Autonomous Robots 13, 169–184, 2002

Multi-Modal Interaction of Human and Home Robot in the Context of Room Map Generation

SAEED SHIRY GHIDARY, YASUSHI NAKATA, HIROSHI SAITO, MOTOFUMI HATTORI AND TOSHI TAKAMORI
Department of Computer System, Engineering Faculty, Kobe University, Japan
saeed.shiry@gmd.gr.jp

Abstract. In robotics, the idea of human and robot interaction is receiving a lot of attention lately. In this paper, we describe a multi-modal system for generating a map of the environment through interaction of a human and home robot. This system enables people to teach a newcomer robot different attributes of objects and places in the room through speech commands and hand gestures. The robot learns about size, position, and topological relations between objects, and produces a map of the room based on knowledge learned through communication with the human. The developed system consists of several sections including: natural language processing, posture recognition, object localization and map generation. This system combines multiple sources of information and model matching to detect and track a human hand so that the user can point toward an object of interest and guide the robot to either go near it or to locate that object's position in the room. The positions of objects in the room are located by monocular camera vision and depth from focus method.

Keywords: human-robot interaction, human detection, object localization, robot positioning, map generation

1. Introduction

Robots are rapidly moving into everyday life and there is considerable potential for employing robots in the home or office environment. For home robots to be useful and assist people in daily life, they must be easy to communicate with and instruct. When robots are operated inside a home or office environment, they will have to interact with people who are not robotic experts. It seems that the natural interface is the most user-friendly method for interacting between human and robot. The survey by Oestreicher et al. (1999) includes questions about how to communicate with a robot. Results of the survey indicate that most participants preferred speech, followed by touch screen, gestures, and command language.

People incorporate natural language and natural gesture in order to communicate with others and even with pets. People usually adapt themselves to use the proper interfaces for different situations. The type of gesture or verbal phrase differs when ordering or requesting from an adult, a child, or a pet. This property lets a home robot with even little ability in natural interface to be useful, practically.

Research in the area of human-robot interaction includes applications in object manipulation, manufacturing, material handling, micro machining, telerobotics, robot teaching, multi-robot systems, service and home robots, rehabilitation robotics, mobile robots, space robots, and planetary robots (Agah, 2000). Through appropriate communication, robots can share human knowledge and intelligence, and possibly, can integrate the recognition capabilities of humans in order to execute a complex task.

Recently, several systems for robot-machine interaction have been developed using different strategies and approaches. The Jijo-2 robot (Matsui et al., 1999) provides office services, such as answering queries about people's location, route guidance, and delivery tasks by natural spoken conversation with the office dwellers. This system contains sound source detection, navigation, and face recognition behaviors.

The human-robot interaction system introduced by Alford et al. (1999) describes an agent-based software system for robot control which consists of the Human agent and the Self agent. This system provides limited conversation with the robot, including commands to the robot and queries about the robot's abilities and internal state.

HERMES, the humanoid robot (Bischoff and Jain, 1999) combines visual, kinesthetic and tactile sensing for enabling natural communication and interaction with humans. HERMES is capable of speaker-independent speech recognition and speech output. A special behavior-based architecture, based on an understanding of situations is employed to integrate these key technologies.

Torrance (1994) developed a natural language interface for teaching mobile robots the name of places in an indoor environment. It involved the construction of a mobile robot system that performed simple language understanding and generation for the purpose of goal-directed navigation to places described in human terms.

In our approach, we focus on the subject of environment map generation through interaction between human and robot. Map generation is a vital consideration for mobile robots. Although there are some robotic systems capable of autonomously mapping environments by sonar (Schultz and Adams, 1998) or laser (Horn and Schmidt, 1995), an intelligent robot is not always able to process incomplete information from its sensors. In this situation, it is advantagous to query the human for the missing information.

