# A Service Robot Acting by Occasional Dialog

- Object Recognition Using Dialog with User and Sensor-Based Manipulation -

Yasushi Makihara, Masao Takizawa, Kazuo Ninokata, Yoshiaki Shirai, Jun Miura, and Nobutaka Shimada: Osaka University, 2-1 Yamadaoka, Suita, Osaka 565-0871

**Abstract:** In this research, we develop a service robot which can bring a user-specified object from a refrigerator to a user by a dialog with him or her. This paper describes two of the functions necessary for that robot: one is to recognize objects in the image by using a dialog with a user; the other is to manipulate objects by using force and tactile sensors under the uncertainty of vision information.

**Key Words:** Service robot, Object recognition, Dialog, Sensor-based manipulation

## 1 Introduction

Nowadays, there is a growing necessity of robots for welfare in this aging society and, therefore, various researches on service robots have been performed[1][2]. It is important for such a service robot to bring a user-specified object. This paper describes a service robot which can bring can, bottle, and PET bottle from the refrigerator.

In order to bring a target object, the robot has to know the position of the object. One method to compute the object position is recognizing the object using vision. In a complicated scene like the inside of a refrigerator treated in this paper, however it is sometimes difficult to automatically recognize an object because of the partial occlusion of objects and the change of the lighting condition. In such cases, we need to use information other than the vision. There are some researches on connecting visual information and verbal information. Some methods[3][4] generate a scene explanation based on the visual recognition. Watanabe et al.[5] 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[6][7]. These methods search for the position where the image features are most consistent with the user's advice. However, the methods don't recover recognition error using verbal information. Takahashi et al.[2] proposed a robot with verbal and gestural interaction to directly point the object position. However, they use verbal information in order to choose one objects from multiple extracted objects. In this research, we use verbal information to help the recognition by using a dialog with a user when the system fails in recognition.

Moreover, the system has to manipulate target objects. Though the system recognizes the shape and the position of the object, they usually contain uncertainty. Therefore the system analyzes the tactile state based on sensor information in order to grasp and manipulate objects reliably.

## 2 Overview

Our service robot consists of three major parts: object recognition, recognition assistance by dialog, and sensor-based object manipulation. First, a user watches the image in the display and specifies an object with its name or color by speech. Next, the robot detects the object based on color and shape and computes its 3D position. If multiple objects are detected or the system fails in recognition, the system interacts with the user in order to get helpful information. Moreover, if necessary, the robot learns an object's color through dialog in order to improve the reliability of the recognition in subsequent operation after this. Lastly the robot grasps the object by using the force and tactile sensor and hands it over to the user.

The overview of our system is shown in Figure 1. Currently, the robot arm is fixed at the desk. The host computer processes the image, the dialog, and the information sent from the arm controller and sends a command to the arm controller based on the processed result. The arm controller controls the robot arm using sensor information. Figure 2 shows the flow of operations of our system.

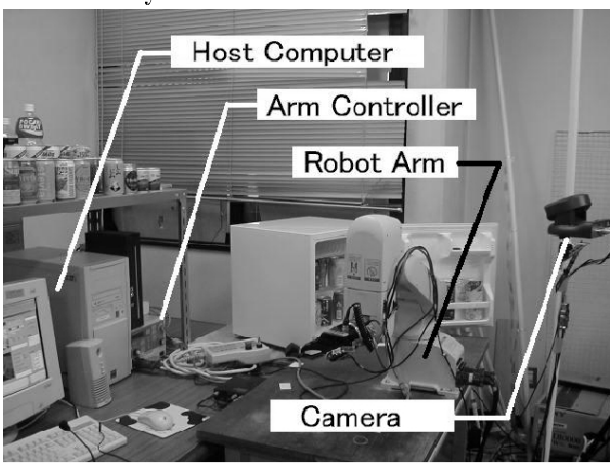
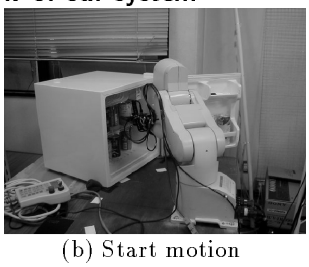



