12]. A stereo camera is made up of two “FCB-IX11A” camera modules made by SONY Co.,Ltd. They have electric zoom lenses that change optical zoom factor from 1.0 to 9.5 times. Combined with $0.7\times$ wide conversion lens “VCB-HG0730X” by SONY Co.,Ltd, their view angle is controlled from 65.7[degree] to 6.9[degree]. With a VGA (640[pixel] $\times$ 480[pixel]) image size and its maximum zoom factor (9.5 $\times$), about 20/13 vision worth of resolution (USA unit) is acquired. Each zoom camera can move independently around its pan joint. The maximum accuracy of the joint angle of the pan axis is 0.00125[degree/count]. This value is sufficiently smaller than the value of the view angle per pixel at the maximum zoom factor, 6.9/640 = 0.01078[deg/pixel].

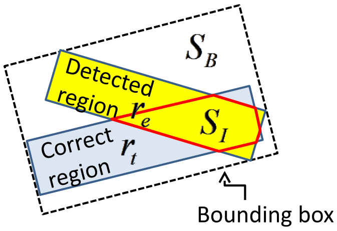


Fig. 8. Evaluation method of the accuracy of character string detection

1) 3D position calculation: For our purpose, the camera system should capture images depending on the needs: wide angle view are needed for detecting the candidate of character strings, and enlarged view are needed for verifying each candidate. The former case, 3D position of the candidates is useful for detecting the viewpoint for next enlarged images. For this reason, two cameras are driven for actualizing convergence eye movement, and capture target characters in the center of enlarged images. The size adjustment can be achieved by using electric zoom lens.

The calculation of the 3D position is explained. We denote a point that lies in the direction $\mathbf{v} = (x, y, z)$ in camera coordinates, and its image coordinates as $\mathbf{P} = (u, v)$. Using intrinsic camera parameters $\mathbf{M}$, we can get the following equation:

$$\mathbf{v}' = \mathbf{M}^{-1} \begin{pmatrix} \mathbf{P} \\ 1 \end{pmatrix}, \qquad (4)$$

where $\mathbf{v} = \mathbf{v}'/||\mathbf{v}'||$. The position of the $\mathbf{P}$ is calculated from a pair of $\mathbf{v}_l$ and $\mathbf{v}_r$ under the condition that the poses of left and right cameras in world coordinates are calculated from robot position, joint angles of the robot body, and the angle of convergence. Although $\mathbf{v}_l$ and $\mathbf{v}_r$ should intersect at one point in ideal condition, it is not in real because of various measurement errors. For this reason, the intersection point is calculated as a middle point where the distance between $\mathbf{v}'_l$ and $\mathbf{v}'_r$ becomes minimum.

2) Zoom ratio calculation: For focusing on a target character string and capturing an enlarged image, both the direction of a camera and the ratio of zoom should be decided.

Let present zoom ratio as z, the width and height of an image as w_i and h_i. When the width and height of a bounding box that includes a target character string are w_b, h_b, the next zoom ratio $z_0$is calculated as follows:

$$z_0 = z \times \left(min \frac{w_i}{w_o}, \frac{h_i}{h_o} \right) \qquad (5)$$

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

VI. EVALUATION AND EXPERIMENTS

A. Performance evaluation of character string detection using ICDAR dataset

The method of character string detection was evaluated by using the dataset that has been used for the ICDAR 2003 robust reading competitions [19]. We resized all images keeping aspect ratio so as to be included in 320 × 240 resolution. The evaluation method was as same as Text Locating Competition at ICDAR 2003 and 2005 [9]. Fig. 8 shows the method. The size of overlapped area $S_B(r_e, r_t)$ between detected bounding box r_e and correct bounding box r_t were defined as follows:

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

Based on them, precision p, recall r and f-measure f were calculated as follows:

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

where E was a set of detected characters and T was a set of correct characters. α was set to 0.5 in this experiment.

Table I shows the result of the evaluation of character string detection using TrialTrain and TrialTest included in ICDAR dataset. ICDAR dataset included images that could not observe all of character strings because of illumination reflectance. In some cases, it was difficult to detect correct contour because of bright background region. However, our method including compensation of character region based on thickness of characters provided feasible results.

B. Performance evaluation using natural images

Images in ICDAR dataset we used only has horizontally aligned character strings. However, robots working in real world should detect and read character strings that captured with inclination and skew. To investigate the effectiveness to the strings, we collected images around our laboratory. The number of images were 107. After closed contour detection, character candidates were detected and character strings were generated. The evaluation method was same with the method described in [9]. The result was as follows: $precision = 0.44$, $recall = 0.46$ and $f - measure = 0.43$. Fig. 9 shows the examples. Various character strings printed on objects could be detected in spite of inclination and skew.