Such an autonomously generated map consists of information about segments of the environment and may be useful for robot navigation. However, it does not provide information about specific objects or places. When a service robot interacts with a human, it is often necessary for the robot to have knowledge about the location of the objects of the environment. Robotic systems that can reliably observe, recognize, and learn about objects in the environment may ultimately provide robotic assistance to human daily life. However, visual recognition and classification of objects into one of many a priori known object types and determining object characteristics, such as pose, is a difficult problem (Aggarwal et al., 1996). Due to limited success in obtaining a general and comprehensive solution to automatic object recognition, it seems that human-robot collaboration can provide intermediate solutions by robot sharing human knowledge.

To enable new robot applications with emphasis on service tasks, it is necessary to develop techniques which allow robots to obtain a similar level of understanding about the environment as that of the human.

In this paper, we describe an interface for human-robot interaction that enables a human to introduce objects in his relatively unstructured room to the robot. The human teaches objects and places by hand gestures and linguistic description. The robot makes a knowledge base of learned information and produces a map of the environment for navigation purposes.

We teach objects of environment to our robot using three different methods. In this system a user points to the interested object by his hand. The robot learns the position of this hand and uses it to estimate the position of the object. Other objects may be introduced with reference to the position of the robot or the position of already learned objects.

1.1. Overview

This paper is organized as follows. Section 2 introduces the architecture of system. Section 3 describes the robot positioning system. This system locates the robot with high accuracy and less time. In Section 4, we introduce our method for finding the position of different objects in the room environment which does not need specific object recognition techniques. It also explains how to use an autofocus system to retrieve depth information from 2D images. Section 5 summarizes basic interactive behavior of this system. Section 6 explains our map generation algorithm and finally, in Section 7, we discuss future work and conclusions.

The results of the actual experiments that have been implemented on our experimental robotic system are presented in each section separately.

2. System Architecture

Figure 1 describes the architecture of our system. This system consists of a Yamabico mobile robotic platform, a powerful host computer, a vision system and a HRPS positioning system (Shiry et al., 1999).

For our research we have been employing a Yamabico robot as shown in Fig. 2. This platform has three wheels: two servo DC driven wheels are fixed at both sides of the mobile robot and one castor is attached at the front side of it. We have equipped this robot with various sensors such as infrared human