**Figure 1. Overview of our system**


(a) Recognize objects



(b) Start motion

(c) Grasp an object

(d) Hand over to a user

**Figure 2. Flow of the operations of our system**

## 3 Automatic object recognition

We consider two cases for automatic object recognition: (1) when the name or the color name of an object is specified, the system tries to recognize it and (2) when a user asks what objects are in the refrigerator, the system tries to recognize all objects. In both cases, the system recognizes objects in the following steps:

1. Extract candidate regions for the object.
2. Determine object types (can, bottle, PET bottle).
3. Recognize objects.
4. Compute the 3-D position of the object.

### 3.1 Extraction of candidate regions

In order to extract candidate regions, the representative color for each object is registered in advance. The representative color is defined as a range in the YIQ-space. When the name of an object is specified, the system retrieves the representative color of the object and extracts regions with that color. When the color name of the object is directly specified, the system retrieves a color range corresponding to that color name and extracts regions with that color. Such a color-based extraction may extract regions other than objects (see Figure 3(1b)). We eliminate the regions which are adjacent to the wall of the refrigerator and too large regions, as the background. The resultant candidate regions are shown in Figure 3(1c).

When the system tries to recognize all objects in the refrigerator, it segments all of the area in the refrigerator into uniform color regions in the following steps[8][9].

1. Calculate the histogram of each of YIQ value for all of the area in the refrigerator.
2. Compute the most prominent valley in all the histograms and segments the regions with the valley.
3. Repeat the above two steps for each segmented regions while prominent valleys exist.

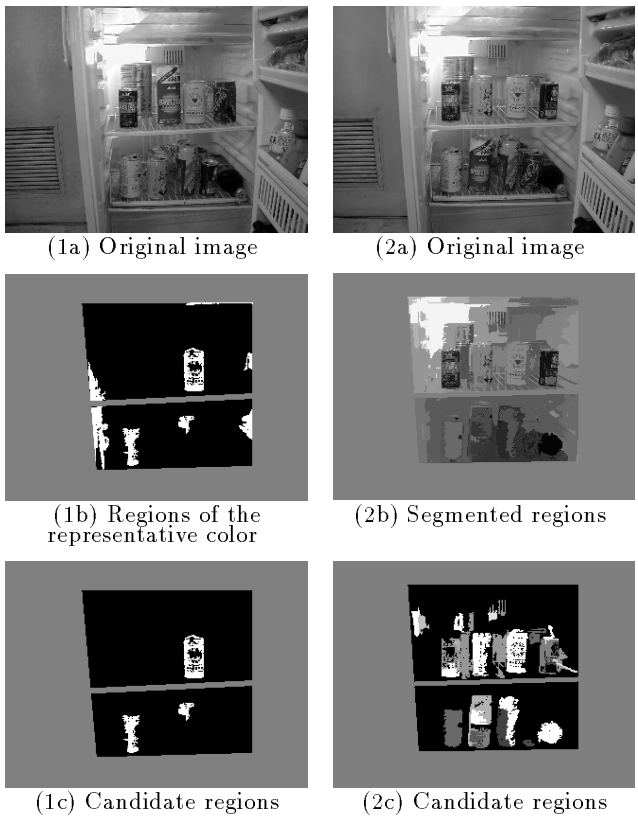
The resultant uniform color regions are shown in Figure 3(2b). Then the system removes the background in the same way as the first case. The resultant candidate regions are shown in Figure 3(2c).

If two objects of the same color are too adjacent to each other, the system extracts them as one connected regions (see Figure 4(b)). Therefore, if the system extracts a region whose width is approximately equal to that of two adjacent objects, the system has to divide it into two regions with the proper vertical line (see Figure 4(c)). The system determines the line in the following steps. First, the system makes a histogram by projecting the region to the horizontal axis. Then, the system computes the most prominent valley of the histogram near the center because we currently deal with the case of two adjacent objects. The resultant regions are shown in Figure 4(d).

