Object Recognition Supported by User Interaction for Service Robots *

Yasushi Makihara, Masao Takizawa, Yoshiaki Shirai, Jun Miura and Nobutaka Shimada
Dept. of Computer-Controlled Mechanical Systems, Osaka University
2-1, Yamadaoka, Suita, Osaka 565-0871, JAPAN
{makihara, takizawa, shirai, jun, shimada}@cv.mech.eng.osaka-u.ac.jp

Abstract

This paper describes an interactive vision system for a robot which finds an object specified by a user and brings it to the user. The system first registers object models automatically. When the user specifies an object, the system tries to recognize the object automatically. When the recognition result is shown to the user, the system interacts with the user by speech. The user may help with providing additional information such as which object is the correct one (for example, "I want the left object"), the relative position of the object (for example, "It exists on the left of object A", "It exists behind object B"), or what the system mistakes the object for (for example, "You mistake object A for object B"). Then, according to the advice, the system recognizes it again. Experiments with real refrigerator scenes are shown.

Keywords: Object Recognition, Using a Dialog, Service Robot

1 Introduction

In the recent aging society, it is increasingly demanded to realize service robots for taking care of elderly people. We focus on a house helper robot bringing various goods to a physically handicapped user who can't walk around but can speak clearly. We suppose that the user can tele-communicate to the robot with a mobile phone, and that the user can look into the refrigerator through the mobile phone's display.

The system has to know the location of a target object. Following methods are considered to achieve this. In first method, the user points the location of an object with a pointing devise. However, the devise is less common than the mobile phone. In second method, the system memorizes the location of an object when the system puts it. However, other persons may use the refrigerator. In third method, the system fixes the location of objects in the refrigerator. However, when the fixed location is full of objects, the system can't put the object in other locations.

Above methods have those defects, so we adopt a method that the system recognizes an object in natural scenes in the refrigerator and computes the location of the object. However, the full automatic recognition is difficult because of the variety of diversity of shape, occlusion and illumination condition. Therefore a recognition technique using other clues in addition to visual information is required to deal with such a complicated scene.

Some robots use verbal information in scene recognition. Watanabe et al.[2] proposed a system to recognize flowers and fruits in a botanical encyclopedia using explanation texts attached to each figure. Other methods use the user's advice[3][4]. These methods search for the location where the image features are most consistent with the user's advice. However, the methods don't recover mis-selection. Takahashi et al.[1] proposed a robot with verbal and gestural interactions to directly point out the object location. However, no solution is given to recognition errors. Some methods[5][6] generate a scene explanation based on the successful recognition. When the recognition fails, they say just "Not found." because they assume good segmentation and no trial for mistake recovery is done.

The technical focus of this paper is a solution to recognition errors using a dialog with the user. When the system makes a mistake in recognition, the user gives information of the mistake or a recovery method. If the system discovers a mistake by itself, it may ask for the recovery information by showing result to the user. Then the system can learn to avoid the same error in the future by the user's suggestions.

This paper discusses three issues through an implementation of a service robot which recognizes drinks in a refrigerator: (1) what kind of automatic vision functions are required, (2) how to pick up effective information for scene recognition from the dialog, and (3) how to modify the recognition procedure according to the dialog.

Although the robot should move to the refrigerator and grasp objects, these topics are not dealt with in this paper (see [7]).

2 Overview

Our scene recognition system consists of two major parts: automatic object recognition and recognition assistance based on dialog analysis.

The former part is an bootstrap of recognition process. In order to recognize various objects, the object models should be registered in advance. A database of object models is constructed by registration of useful features for identification. Since the object appearance changes depending on the viewpoint, the features observed from all the considerable viewpoints are registered for one object (see sec. 3).

It is difficult to extract exact object regions from a single feature in the refrigerator scene because of mutual occlusion or the object's color change caused by

*Proc. of *ACCV2002*, Vol. 2, pp. 719-724, January, 2002

the different illumination conditions. Therefore the extracted region is first classified into the object types like "can", "bottle" or "PET bottle" according to the geometric locations of multiple features. This classification is also useful to respond to the user's request like "take a bottle of beer". The classified object region is matched to each registered object model considering color shift and viewpoint change. (see sec. 4).