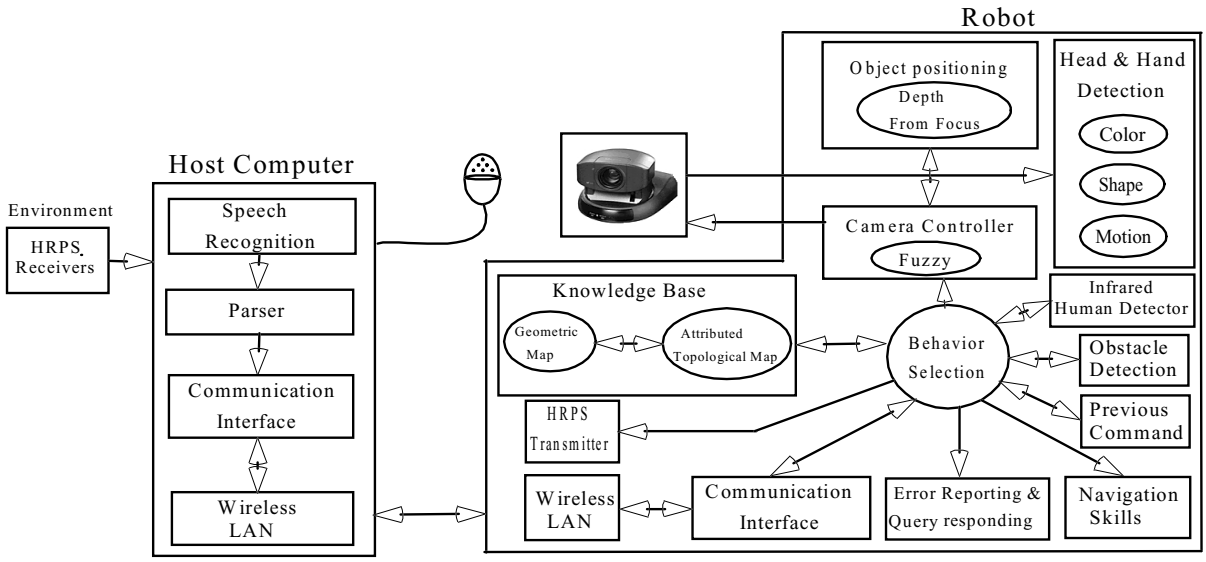

Figure 1. Architecture of our system.

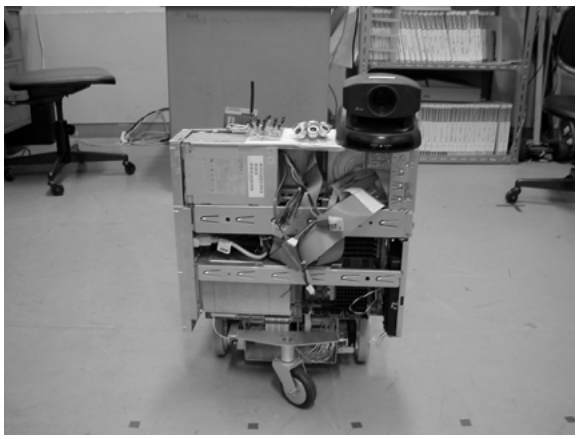

Figure 2. Our robot.

detection, odometric sensor, HRPS transmitter and ultrasonic obstacle avoidance system. It has an on-board computer which is connected to the LAN through wireless Ethernet. This connection provides communication between the robot and the host computer. The robot uses a Sony EVI-D30 PTZ monocular color camera for visual information processing.

In this architecture, the host computer has several key functions. In the positioning system, it collects range data from stationary receivers and computes global position of the robot. In the interaction system, it handles the computational workload of speech recognition. This computer uses a commercially available speech recognition system (IBM ViaVoice) to convert the voice input into a text string. By employing our natural language processing system, the computer parses and decodes this text into robot commands. These commands are sent to the mobile robot by the communication layer and wireless LAN. This design allows large vocabulary recognition and remote communication with the robot.

The robot uses a monocular color camera for image processing. Vision processing includes human detection, hand detection, and measurement of 3D position of objects by the depth from focus method.

For robot positioning we have developed a Home Robot Positioning System or HRPS. This is a very fast robot localization system using ultrasonic and infrared sensors. HRPS plays an important role in our map generation system. Data from HRPS is used for robot positioning, target localization, map generation, and navigation.

3. Robot Positioning

Robot positioning is a major concern in mobile robot navigation. This process is crucial to a home or service robot because most of robot operations are related to position. This position must be found easily and repeatedly. Robot positioning can be done in two basic methods: absolute and relative positioning, or a combination of both. Relative positioning is usually based on dead reckoning (i.e., monitoring the wheel revolutions

to compute the offset from a known starting position). Dead reckoning is simple, inexpensive, and easy to accomplish in real time. The disadvantage of dead reckoning is its unbound accumulation of errors. Absolute positioning methods usually rely on navigation beacons, active or passive landmarks, map matching, or satellite based navigation signals (Borenstein and Feng, 1994).

When a robot does not know its initial position and orientation, it would be very costly for the robot to find its position and orientation autonomously. GPS (Global Positioning System) is one of the solutions to this problem, however it is inaccurate for navigating mobile robots and can only be used outdoors.

Ultrasonic sensors have been widely used in positioning systems for robots (Beom and Cho, 1995; Arai and Nakano, 1983; Kleeman, 1992). These systems may be divided in two categories; one is the area of map building by acquiring complete information about the unknown environment and the other is the localization of mobile robot by using sensory information about the environment (Beom and Cho, 1995).