### 3.2 Determination of object types

The system determines object types by comparing the candidate regions to shape models. We describe each model below.

A can is regarded as a rectangle in the image. Therefore, we search for the four edges around the candidate regions (see Figure 5(a)). If the edges are extracted and the aspect ratio of the rectangle is approximately 2.0, the system gives a high evaluation value as a can.



(1a) Original image (2a) Original image

(1b) Regions of the representative color (2b) Segmented regions

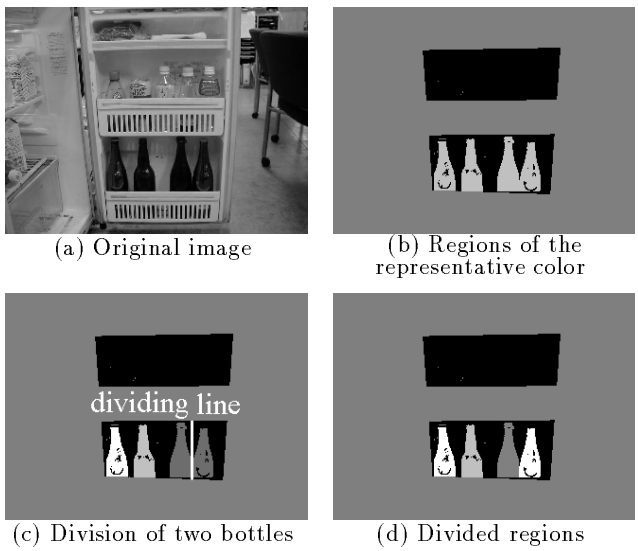
(1c) Candidate regions (2c) Candidate regions

left side: case(1), right side: case(2)

**Figure 3. Candidate regions**

A bottle has two pairs of vertical lines of the neck and the body (see Figure 5(b)). First the system computes the direction and the curvature on each boundary point of the candidate region. Next the robot extracts a series of boundary points as a vertical line, if the direction of the point is nearly vertical and the curvature of it is small. If the lines are extracted and the distance between lines of the neck is shorter than that of the body, the system gives a high evaluation value as a bottle.

A PET bottle consists of three parts: the cap, the




(a) Original image (b) Regions of the representative color

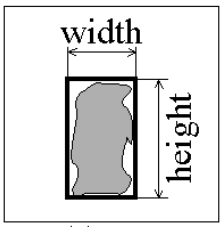
(c) Division of two bottles (d) Divided regions

**Figure 4. Division of two adjacent objects**

label, and the lower part (see Figure 5(c)). Because the cap is small and the view of the lower part varies according to the amount of the contents, the label is extracted as the candidate region. The system first extracts four edges around the candidate region. Then it extracts the cap with edges in a search 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 high evaluation value as a PET bottle.

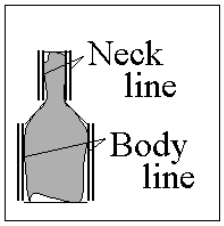
After computing those evaluation values, the system regards the region as the object type with the highest value. If all of the evaluation values are low, the system regards the region as an unknown object.

The resultant regions of each object type are shown in Figure 5(d)(e)(f).



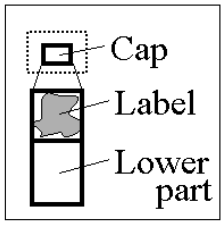

(a) Model of a can

(d) A region of a can




(b) Model of a bottle

(e) A region of a bottle




(c) Model of a PET bottle

(f) A region of a PET bottle

■: Region of candidate =: Vertical line
−: Edge line to extract ⬚: Searching region for cap

**Figure 5. Region of each object type**

### 3.3 Recognition of objects

The system recognizes objects by comparing each extracted regions to color models belonging to each recognized object type. The color model consists of multiple sets of the percentages of the following 13 colors in each object: 10 colors of Munsell color specification system [10], white, gray, and black.