C. Combination with OCR

The method of character string detection presented in this paper makes it possible to output a set of horizontally-aligned character strings. That is, these results respond to the requirement of common OCR (Optical Character Reader). To cope with English and Japanese, tesseract-ocr [20] and NHocr [21] were connected to our character detection software.

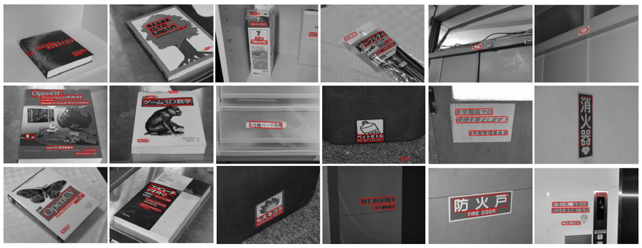


Fig. 9. Examples of character string detection on home & office environments

However, just joining of them was insufficient because of following reasons.

- As an aligned character string was an image region where skew changes were added, the region had potential to include quantization error.
- If an image rotation angle estimated by Hough transformation had slight mis-alignement, the success rate of character reading by OCR is decreased.

Both of them caused the failure of character reading. To reduce these problems, a similarity measure BLEU [14] was adopted. BLEU is a criterion for evaluating similarity of characters by using n-gram correspondence.

D. Robotic application: Separation of trash

To dump garbage into trash correctly, we have to know which is a proper trash box for the garbage. In many cases, we can get the information by sign of character strings printed on trash boxes.

Under the assumption, we developed an robotic application that found proper trash box and throw garbage into it. A life-sized humanoid robot mounting a camera system described in V.B was used. Two trash boxes each of which had labels ’combustible material’ and ’plastic bottle’ in Japanese and English were placed on a room. A plastic bottle was placed on a table near to the beginning position of a robot. The robot found and grasped a plastic bottle, and trashed it into correct trash box.

The difficulty of the application is viewpoint limitation of the robot. Because character strings were printed lower on the trash boxes, the strings in image captured by the robot must have perspective skew. In such case, our method described in section V was useful. As shown in a center figure of Fig. 10, the robot first detected the candidates of character strings by using wide view angle. After that, each candidate was focused on, and high resolution images were captured. Character string detection and its alignment were performed, and the results were input to OCR. Their outputs were used to select right trash box. Fig. 11 shows a sequence of the experiment.