In recent years many other studies have been made to find the robot position by using a CCD camera and landmarks. In these systems landmarks are placed at prespecified positions in the environment, and the vision sensor which usually is mounted on the mobile robot, obtains the image. The positional relation between camera and landmark is then found by using image processing techniques (Kim and Cho, 1992; Koh et al., 1994). These methods are computationally expensive. They use different patterns for landmark (circular, rectangular or linear) and highly depend on camera calibration and image sensitivity.

In this research, we have developed a Home Robot Positioning System (HPRS) for localization of a mobile robot in an indoor environment. By using HRPS, a mobile robot can find its location inside a known environment with less effort and time. This is actually realized by using a map of the environment and the sensory readings to compute the location of the robot.

HRPS has the following properties:

(a) It is very fast and works in real time without any need to scan the environment. Measuring time is less than 100 ms.
(b) There is no limitation in positioning in different parts of the room.
(c) It is possible to find the heading of the robot without the use of a compass or gyroscope.
(d) It finds the initial position of the robot by itself.
(e) It can be implemented economically.
(f) In this system, we did not use the reflected beam of an ultrasonic wave so problems associated with the reflected beam are cancelled and we can use more reliable data.
(g) It can locate multiple robots.

3.1. Principle of Measurement

The proposed measuring system consists of one transmitter module mounted on the mobile robot, and some receiver modules (6 in our system) that are installed at fixed points in the ceiling of the room (Fig. 3). The transmitter module consists of ultrasonic and infrared transmitter arrays and can produce a burst of both signals at the same time. The receiver module consists of ultrasonic and infrared sensors as well as a necessary circuit for detection and amplification of these signals.

When the transmitter module sends an ultrasonic and infrared pulse couple, the receiver module receives the infrared signal almost instantly. This signal starts a counter at the receiving circuit, which runs at 23.1 KHz. This counter will continue to count until the reception of the ultrasonic signal, which will stop the counter. The value of this counter is proportional to the time of flight of the ultrasonic pulse and can be used to calculate the distance between the receiver and the transmitter as follows:

$$\begin{aligned} t &= n * f - t_d \\ d &= t * v \end{aligned} \tag{1}$$

where f is the frequency of counter clock, n is the timer count, t_d is the delay of detection circuit, t is the total

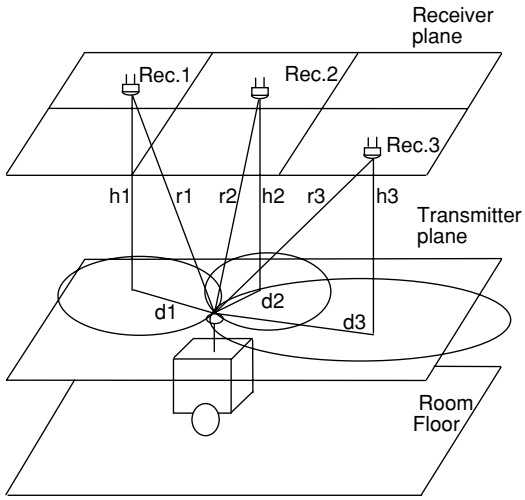

Figure 3. Principle of robot positioning by HRPS.

time of flight of sound, v is the velocity of sound and d is the range data.

For measuring the position of the robot, the host computer sends the "measure" command via wireless LAN to the robot and the robot produces an ultrasonic and infrared signal couple. The host computer then collects the data of the receivers and uses them to calculate the position of the robot. The location and the number of receivers are designed so that at least 3 receivers will catch these signals. The range data can be read every 100 ms.