The automatic recognition may make errors. In order to correct the errors, the system shows the result of the automatic recognition to the user by displaying the extracted region and explanation by speech. For example, (1) "A beer bottle is found in the door pocket". When recognition fails, the system generates the utterance for recovery trial like (2) "Coke is not found but two red objects are found in the right side". There are two requirements in order to generate such a utterance. First the system should know that coke is a can and the color is red. Such a proper noun is registered when the object is stored in the refrigerator. Second, the system should have case study of own mistakes. If it is known that the contour and color of a coke is not always well-extracted, red regions not looking like the can's shape or can-shaped regions not looking red can be the second best candidates.

Monitoring the displayed regions, the user selects one from the candidates and tells the system. If no correct candidate is present, the user gives helpful information of the scene based on the current recognition: (3) "It is behind the red things". This implies that occlusion causes the recognition error. Then the system inspects the specified region more carefully using weaker edges or tolerant similarity of color and texture.

While appropriate image processing is invoked according to the user's suggestion, the system reaches the correct recognition. If not, the dialog is continued. When correctly recognized, the system also learn by registering the new sample feature. This recognition assistance using dialog is described in sec. 5.

3 Registration of object models

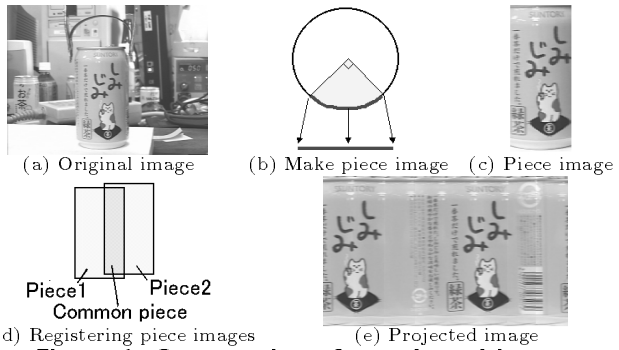
In order to recognize an object from any direction, the system observes it from many directions and extracts its features at each direction and registers them. Features consist of the size of an object, representative colors, and secondary features such as the color, the position, and the size of uniform color region. In this paper, we propose a strategy to register minimum number of features hierarchically for discriminating the object. Therefore if there are no objects with the same representative colors, the system registers only the representative colors. Every time an object with the same feature is registered, the system adds distinguishable features. We describe a method of registration of object models below.

3.1 Construction of a projected image

Because it is troublesome to extract features from images at many directions, we use one image made by projecting the surface texture of the object from the center axis to the virtual cylinder and by developing the cylinder into a rectangle plane. This is called "projected image". The approximate projected image is constructed in the following steps:

1. Take images of the object from 8 directions automatically by rotating it by 45 degrees with a manipulator (Figure 1(a)).
2. Extract the contour of the object in each image by using edges and make a piece of the image from the part of 90 degrees near the center of the object (Figure 1(b), (c)).
3. Connect the adjacent image pieces (Figure 1(d)) and register them so that the common part of them may match the best.

Thus the projected image is constructed (Figure 1(e)).



(a) Original image (b) Make piece image (c) Piece image
(d) Registering piece images (e) Projected image

Figure 1. Construction of a projected image

3.2 Registration of features of objects

In general, the size of the object varies visually at each direction, but assuming that objects are approximately axial symmetric and that the size of the object doesn't vary at each direction, the height and the diameter of the object are computed from the projected image.

On the other hand, the representative colors and the secondary features are dependent of the viewing direction. Instead of registering feature values for every direction, we determine the intervals of the similar feature values. For example, the projected image in Figure 2(a) is divided into the interval I_1 of white and the interval I_2 of blue. An interval is further divided into multiple intervals according to secondary features if distinct secondary features are extracted in the interval. We continue this process until all the intervals are distinguishable from other objects. For example, when object A in Table 1(a) is already registered, object B with the same feature (blue, [white]) as object A is added. Then secondary features of object A and object B are added as shown in Table 1(b) to distinguish object A from object B.

