H-016

Development of an Intelligent Shopping Robot Based on Human-Robot Visual Communication: Segmentation and Recognition of a Display Object's Barcode

Flora Thach Thao Pham † Yui Tanjo †

1. Introduction

We are currently facing the issue of an aging population [1]. With age comes various health impediments, among them, there is reducing mobility [2]. Having limited mobility greatly affects one's quality of life. In fact, an important aspect of one's quality of life is linked to the ability to move, the satisfaction with functional capacity, in other words to have the autonomy to achieve basic everyday life activities [3]. We believe that assistive robots can be part of the solution to bring more autonomy to elderly people.

We develop a shopping robot. Not only doing groceries is an essential activity, it also can represent a long trip for a person with reduced mobility. The objective is not simply to provide the items to the user but also to allow the user to enjoy a groceries experience as if they were in a store through a robot.

A very important criterion for this system is the ease of use. Particularly for 80 years-old and older people, issues related to the ability of using technology were observed [4]. By using visual communication that requires minimal effort from the user side, we hope to provide a solution accessible to anyone. In practice, the RGB images from the RGB-D camera mounted on the robot are streamed on the user computer. One click on the object displayed on screen launches the segmentation, and another click validates the segmentation result or not. To perform the segmentation, both depth and RGB information are used.

When doing groceries, an important part of choosing between a product from different brands is the composition. Furthermore, elderly people can have dietary restrictions due to their medication [5]. Composition can be crucial in case of severe allergies. Therefore, we add barcode recognition functionality to the robot. Using a product barcode number, we can find its composition in an online dataset.

In this paper, we present the software of the robot with its various algorithms. The focus will be the barcode recognition feature where both algorithm (Fig. 3) and experimental results will be presented.

2. Related work

This paper presents the continuation of a human-robot cooperative system based on visual communication [6]. This time, a barcode recognition feature has been added to respond to the need for information on the product the user wishes to acquire.

† Kyushu Institute of Technology

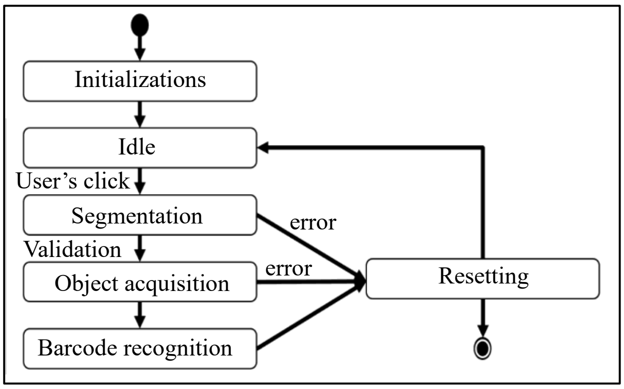

Fig. 1 Robot state-transition diagram

Barcode recognition using visual information has been a deeply studied topic with a high demand from various industries. Using deep learning approach in a barcode recognition algorithm has been the trend for the past few years [7]. Some researches use artificial intelligence purely for detection purposes [8][9]. Others use it for denoising and recognition purposes [10]. There is also a research that combines various models to present a barcode recognition solution entirely powered by deep learning [11]. These papers' algorithms take images or a video of non-moving barcodes as input. For moving barcode, we can find few works with conveyer belt for the shipping industry [12].

However, in all those papers, it is assumed that the barcode will be directly visible by the camera, or human intervention will make the barcode visible. In our research, the objective is to develop a system able to autonomously manipulate an item to read its barcode.

3. Methodology

The robot system uses a RealSense D415 camera [13] as a sensor for both visual and depth information. A five-degree-of-freedom Dynamixel arm, equipped with a two-finger gripper end-effector is used for object manipulation.

Most of the image processing is handled in a C++ program using OpenCV [14]. The zbar library [15] is used to decode the barcode. To detect and localize the barcode, the robot system uses the YOLOv9 model, trained with a public dataset [16] and additional images taken from experiments, in a Python program. The two programs exchange information using socket communication. Figure 1 gives the state-transition diagram of the robot software.