As the receivers are fixed to a plane with known heights and the transmitter is fixed at a known point on the robot, the vertical distance between the transmitter and receivers are known a priori. Thus, the problem of localization of the robot can be done in two dimensions. As shown in Fig. 3, the range (r) to any given receiver is projected to the plane of the transmitter (d), which maps a circular locus of possible positions.

$$d = (r^2 - h^2)^{1/2} \tag{2}$$

The interception of two or more of these loci gives the position of the robot. We use at least three range data samples in order to determine the position. Analytic geometry can be used to solve simultaneous equations of three circles to find the common point:

$$\begin{aligned} (x_1 - x_r)^2 + (y_1 - y_r)^2 &= d_1^2 \\ (x_2 - x_r)^2 + (y_2 - y_r)^2 &= d_2^2 \\ (x_3 - x_r)^2 + (y_3 - y_r)^2 &= d_3^2 \end{aligned} \tag{3}$$

x_i, $y_i (i = 1\text{–}6)$ are the location of sensors in the global 2D coordinate space system which are measured exactly and used in the calculations. d_i are the radii of circles, which are computed from range data as in Eq. (2) and x_r, y_r are the computed position of the robot.

Due to random errors introduced in the range data, they will not coincide exactly on a single point, hence the least squares method (Farebrother, 1988) is used to reduce combined differences to a minimum.

3.2. *Result of Experiments*

The proposed method was tested using the mobile robot in an indoor 6 m × 4 m room. This room is divided into six 2 × 2 m^2 areas and one receiver module is placed in the center of each area. The sensors are installed at a height of 3 m. The transmitter is mounted on the robot at a height of 80 cm. The 3D positions of the receiver modules are measured with surveying devices and receivers are set up to cover their supposed effective area. The system is calibrated by measuring the range from a number of known locations and each receiver is calibrated separately. The resolution of the receiver circuit in measuring the range is 1.5 cm and it can detect signals coming from a distance of up to 4 m.

Table 1. Results of static positioning with 10 readings for each point.

Actual position (cm)	Mean measured position	Max error	Standard deviation
$X = 250$	$X = 250.8$	$\Delta x = 1.5$	$\text{Std}(x) = 0.51$
$Y = 280$	$Y = 278.8$	$\Delta y = 3$	$\text{Std}(y) = 1.08$
$X = 430$	$X = 429.4$	$\Delta x = 2$	$\text{Std}(x) = 0.78$
$Y = 210$	$Y = 209.3$	$\Delta y = 3$	$\text{Std}(y) = 1.18$
$X = 150$	$X = 148.5$	$\Delta x = 2.5$	$\text{Std}(x) = 0.78$
$Y = 150$	$Y = 149.6$	$\Delta y = 3.5$	$\text{Std}(y) = 1.18$

3.2.1. Static Position Measurement. The purpose of this experiment is to evaluate the ability of the system in estimating the initial position of the robot. In this experiment, the robot is located at some known points in the room and its position is measured by this system. Measurement is repeated 10 times for each point. The measurements for three sample points are shown in Table 1. According to the experiment results, the error in positioning is less than 5 cm.

3.2.2. Dynamic Position Measurement. The objective of this experiment is to demonstrate the feasibility of the proposed positioning system in real time navigation and path tracking.

This experiment was conducted involving straight motion of the robot with a speed of 0.3 m/sec. The robot is given a target position and is asked to follow a straight path toward the target. The path is then measured while the robot is following the determined path. Figure 4 shows the dynamic position measurements of the robot while it is following the straight line at $y = 285$ cm. In this figure, the solid line indicates the reference path and the '+' marks indicate the position measured during motion.

3.2.3. Estimation of Heading Angle. For measuring the heading of the robot, we find its initial position

Table 2. Experimental robot heading angle measurements.