Table 1. Registration of object B ([interval index], representative color, [secondary feature])

(a) Before registering object B with (blue, [white])

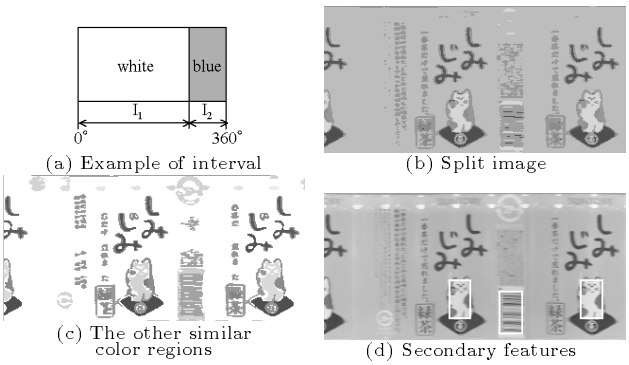
A	([1], yellow), ([2], green), ([3], blue, [white])

(b) After registering

A	([1], yellow), ([2], green), ([3], blue, [white-top-large])
B	([1], blue, [white-middle]), ([2], blue, [white-top-small])

In order to extract those direction-dependent features, the projected image is split into similar color regions by using histograms of YIQ (Figure 2(b)), and the system computes mean μ and variance σ^2 of the largest region at each direction. Because the color in the similar color region varies according to the illumination condition, we assign a range $[\mu - \sigma, \mu + \sigma]$ to the representative color. Then the system extracts the largest region in the other similar color regions (Figure 2(c)) as the secondary feature at each direction. If more than one secondary features are needed, the system extracts necessary number of regions according to priority of the size. We assign proper ranges to

the color, the position, and the size of the secondary feature. Figure 2(d) shows an example of secondary features with white boxes on the projected image.



(a) Example of interval (b) Split image

(c) The other similar color regions (d) Secondary features

Figure 2. Exraction of features

4 Automatic object recognition

We consider two cases for automatic object recognition: (1) the name or the class of the registered object is specified and (2) the system tries to recognize all objects in the refrigerator. In both cases, the system recognizes objects in the following steps:

1. Extract candidate regions for the object.
2. Recognize object types (can, bottle, PET bottle, or unknown object type) and shapes for each region.
3. If the region is recognized as a known object type, classify the region into registered objects.

In the case (1), in step 3 the secondary features are used to recognize the object because the candidate region is extracted with the representative color. Though we can omit step 3 if the system extracts candidate regions with both the representative color and the secondary features, the system can't extract the secondary feature without the object region because the secondary feature includes not only the color but also the position in the object. Therefore we adopt these 3 steps.

4.1 Extraction of candidate regions

When the name or the class of the object is specified, the system extracts regions with the representative colors given by the object models. If the system knows a concrete object name, it uses one model of the object. If the system knows the class of objects such as coffee or orange juice, it uses multiple models of the same class. If the color of extra regions is similar to the representative color, the candidate regions sometimes include the extra regions (Figure 3(b)). In order to separate them from candidate regions, the system splits the regions into similar color regions more precisely. The regions adjacent to the wall of the refrigerator and too large regions are regarded as the background. The resultant candidate regions are shown in Figure 3(c).

If the representative color is similar to that of the background, the system extracts the regions with the secondary feature (Figure 4(b)) and adds the region of representative color around each region. The resultant candidate regions are shown in Figure 4(c). Note that the system can distinguish either the representative color or the secondary feature's color from that of the background because the secondary feature's color is different from the representative color. However, if the secondary feature isn't registered or is hidden by another object, the system can't extract the candidate region. We deal with this problem in sec. 5.

When the system tries to recognize all objects in the refrigerator, it splits all of regions in the refrigerator into similar color regions (Figure 5(b)) because it can't use the representative color of the object. Then the system removes the background in the same way as the first case. The resultant candidate regions are shown in Figure 5(c).