3.1 Object segmentation

Using Intel RealSense SDK 2.0 [17], we can compute the normalized normal vector map of the camera depth frame. Using the user's click as the seed, a region growth algorithm is run to obtain a first mask. Since there is missing data noise, a hole filling algorithm is applied to improve the mask.

If more than one object is on the mask because they were displayed juxtaposed, we detect the shadow line at the edge of the target to isolate it in the RGB image. Using noise filters and the Canny filter on the RGB image masked by the depth segmentation[18], a binary image of the objects' edges is obtained. We can then apply a conventional Hough transform to find the lines[18]. Another region's growth algorithm using the user click as the seed is used to obtain the final region. The smallest rectangle containing the region is used for object acquisition.

3.2 Object localization and size estimation

Using the principle point of the camera (ppx, ppy), the focal length f and the average depth of the segmentation rectangle d, the center point, height and width of the object segmentation rectangle in pixels ((x_p, y_p), h_p, w_p) can be converted in meters in the camera coordinate plane ((x_m, y_m) h_m, w_m).

$$x_m = \frac{(x_p - ppx)d}{f}, \qquad y_m = \frac{-(y_p - ppy)d}{f}$$

$$h_m = \frac{h_p \times d}{f}, \qquad w_m = \frac{w_p \times d}{f}$$

The metric coordinates of the rectangle center are converted from the camera coordinate system to the arm coordinate system.

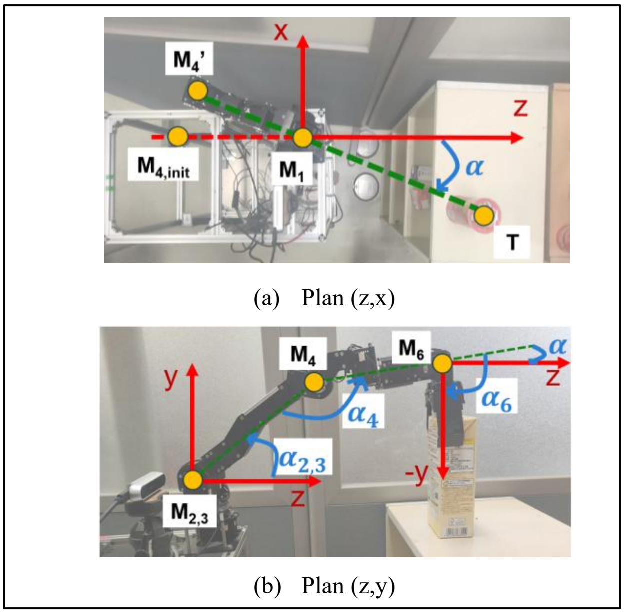

Fig 2　Kinematic problem decomposition

3.3 Arm movement control

There are 2 movement sequences performed by the robot: grabbing the object and showing various sides of the object for barcode recognition. They require computations to ensure the correct placement of the robot arm according to the object dimensions. For the first sequence, we want to place the arm to be able to grab the object from the top, the barcode being at the lower part of the packaging in most cases. For the second sequence, the objective is to display the barcode to the camera for recognition. We want the bottom of the object to be around the center of the camera view for different sides.

The ROBOTIS Dynamixel SDK [19] is used to control the motors of the arm according to the calculation. The arm movement is a three-dimensional kinematic problem. We simplified the computation of the solution by dividing the problem into two two-dimensional planes (Fig. 2).

At the initial position of the arm, only the motor M1 has a rotation along the y-axis. The motor M1 angle to align the arm with the target object T can be computed using trigonometry.

$$\alpha = arccos\left(\frac{\vec{z}.\overrightarrow{M_1T}}{\|\overrightarrow{M_1T}\|}\right)$$

To compute the angles for the motors $M_{2,3}$ and M_4, a two-dimensional version of the FABRIK algorithm [20] is used with the motor M_6 as the end-effector. The angle for the motor M_6 is computed with the following trigonometry formula.

$$\alpha_6 = \pi + arccos\left(-\frac{\vec{z}.\overrightarrow{M_4M_6}}{\|\overrightarrow{M_4M_6}\|}\right)$$