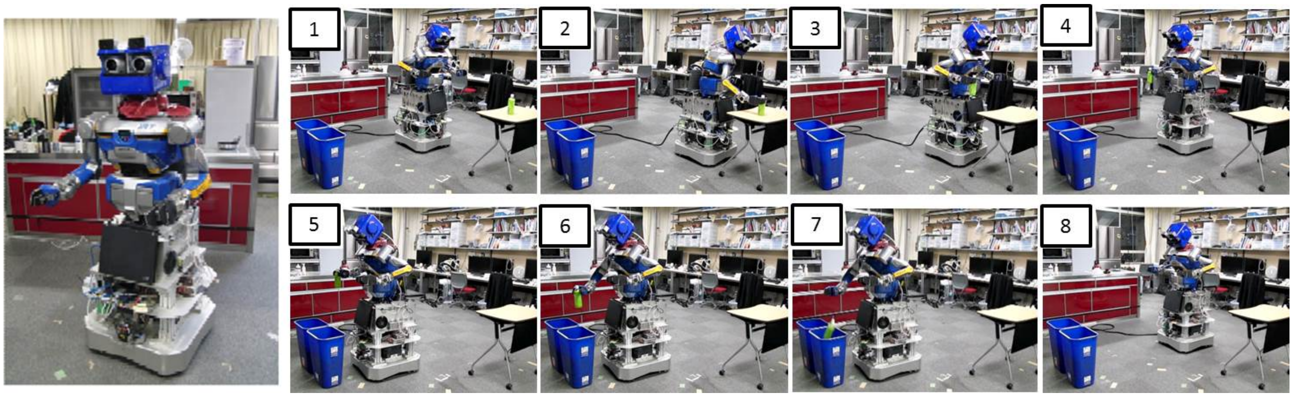


Fig. 11. A sequence of plastic bottle detection and trashing

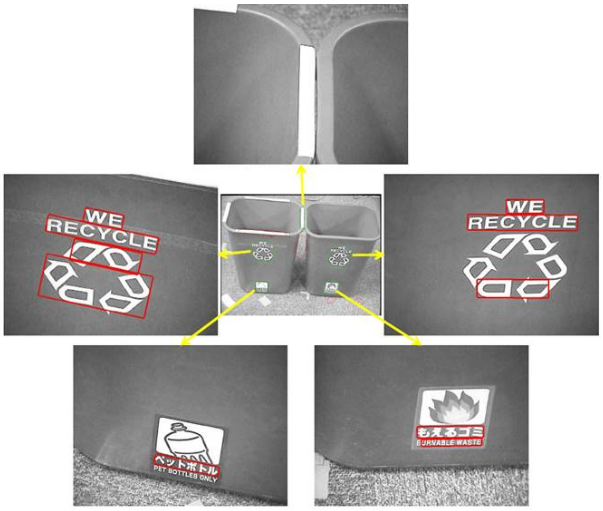


Fig. 10. Character string candidates extracted on trash boxes

VII. Conclusions

In this paper we reported the use of character information by robots that work in daily environments. To find character strings that are observed with various size, inclination and skew were targeted, visual functions and robotic hardware system were proposed and proven. By combining the detection results with OCR, an application to daily assistance by an autonomous robot was introduced.

For future work, because the present system can only cope with character strings printed on planar surface, we will extend the method to various curvature surface such as cylinder.

References

[1] P. Clark and M. Mirmehdi: "Finding text regions using localised measures," In Proc. of the 11th British Machine Vision Conference, pp. 675 – 684, 2000.

[2] B. Ibraheem, A. Hamdy and N. Darwish: "Textual Signs Reading for Indoor Semantic Map Construction," in Proc of Int'l Conf. on Computer Applications, Vol.53, No.10, pp.36 – 43, 2012.

[3] K. Jung, K. I. Kim, and J. Han: "Text extraction in real scene images on planar planes," In Proc. of the 16 th Int'l Conf. on Pattern Recognition, Vol. 3, pp. 469 – 472, 2002.

[4] 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.

[5] S. Kim, D. Kim, Y. Ryu, and G. Kim: "A robust license-plate extraction method under complex image conditions," In Proc. of Int'l Conf. on Pattern Recognition, Vol. 3, pp. 216 – 219, 2002.

[6] E. Klingbeil and B. Carpenter and O. Russakovsky and A. Y. Ng: "Autonomous operation of novel elevators for robot navigation," in Proc. of IEEE Int'l Conf. on Robotics and Automation, pp. 751 – 758, 2010.

[7] J. Liang, D. Doermann, and H. Li: "Camera-based analysis of text and documents: a survey," International Journal on Document Analysis and Recognition, Vol. 7, No. 2, pp. 84 – 104, 2005.

[8] Lienhart R and Stuber F.: "Automatic text recognition in digital videos," In Proceedings of SPIE, Vol. 2666, pp. 180 – 188, 1996.

[9] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong and R. Young. "ICDAR 2003 robust reading competitions," In ICDAR'03, pp. 682 –687, 2003.

[10] A. Miene, Th. Hermes, and G. Ioannidis: "Extracting textual inserts from digital videos," In Proc. of ICDAR, pp. 1079 – 1083, 2001.

[11] J. Miura , T. Kanda , Y. Shirai: "An Active Vision System for Real-Time Traffic Sign Recognition," in Proc. of IEEE Int'l Conf. on Intelligent Transportation System 2000.

[12] K. Nagahama, T. Nishino, M. Kojima, K. Yamazaki, K. Okada, M. Inaba: "End Point Tracking for a Moving Object with Several Attention Regions by Composite Vision System," in Proc. of IEEE Int'l Conf. on Mechatronics and Automation, pp.590 – 596, 2011.

[13] T. Nishino, K. Yamazaki, K. Okada and M. Inaba: "Extraction of Character String Regions from Scenery Images Based on Contours and Thickness of Characters," in Proc. on IAPR Conf.on Machien Vision Applications, pp. 307 – 310, 2011.

[14] K. Papineni, S. Roukos, T. Ward, and W. Zhu: "Bleu: a method for automatic evaluation of machine translation," In Proc. of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311 – 318, 2002.

[15] T. Q. Phan, P. Shivakumara, B. Su and C. L. Tan: "Gradient Vector Flow-Based Method for Video Character Segmentation," in Proc. of the 2011 Int'l Conf. on Document Analysis and Recognition, pp. 1024 – 1028, 2011.

[16] A. V. Reina, R. J. Sastre, S. L. Arroyo and P. G. Jimenez: "Adaptive traffic road sign panels text extraction," in Proc. of the 5th WSEAS Int. Conf. on Signal Processing, Robotics and Automation, pp. 295 – 300, 2006.

[17] H. Takahashi, K. Kasai, D. Kim, M. Nakajima: "Extraction of text regions from scenery images using multiple features," ITE Technical Report, vol.25, no.11, pp.39 – 44, 2001.

[18] ICDAR2003 Datasets, http://algoval.essex.ac.uk/icdar/Datasets.html.

[19] The ICDAR 2003 competitions, http://algoval.essex.ac.uk/icdar/Competitions.html.

[20] tesseract-ocr http://code.google.com/p/tesseract-ocr/.

[21] NHocr http://sourceforge.jp/projects/nhocr/.