(a) Original image (b) Regions of the representative color (c) Candidate regions

Figure 3. Candidate regions with the representative color

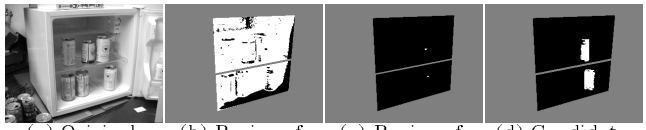

(a) Original image (b) Region of the representative color (c) Region of the secondary feature (d) Candidate regions

Figure 4. Candidate regions with the secondary feature

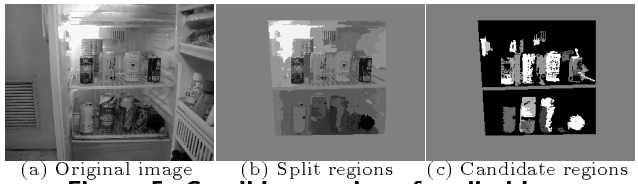

(a) Original image (b) Split regions (c) Candidate regions

Figure 5. Candidate regions for all objects

4.2 Recognition of object types and shapes

The system recognizes object types and shapes by comparing the candidate regions to shape models. The shape models consist of approximate models and detailed models. The approximate models include features of the object types. The detailed models include the size of the object in addition to the features of the object types.

When the system tries to recognize all objects in the refrigerator, it compares the regions to each approximate model because it is inefficient to compare the regions to all the detailed models of the objects. If the region matches to one of the models, the region is regarded as the object type. Otherwise the region is regarded as an unknown object type.

We describe each approximate model below.

A can (Figure 6(a)) has four edges on the upper, lower, left, and right sides of the region. If the edges are extracted and the aspect rate of the rectangle surrounded by the edges is about 2, the system gives a good evaluation value as a can.

A bottle (Figure 6(b)) has two pairs of vertical lines of the neck and the body. If the lines are extracted and the distance between lines of the neck is shorter than that of the body, the system gives a good evaluation value as a bottle.

A PET bottle (Figure 6(c)) consists of three parts: the cap, the label, and the lower part. Because the cap is small and the lower part varies visually according to the amount of the contents, the label is extracted as the

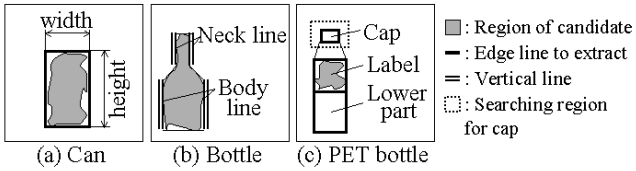


Figure 6. Object models for approximate shapes

candidate region. The system first extracts four edges on the upper, lower, left, right sides of the region. Then it extracts the cap with edges in the searching region above the label, and extracts the lower part with edges under the label. If they are extracted and the spatial relation among them is proper, the system gives a good evaluation value as a PET bottle.

In each case of the models, if most of features of the model are extracted, the evaluation value doesn't descend rapidly due to lack of a few features. For example, we consider the case where the region of a can is slightly hidden at the left side by another object. The system can't extract the left edge, but if the other edges are extracted and the aspect rate is proper, the system gives a moderate evaluation value.

The resultant regions of each object type are shown in Figure 7.

When the name or the class of the object is specified, the system compares the regions to the detailed models. In the detailed model, the size of each part such as the height or width of the can is checked in addition to extracting features of the approximate model. If the object model doesn't include secondary features, the system regards the region as the object immediately.

Figure 7. Region of each object type

4.3 Recognition of objects

When the name or the class of the object is specified, the system searches the object models for the intervals whose representative color is similar to that of the extracted region. Next the system extracts secondary features registered at the intervals from the extracted region. If the all the secondary features of one of the intervals are extracted, the region is regarded as the object with the interval. Otherwise the system regards the regions as the object candidates with the intervals of which the most features are extracted. In practice, some intervals have a secondary feature in common, so the system extracts it for once.