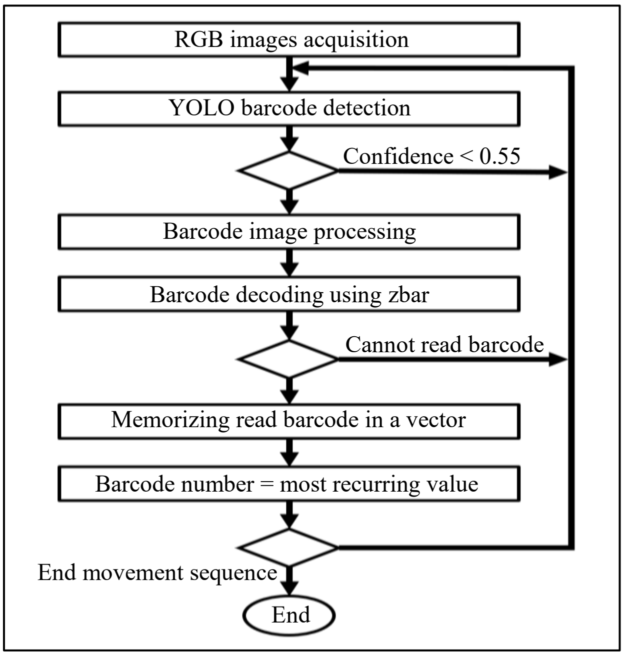

Fig 3　Barcode recognition

Fig 4　Experimental data for training

3.4 Barcode recognition

The YOLOv9 model was trained for barcode detection. An initial public dataset [16] was completed with images taken from experimental videos that contained more variety in linear distorted images (Fig. 4). In total, 1,113 images were used for training.

The image cropped with the bounding box resulting from YOLO is then processed. First, it is converted into a grayscale image. If either height or width is smaller than 200 pixels, it is resized. The zbar library reads grayscale images with a relatively large barcode.

A threshold filter is used to obtain the binary image of the barcode and a mask. The threshold value is computed for each frame. Since the barcode should be in the center of the bounding box, we use the center 50x50 square as input for the threshold calculation. Using its histogram, we can find the smallest value so that at least 55% of the pixels are darker and use it as the threshold.

Using a morphological closing filter, the mask is processed to find a more precise segmentation of a portion of a barcode. The minimum enclosing rectangle of the segmentation mask is used to find the angle for the rotation correction.

The zbar library scanning function is used on the processed barcode to attempt to read the barcode number. We attempt to read both binary and grayscale images with rotation correction as the binarization sometimes loose too much information. The barcode image processing is shown in Figure 5.

To ensure a robust reading, the reading for all frames during the second movement sequence of the robot are memorized. The result of the barcode recognition algorithm is the most recurring reading among the memorized barcode numbers.

4. Experiment

4.1 Experimental setup

In In this paper, we are focusing on the performance of the barcode recognition feature. Therefore, in the experiment, one or several objects are displayed about 45 cm away from the robot. There are at least 10cm spacing between objects to simplify object acquisition.