Actual heading (deg.)	Absolute value of error in measurement of heading angle (deg.)			
	At 20 cm	At 30 cm	At 40 cm	At 50 cm
0	9.8	6.6	4.9	3.2
90	10.1	5.9	5.2	3.1
45	13.4	8.8	6.9	4.1

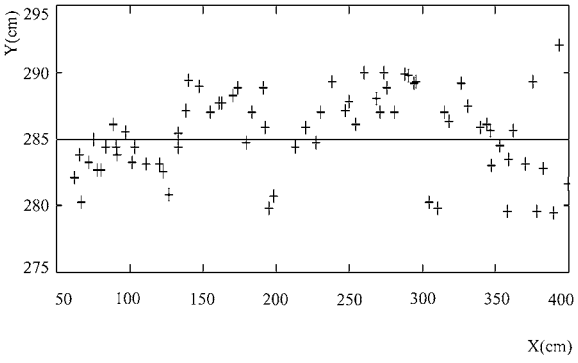

Figure 4. Experimental dynamic position measurements of the robot while it is following a straight line at $y = 285$ cm.

(x_0, y_0) using the HRPS. The robot is then asked to move to a new point (x_1, y_1). The heading angle can be calculated as:

$$\theta = \tan^{-1}[(y_1 - y_0)/(x_1 - x_0)] \tag{4}$$

Results of this experiment are reported in Table 2. These results show that the minimum distance required to measure the heading angle is about 50 cm.

4. Object Recognition and Localization

Object recognition is an important capability for autonomous robotic systems. Research on automatic visual object recognition is several decades old. Despite many encouraging results, current object detection and recognition techniques are object specific and limited to problems with carefully engineered object presentation. It is not yet possible to fully recognize all objects of the environment by a computer system.

An intermediate solution is for a human to provide the robot with information about objects of environment. Humans perform object recognition very efficiently. By providing a suitable interface between the human and robot it is possible for the robot to use the knowledge and intelligence of the human.

In this system, instead of using different recognition methods for every object we use a simple method to recognize the human's hand in the environment. The human points to the objects using his hand and guides the robot to learn the position of hand and use it as an estimation for object position.

For hand detection, we use a method similar to our previously implemented algorithm for human detection and localization in the room environment (Shiry et al., 2000). In this method, the robot uses a CCD camera to search for the human by means of skin color information, motion detection and shape analysis. Then, by active control of camera parameters, the robot makes the camera automatically focus on the hand or face and uses information from the focusing ring of a camera to measure the distance between the human and camera. By having information about the absolute position of the robot, the camera, and its pan and tilt angles, it is possible to find the absolute position of the user in the room.

4.1. Using Color for Hand and Face Detection

Color is a powerful fundamental cue and is one of the most important characteristics of the human skin. It is often used as a method to locate and track a human face or hand in images (Yang and Waibel, 1996; Yin et al., 2001) and is relatively robust with respect to changes in viewpoint, scale, and shading. Color segmentation is computationally fast because it generally includes a simple thresholding in chrominance space. However, using color as a feature for tracking a human face or hand has several problems: Color is influenced by lighting condition, camera characteristics, variation of skin color between persons, partial occlusion, and rotation of the head.

4.1.1. Skin Color Model. Although the different people have different color appearances, several studies have shown that such a difference can be reduced by intensity normalization and the skin colors of different races fall into a small cluster in the normalized RGB or HSV color spaces (Yang et al., 1998).

The most common way of representing color is through the RGB color space. However, this color model is quite sensitive to lighting conditions since the color attribute is combined with the brightness one. In order to minimize dependency on luminance, we use

normalized chromatic colors:

$$r = R/(R + G + B),$$
$$g = G/(R + G + B) \tag{5}$$

We use a single Gaussian distribution $N(\mu, \Sigma)$ to characterize the properties of skin color. In this distribution $\mu = (\mu_r, \mu_g)^T$ where:

$$\mu_r = \frac{1}{N}\sum_{i=1}^{N} r_i$$
$$\mu_g = \frac{1}{N}\sum_{i=1}^{N} g_i \tag{6}$$
$$\sum = \begin{bmatrix} \sigma_r^2 & \sigma_{rg}^2 \\ \sigma_{gr}^2 & \sigma_g^2 \end{bmatrix}$$

We trained the statistical classifier from a sampling of peoples of a variety of races and skin colors. The parameters of the distribution are obtained by maximum likelihood estimation. A pixel is identified to have skin color if the corresponding probability is greater than a threshold.