First the system computes a set of 13 percentages in each region. Next the system computes the evaluation value for each region by comparing it to color models. The more corresponding percentages resemble, the higher evaluation value is given.

When the name or the representative color of the target object is specified, the system compares each extracted region to color model of the target only. Then the system regards the regions whose evaluation values are higher than a threshold as the target object.

When the system tries to recognize all objects in the refrigerator, the system compares each extracted region to color models of all registered object. Then the system regards each region as the object with the highest evaluation value in each region.

### 3.4 Computation of the 3-D position of objects

Because the system extracts the region of the object, it knows the bottom position of the object in the image. In this research, we calibrate camera parameters in advance and we suppose that the height of the shelves in the refrigerator, that is, the height of the bottom of the object, is given. First the 3-D position of the bottom point of the object in the image is computed as the intersection of the plane of the shelf and the line of sight that passes the bottom point. Then the system computes the 3-D position of the center point of the object using the height and the radius of the object. Note that the height and the width of the target object are registered in advance.

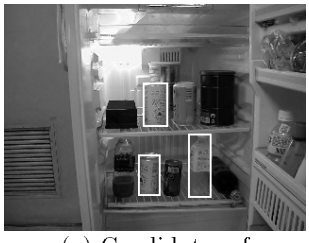
## 4 Recognition supported by dialog

When the system cannot recognize the target object automatically, it uses dialog with the user to obtain more information for recognition. Furthermore, the system sometimes recognizes an object as the target object mistakenly due to incomplete model. When the user points out the mistake, the system refines color models in order to avoid the same mistake. We use ViaVoice by IBM for speech recognition.

### 4.1 Selection of the target object from multiple candidates

When the system extracts multiple candidates of the target object, it cannot select the target object automatically. In such a situation, the system asks the user to specify the location of the target object.

Figure 6 shows an example. In Figure 6 (a), because some candidates of the target object are extracted, the system tells the result that two cans and one PET bottle are found and asks which the system should bring the user. Figure 6 (b) shows the result after the can on the upper shelf is specified.



(a) Candidates of the target object

(b) Recognition result

**Figure 6. Selection of the target object from multiple candidates**

### 4.2 Recognition of the object partially occluded by another object

Because there are many objects in a refrigerator, an object is often occluded by another object. When the target object is occluded, the system cannot recognize it automatically. This case is divided into the following two cases and the system tries to recognize the target object according to user's utterance.

#### 4.2.1 Recognition of the object occluded by another object of the different color

Because the visible part of an occluded target object is too small (see Figure 7(b)), the system cannot find the object. In such a situation, the system asks the user its location and the user answers the name of the occluding object. Then the system tries to recognize the occluded object by the following method.

First the system detects the occluding object (see Figure 7(c)). Second the system searches the both sides of the occluding object for the region of the representative color of the target object and extracts a vertical edge there. Third the system regards the vertical edge as a side edge of the occluded object and predicts the location of another side edge. If other edge lines (e.g. the upper and lower ends in the case of can) are extracted, the system regards the regions surrounded by them as the target object (see Figure 7(d)).

(a) Original image (b) Visible part of the target object

(c) Recognition of the occluding object (d) Recognition result (Broken lines show occluded edges)

**Figure 7. Recognition of the occluding object**

#### 4.2.2 Recognition of two overlapping objects of the same color

The system regards two overlapping objects of the same color as one big object (see Figure 9(b)). The system asks whether it is the target object because the evaluation value as the target object is low. When the user gives the information that two objects overlap, the system tries to recognize each object by the following method.

First, the system tries to extract the region of the occluding object. Because the objects are viewed from a upper position, the bottom boundary of the occluding object is projected at the lower position in the image than that of the occluded (see Figure 9(a)). Therefore the shape of the bottom boundary of two overlapping objects is divided into three types depending on the configuration of the objects: (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 by line fitting (see Figure 8). According to the 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 occluding object region is decided (see Figure 9(d)).

If the occluded object is specified, the system recognizes it in the same way as sec. 4.2.1.