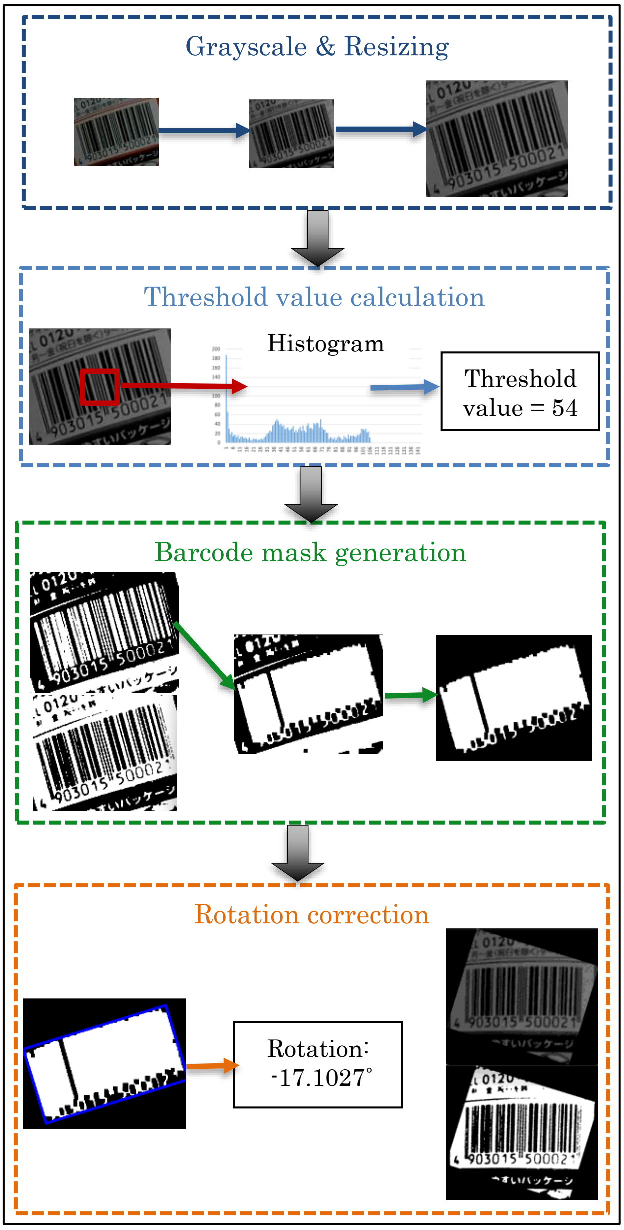

Fig 5　Barcode image processing

An experiment corresponds to the following scenario. The user will click on the displayed object to launch and validate the segmentation. The robot will grab the object and present the front and back side of the item. The barcode is either on the front or back side of the object. A total of 70 experiments were conducted.

4.2 Metrics

For each experiment, specific events are observed:

- Whole: the whole barcode was displayed on camera
- Detect: the barcode was detected
- Read: some information was retrieved from reading the barcode

Table 1 Experiments results

	whole	detect	read	correct
No. of experiments with the event	60	60	44	43

Table 2 Computer specifications

Processor	12th gen Intel Core i7-12700H, 2300MHz
RAM	32.0 GB
System type	64-bit OS

- Correct: the algorithm result is the correct barcode number

To specifically measure the performance of the barcode recognition software, we used a readability score *R*. This metric gives an indication of the barcode performance regardless of the object manipulation performance. It is computed using the number of experiences where the whole and correct events were observed, respectively N_W and N_C.

$$R = \frac{N_W}{N_C} \times 100$$

5. Results

Table 1 presents the detailed results for each experiment. Due to movement restrictions of the robotic arm, only two sides of the object can be shown. For cylindrical objects with the barcode initially at the back, the manipulations only exposed a partial barcode. There were also some occurrences where the object was dropped during the manipulations. This leads to only 60 experiments where the barcode is shown. The YOLOv9 model was able to detect the barcode in every experiment where it was fully shown. The readability score of R=72% was achieved.

The average runtime of the barcode recognition for each frame is 79.2ms. The program can be run at 12fps.

6. Conclusion and future work

We have developed a system able to acquire displayed objects and manipulate them to read its barcode. We were able to detect barcodes with non-rectilinear movements. The feasibility of an automated system to read barcodes on objects, even if they are not directly visible, was verified.

While the barcode recognition displays promising results, there is still room for improvement in both speed and reading ability. Firstly, a robot arm with at least one more degree of freedom will allow to do a 360° rotation of the object in front of the camera. This would eliminate the experiments where the whole barcode is not shown on a camera. Furthermore, adjusting the arm position based on the barcode detection might improve the number of readable frames. Focusing on the recognition algorithm, a different approach in the barcode processing, using only a horizontal section of the barcode, is considered to improve the processing time.