For example, when the objects in Table 1(b) are already registered, the system tries to recognize object A. If the yellow region is extracted, it is regarded as object A without extracting secondary features. If the blue region is extracted, the system tries to extract the secondary feature [white-top-large]. If it is extracted, the region is regarded as object A.

When the system tries to recognize all objects in the refrigerator, the system computes the representative color of each object region and finds the intervals whose representative color in all the object models is similar to the representative color of the extracted region. Then the system recognizes each object in the same way as the first case.

5 Using dialogs

This section describes dialogs to detect the target object and how to recover errors using the dialogs.

5.1 Dialogs to detect the target object

First, if the user asks what objects are in the refrigerator, the system tells all the found objects. If the user wants one of them, he selects it. If he wants another object or the result of recognition is wrong, the dialogs are similar to those described above.

Second, we deal with a case where the user specifies the name or the class of an object to get. If the target object is registered, the system tries to detect it(see sec. 4) and shows the result. Otherwise, the system tries to detect it using a dialog (see sec. 5.2.1).

We classify the results of the recognition into the following only four cases:

1. One object is found.
   The system says, "I have found one object. Is it all right?" If correct, the user says, "Yes." Otherwise, (a) the user specifies the location of the target object, and if necessary, (b) the user can make the system learn to avoid the same mistake.
   (a) How to specify the location
   The specification of the location depends on the following two cases:
   (1) The target object is not hidden by another object.
   The user specifies the absolute location of the target object in the refrigerator (e.g. "At the right of the upper shelf"), or the relative location (e.g. "At the right of object A"). For this case the system tries to find it in the specified location (see sec. 5.2.2).
   (2) The target object is hidden by another object.
   The user specifies the relative location, (e.g. "At the back of object A"). If object A is registered, the system can find it. If the target object is not completely hidden by object A, the system can find the target object by searching carefully near the object A (see sec. 5.2.3). Otherwise, the system first removes the object A and then finds the target object. If the object A is not registered, the system asks the location. Then the system grasps it by feel and removes it (this case is not dealt with in this paper).
   (b) Learning
   If the user corrects the result, the system learns by modifying the registered information (the learning is described in sec. 5.2.5). For learning the system needs the correct name of the object. If the user has not give the name, the system asks it. Note that this dialog for learning starts after the system begins to get the target object, because getting it is more important than the learning.
2. More than two objects are found.
   The system says, "I have found n objects", and selects one object among n objects and says, "I will bring this. Is it all right?" The user's answer is divided into the following four cases.
   (a) If it is correct, the user says, "Yes."
   (b) If the user wants to select another object, the user specifies it in the following ways:

(1) Specification of the location
(2) Specification of the type of the object
Because the system has already recognized the type of each found object, it can determine the target object.

(c) If the target object is not found, the user specifies the location of the target object (see 1(a)).
(d) If some of the found objects are not correct, the user can make the system learn (see sec. 5.2.5).

After the system begins to get the target object, the system asks whether unselected objects are the target objects. If the user tells another name, the system learns to avoid the same mistake (see sec. 5.2.5).

3. The target object is not found although the candidate region is found.
The system says, "I have no confidence. Is this all right?", in order to ask the user to look at the candidate region carefully. This case is further divided into the following three cases.
(a) If it is correct, the user says, "Yes." In order to avoid the same mistake, the system learns (see sec. 5.2.5).
(b) If the target object overlaps with another object of the same color, they may be regarded as the same object. In this case the user helps the system in one of the following ways:

(1) The user selects the front object. The system tries to recognize the target object considering the overlap (see sec. 5.2.4).
(2) The user selects the back object. Because the system has to recognize the front object in order to recognize the back object, it asks what the front object is. Then the system recognizes the front object and tries to recognize the back object carefully (see sec. 5.2.4).
(3) The user points out that two objects overlap. Then the system asks which object to bring. Depending on the user's response, the system acts just as (1) or (2).