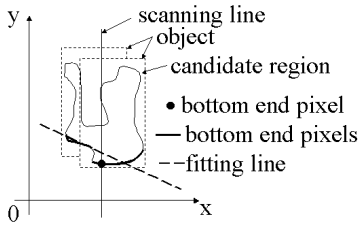
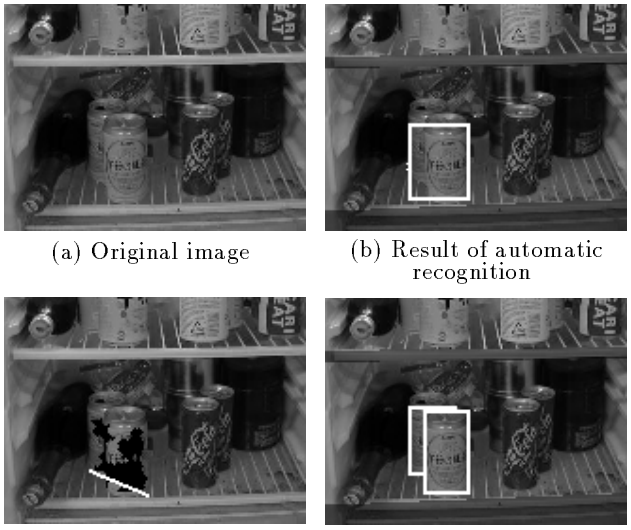



**Figure 8. Line fitting**



(a) Original image (b) Result of automatic recognition

(c) Recognition of the configuration of objects (d) Recognition result

**Figure 9. Recognition of two overlapping objects of the same color**

### 4.3 Refinement of color models

Because of the change of the lighting condition, the system sometimes recognizes an object (let it denote A) as the target object (let it denote B). When the user points out the mistake, the system adds two things to the database in order to avoid the same mistake.

First the system learns that A resembles B. Next the system computes the evaluation value (see sec. 3.3) as A for the region regarded as B. If the evaluation value as B is higher than that as A, the system adds the percentages of the 13 colors in the image to color model of A as a new variation of color appearances.

After this, when A or B is specified, the system can avoid the same mistake in the following method. First

the system computes the evaluation values as both A and B for each extracted region. Then the system regards it as the object with higher evaluation value than that as another one.

## 5 Object manipulation

### 5.1 Hand Position Control Using Force and Tactile Sensors

Object position determined by vision usually includes uncertainty. Such an uncertainty may cause undesirable hand movements such as hitting the target or adjacent objects by its fingers. To overcome this problem, we use both force sensor and tactile sensors to appropriately adjust the hand position for reliable grasping of the object.

Figure 10 shows the sensor placement on the hand. The force sensor is used for determining if an object hits the hand. It is also used for detecting the user holding a grasped object when the robot hand it over to the user. Tactile sensors are attached to the fingertips and the inside of the fingers. The tactile sensors at the tips detect the contact with an object. The ones at the inside determine whether the hand is firmly holding an object.

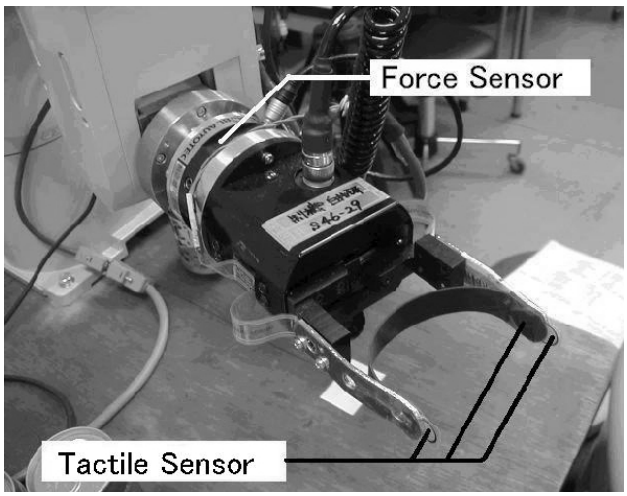



**Figure 10. Force sensor and tactile sensor**

When a target object stands in isolation from other objects, contact information of the fingertips is directly used to adjust the hand position for grasping. When there are other objects adjacent to the target object, however, only contact information is not enough; the finger may have contact with an adjacent object. To distinguish between a contact with the target and one with other objects, we additionally use the surface orientation of the object in contact, which is estimated by two consecutive finger positions during a surface-following movement of the hand.