References

[1] World Health Organization, “WHO calls for urgent transformation of care and support systems for older people,” Oct. 2024. [Online]. Available: https://www.who.int/news/item/01-10-2024-who-calls-for-urgent-transformation-of-care-and-support-systems-for-older-people
[2] L. Ferrucci, R. Cooper, M. Shardell, E. M. Simonsick, J. A. Schrack, and D. Kuh, “Age-Related Change in Mobility: Perspectives From Life Course Epidemiology and Geroscience,” J. Gerontol. A Biol. Sci. Med. Sci., vol. 71, no. 9, pp. 1184–1194, Sep. 2016, doi: 10.1093/gerona/glw043.
[3] S. La Grow et al., “The Impact of Mobility on Quality of Life Among Older Persons,” J. Aging Health, vol. 25, no. 5, pp. 723–736, Aug. 2013, doi: 10.1177/0898264313490198.
[4] P. Rasi-Heikkinen and M. Doh, “Older adults and digital inclusion,” Educational Gerontology, vol. 49, no. 5, pp. 345–347, 2023, doi: 10.1080/03601277.2023.2205743.
[5] R. Bushra, N. Aslam, and A. Y. Khan, “Food-Drug Interactions,” Oman Med. J., vol. 26, no. 2, pp. 77–83, Mar. 2011, doi: 10.5001/omj.2011.21.
[6] J. K. Tan, and A. Sato, “Human-robot cooperation based on visual communication”, *International Journal of Innovative Computing, Information and Control*, vol.16, no.2, pp. 543-554, 2024.
[7] R. Wudhikarn, P. Charoenkwan, and K. Malang, “Deep Learning in Barcode Recognition: A Systematic Literature Review,” IEEE Access, vol. 10, pp. 8049–8072, 2022, doi: 10.1109/ACCESS.2022.3143033.
[8] H. Zhang, G. Shi, L. Liu, M. Zhao, and Z. Liang, “Detection and identification method of medical label barcode based on deep learning,” in Proc. 2018 8th Int. Conf. Image Process. Theory Tools Appl. (IPTA), Xi'an, China, 2018, pp. 1–6, doi: 10.1109/IPTA.2018.8608144.
[9] T. Kamnardsiri, P. Charoenkwan, C. Malang, and R. Wudhikarn, “1D Barcode Detection: Novel Benchmark Datasets and Comprehensive Comparison of Deep Convolutional Neural Network Approaches,” Sensors, vol. 22, no. 22, p. 8788, 2022, doi: 10.3390/s22228788.
[10] S. M. Youssef and R. M. Salem, “Automated barcode recognition for smart identification and inspection automation,” Expert Syst. Appl., vol. 33, no. 4, pp. 968–977, 2007, doi: 10.1016/j.eswa.2006.07.013.
[11] C. Wang, N. Guevara, and D. Caragea, “Using Deep Learning to Improve Detection and Decoding Of Barcodes,” in Proc. 2022 IEEE Int. Conf. Image Process. (ICIP), Bordeaux, France, 2022, pp. 1576–1580, doi: 10.1109/ICIP46576.2022.9897371.
[12] E. Sulistyo, M. I. Ilham, D. Prameswari, and Yudhi, “Barcode Scanning Conveyor Automation System for Expedition Goods,” emitor, vol. 25, no. 1, pp. 86–96, Apr. 2025, doi: 10.23917/emitor.v25i1.6164.
[13] “Depth camera d415,” Oct. 2024. [Online]. Available: https://www.intelrealsense.com/depth-camera-d415/
[14] “OpenCV 4.0,” Nov. 2018. [Online]. Available: https://opencv.org/blog/opencv-4-0/
[15] “ZBar bar code reader,” Jul. 2012. [Online]. Available: https://zbar.sourceforge.net/
[16] “Barcode Detection Annotated Dataset,” 2021. [Online]. Available: https://www.kaggle.com/datasets/whoosis/barcode-detection-annotated-dataset/data
[17] “Intel RealSense SDK 2.0,” Jun. 2024. [Online]. Available: https://www.intelrealsense.com/sdk-2/
[18] F. T. T. Pham, K. Koyama, and Y. Tanjo, “Segmentation of displayed objects using an RGB-D camera”, Joint Conf. of Electrical, Electronics and Information Engineers (JCEEE), pp. 93-94,2024.
[19] ROBOTIS, “DynamixelSDK,” Apr. 2025. [Online]. Available: https://github.com/ROBOTIS-GIT/DynamixelSDK/releases
[20] A. Aristidou and J. Lasenby, “FABRIK: A fast, iterative solver for the Inverse Kinematics problem,” Graph. Models, vol. 73, no. 5, pp. 243–260, 2011, doi: 10.1016/j.gmod.2011.05.003.