(c) If the candidate region is something else (the background or an unregistered object), the user specifies the location of the target object (this case is handled in the same way as 1(a)) or gives the name of the unregistered object (in this case the system needs to registers it, but it is not dealt with in this paper).
4. Neither the target object nor candidate regions are found.
The system says, "I have not found it. Where is it?" While waiting for the user's response, the system tries to recognize all objects in the refrigerator (described in sec. 4.1). If the user specifies the location before the system finishes recognizing all objects, this case is handled in the same way as 1(a). Otherwise, the system shows the found objects so that the user may specify easily. The user can specify the location relative to the found object (see sec. 5.2.2 and 5.2.3).

5.2 How to recover errors using dialogs

This section describes how to recover errors using dialogs by described in section 5.1.

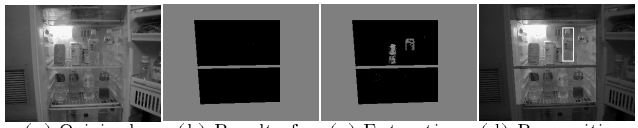
5.2.1 Recognition of the unregistered object

First, the system asks the representative color. Second, the system extracts candidate regions using the specified representative color in the same way as case (1) in sec. 4. If the representative color is similar to that of the background, the system asks the location (see sec. 5.2.2). Third, the system recognizes object types of the extracted candidate regions using approximate models in the same way as the case (2) in sec. 4. Last, the system selects the region which worst matches the registered object models as the target object by the following method. The system searches the object models for the intervals whose representative color is similar to the specified color. If there are no intervals, the system regards the region as the target object. Otherwise, the system tries to extract the secondary features. Then the system regards the region with the least number of extracted features as a candidate of the target object.

5.2.2 Recognition for a given location of the target object

Recognition method depends on whether the representative color of the target object is similar to the background color or not.

1. The representative color is different from the background
The recognition error may have been due to the color shift of the target object by the change of the illumination condition. Therefore the system tries to find the target object in the specified location by extending the representative color range. If no objects are found, the system tries again by further extending the color range.
Figure 8 shows an example.



(a) Original image (b) Result of automatic recognition (c) Extraction in the specified location (d) Recognition result

Figure 8. Recognition for a given location of the target object which color is different from the background color

2. The representative color is similar to the background.
The target object may not be found when no secondary features are found or registered. Therefore the system extracts vertical edges in the specified location, selects a pair which matches the width of the target object, and extracts other edge lines (e.g. the upper and lower end in the case of can) between selected vertical edges. If this is successful, the system regards it as the target object.
Figure 9 shows an example.

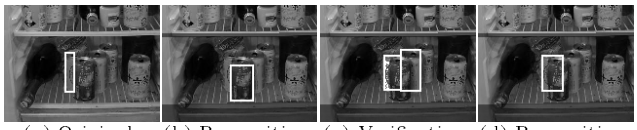
(a) Result of automatic recognition (b)Extraction of both ends (c) Recognition result

Figure 9. Recognition for a given location of the target object which color is similar to the background color

5.2.3 Recognition of the object hidden by another object with the different color

The system first finds the front object and extracts vertical edges near it. Regarding each vertical line as a

side edge of a back object, the system predicts another side edge, and extracts other edge lines (e.g. the upper and lower ends in the case of can). If this is successful, it regards the regions surrounded by them as the candidate regions, and then verifies them by the representative color.
Figure 10 shows an example. The white rectangle in Figure 10(a) shows the visible part of the target object. In Figure 10(c), the white region in the left white rectangle shows the region with the representative color.



(a) Original image (b) Recognition of the front object (c) Verification by the representative color (d) Recognition result

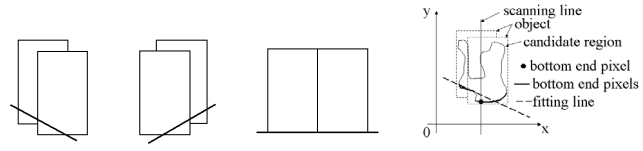
Figure 10. Recognition of the hidden object

5.2.4 Recognition of two overlapping objects of the same color

The system needs to extract the region of the front object. Because the objects are viewed from upper position, the bottom boundary of the front object is projected at the lower in the image than that of the back object (Figure 12). Therefore the shape of the bottom boundary of two overlapping objects is divided into three types depending on the object configuration as shown in Figure 11: (1) tilted to right (i.e. the right part is front), (2) tilted to left (i.e. the left part is front), (3) almost horizontal (i.e. side by side). The system approximately estimates the slope of the bottom boundary as follows. First the bottom end pixel for each x coordinate is obtained by scanning the region along the y axis (Figure 11(d)). Then the slope is estimated by line fitting. According to the object configuration recognized by the slope, the system determines the search area of the occluding boundary edges between the two objects. Based on the extracted edges the front object region is decided.