We analyzed the possible contact states and transitions between them[11] for the case where the target object is a can and there is at most one adjacent can to the target object. Figure 11 shows the six possible adjacency patterns. We also assume that the error in object position calculated by vision is less than the radius of the target can. Based on the analysis and the assumption, for each state, we selected the motion to

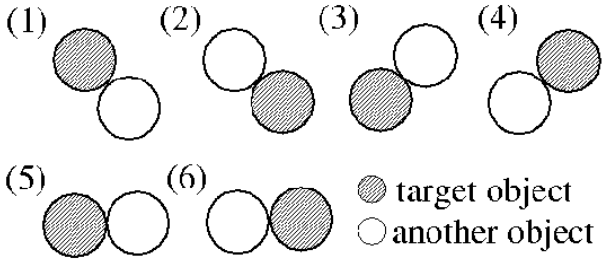



**Figure 11. Adjacency patterns of two cans.**

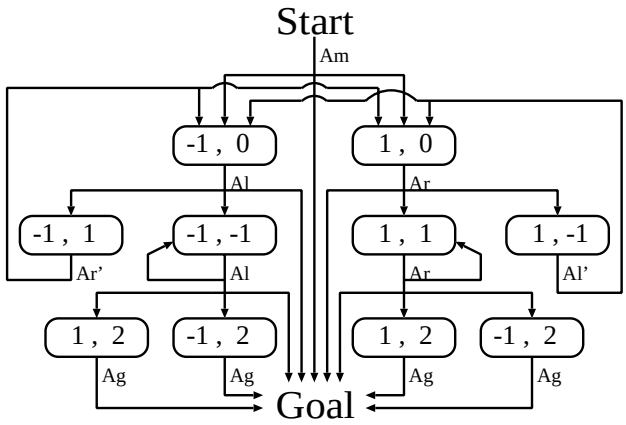



**Figure 12. Connection of contact states and transition**

execute for achieving the goal state (i.e., the hand is at the position for grasping). The analysis result is summarized in Figure 12. The state transition graph in the figure shows the selected motion for each state and the possible transitions to occur by executing the motion. A state $(M, N)$ is defined by a pair of the input of the tactile sensors at the fingertips, $M$, and the estimated surface orientation, $N$. $M$ is one of the following:

- 1: The left fingertip touches the object.
- -1: The right fingertip touches the object.
- 0: No contact.

$N$ is one of the following:

- 1: Surface orientation is positive. The sign of the orientation is determined in the coordinates as shown in Figure 13.
- -1: Surface orientation is negative.
- 2: The fingertip touches a different object from the previous one.
- 0: No surface orientation is available because only less than two contacts have occurred for the current object.

The set of hand motions are defined as follows:

- $A_m$: The initial hand movement towards the object position obtained by vision. The motion stops when (1) a fingertip touches an object, (2) the force sensor detects an object touches the palm, or (3) the hand reaches the object position. For the last two cases, the hand is considered to be ready to close fingers to grasp the object.
- $A_r$: The robot withdraws the hand, shifts it slightly to the right, and moves it forward until one of the above three cases occurs.
- $A_l$: The robot withdraws the hand, shifts it slightly to the left, and moves it forward until one of the above three cases occurs.

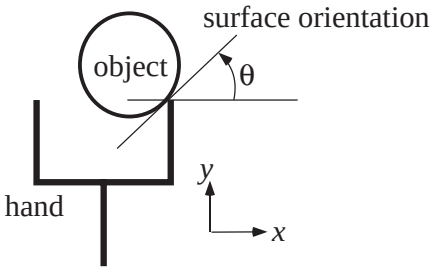



**Figure 13. The coordinates in the hand moving plane.**

- $A'_r$: The robot moves the hand to the right by the distance equal to the radius of the target object, and then moves it forward until a fingertip touches an object.
- $A'_l$: The robot moves the hand to the left by the distance equal to the radius of the target object, and then moves it forward until a fingertip touches an object.
- $A_g$: The robot moves the hand to the object position, since the current contact object is not the target.

Figure 14 shows an example of hand position adjustment for adjacency pattern (1) of figure 11. In figure 14(a), the state is (1, 0) and motion $A_r$ is executed. After two contacts (see Figure 14(b)), the surface orientation is estimated as negative and the state is (1,-1); this means the object in contact is not the target and the robot executes motion $A'_l$. In figure 14(c), the surface orientation indicates the contact with the target object this time (state (1, 1)) and the robot repeats motion $A_r$ until the hand reaches the grasp position (see Figure 14(d)).

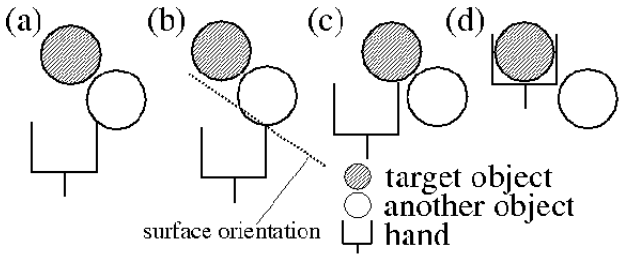



**Figure 14. An example of a grasping motion**

### 5.2 Opening and closing the door and Handing over an object to a user

The manipulator can open and close the refrigerator door using a predetermined hand trajectory specifically designed for the refrigerator used in this experiment. To open the door, the door handle position is first detected by vision, and then the robot places a finger at the handle using force and tactile information to grasp the handle. After open the door, the robot recognizes and picks up the target object and hands it over to the user. When the force sensor detects the user holding the object, the robot opens the hand to release the object. The robot finally closes the door by pushing it until the force sensor detects that the door is certainly closed.

## 6 Conclusion

We have realized a robot that can bring a user-specified object to the user by using following two functions: object recognition in the complicated scene by using dialog and sensor-based object manipulation.

Currently we assume that the user watches the display showing the input image and the recognition result. This enables the user to understand the current situation easily. A future work is to cope with the case where the user does not have such a display; in this case, a function to explain the current situation to the user via voice only is necessary. Another future work is to install current system on a mobile base so that the system can bring objects from a distant refrigerator.

## References

[1] Y. Takahashi, S. Nakanishi, Y. Kuno, and Y. Shirai, "Human-Robot Interface by verbal and Nonverbal Communication", Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 924-929, 1998.

[2] T. Takahashi, T. Komeda, T. Uchida, M. Miyagi, and H. Koyama, "Development of the mobile robot system to aid the daily life for physically handicapped", Proc. of ICMA2000, pp. 549-554, 2000.

[3] 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).

[4] M. Iwata and T. Onisawa, "Linguisitic Expressions of Picture Information Considering Connection between Pictures", Trans. of IEICE D-II Vol. J84-DII No. 2, pp. 337-350, 2001 (in Japanese).

[5] 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).

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

[7] 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.

[8] Y. Shirai "Three-Demensional Computer Vision", Springer-Verlag, pp. 62-68, 1987.

[9] A. Okamoto, Y. Shirai, and M. Asada, "Integration of Color and Range Data for Three-Dimensional Scene Description", IEICE Trans. Inf. and Syst., Vol. E76-D, No. 4, pp. 501-506, 1993.

[10] The Color Science Association of Japan, "Handbook of Color Science", University of Tokyo press, 1989 (in Japanese).

[11] Y. Yokokohji, "Classification of Contact States and Planning of Assembly Sequence", Journal of the Robotics Society of Japan, Vol. 19, No. 2, pp. 15-21, 1993 (in Japanese).