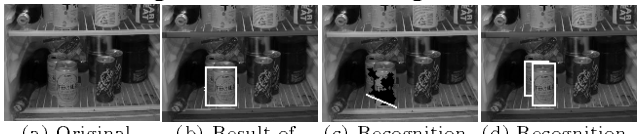
If the back object is specified, the boundary segments of the back object are extracted based on the obtained front object. Because the size and shape of the object is known, the occluded object region can be estimated.

Figure 12 shows an example.



(a) Tilted to right (b) Tilted to left (c) Almost Horizontal (d) Bottom end pixels

Figure 11. Object configuration



(a) Original image (b) Result of automatic recognition (c) Recognition of the object configuration (d) Recognition result

Figure 12. Recognition of two overlapping objects of the same color

5.2.5 Learning by dialogs

This case is divided into two cases: one case that the system regards the target object (A) as the different object, and the other case that the system does not regard the found object, which is A, as it. In both cases, the reason of the recognition error is that the color range of the feature of A (let it denote R_A) is shifted to a range (let it denote R_I) not overlapping with R_A. Therefore the system needs to modify R_A once the error occurred.

The system first modifies R_A for all the intervals of R_A so that the new R_A contains R_I and searches objects which the system can not distinguish from A because of this modification. If such objects exist, the system extracts new features to distinguish them in the projected images, and learns their features. Note that the system can extract the new features using the projected images in the database.

6 Conclusion

In this paper, we show that the system can recognize the hidden object and the object which color is similar to the background color by using a dialog. Furthermore, we show that the system can learn new features of the objects by using a dialog and succeed in recognition of them. By showing above, we show the effectiveness of using a dialog for recognition and learning.

Future works are as follows.

- Recognition of objects which type is different from a can, a PET bottle and a bottle
- Recognition of objects in a shelf besides in a refrigerator
- How the system interacts with a user in the case where a user can not look at a display

References

[1] T. Takahashi, S. Nakanishi, Y. Kuno and Y. Shirai, "Human-Robot Interface by verbal and Nonverbal Communication", Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, pp.924-929, 1998.

[2] Y. Watanabe, M. Nagato and Y. Okada, "Image Analysis Using Natural Language Information Extracted from Explanation Text", Proc. of MIRU'96 Vol.2, pp.271-276, 1996 (in Japanese)

[3] S. Wachsmuth and G. Sagarer, "Connecting Concepts from Vision and Speech Processing", Workshop on Integration of Speech and Image Understanding, 1999

[4] U. Ahlrichs, J. Fischer, J. Denzler, C. Drexler, H. Niemann, E. Noth and D. Paulus, "Knowledge Based Image and Speech Analysis for Service Robots", Workshop on Integration of Speech and Image Understanding, 1999

[5] K. Fujii and K. Sugiyama, "A Method of Generating a Spot-Guidance for Human Navigation", Trans. of IEICE D-II Vol.J82-DII No.11, pp.2026-2034, 1999 (in Japanese)

[6] M. Iwata and T. Onisawa, "Linguisitic Expressions of Picture Information Considering Connection between Pictures", "A Method of Generating a Spot-Guidance for Human Navigation", Trans. of IEICE D-II Vol.J84-DII No.2, pp.337-350, 2001 (in Japanese)

[7] Y. Makihara, M. Takizawa, K. Ninokata, Y. Shirai, J. Miura and N. Shimada, "An Assistant Robot Acting by Occasional Dialog -Object Recognition and Manipulation Using Dialog with User-", Proc. of ROBOMEC'01, 2001(in Japanese).