4.2. *Gesture Recognition*

Gestures provide an intuitive interface for interaction with robots. Use of gesture in robot control includes pointing to a place to indicate its importance, pointing in a direction you want the robot to go, or moving or waving an arm to convey some signal or ask a question or point to an object of interest (Torrance, 1994). Gesture can be generated by hand, head or eye movement. In this work, we limit ourselves to gestures produced by hand. As the robot is not always looking at the human, we are not implementing dynamic gesture recognition and limit our recognition to static gesture (posture) recognition.

We use gesture in two kinds of task. One is to resolve ambiguity of some speech commands such as "*Go that way*" and the other is within the map generation task which we use for object selection and localization. For example, in statements such as "*This is a TV*" we learn about TV and its position through hand gestures.

For gesture recognition we assume that user wears long-sleeved clothes and that the robot is static while observing the gesture.

The robot uses the current view of the camera to search for the pointing posture. In the case of an unsuccessful search, the robot produces an error message and starts a dialog to ask the human to give more verbal information or displace himself so that robot can find his face and hands.

Generally posture analysis relies on analysis of hand shape. As we need a simple and fast algorithm which we can use on our robot in real time, we avoid high level models of the hand which include fingers and joints. We detect only left and right direction and the hand's flat posture. Our system simply searches for a body gesture pointing to an object or place (Fig. 5).

Prior to gesture recognition, it is essential to detect the human in the room. In our previous work (Shiry et al., 2000) we proposed a multi-cue approach consisting of three feature modules sensitive to skin color, motion information, and the circular model of the head for human head detection. We find the human's face by employing a similar method. The robot needs at least 150 cm distance from the person for successful hand and face detection.

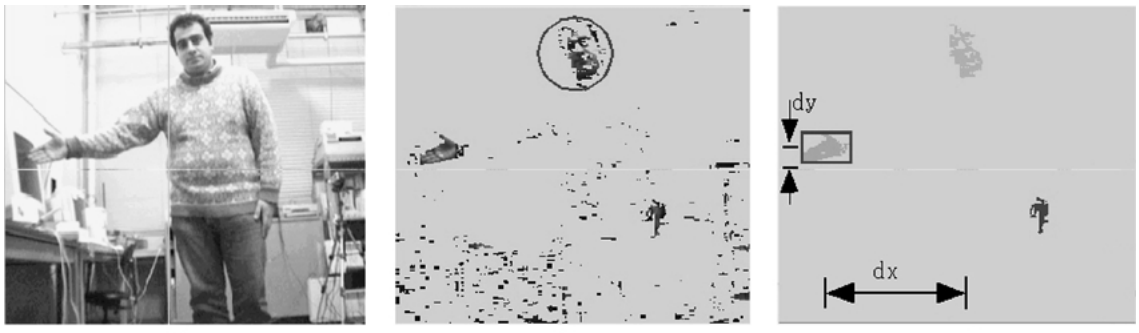

Figure 5. (a) Using gestures for introducing an object. (b) Head detection from skin and motion information. (c) Result of hand detection.

Segmentation of the human hand from the background scenery is done by a statistical skin color classifier (Fig. 5(b)). We find connected components by applying a region growing algorithm and remove unwanted regions based on size criteria (Fig. 5(c)).

We use position of the center of the head and hands to classify the gesture based on the topology of segmented colored areas. The face is a color object on the top of the image and the left or right hand is whichever is leftmost or rightmost with respect to a vertical line passing through the face. There may be some verbal information, which provides further information for hand selection. The center of mass of the selected hand is used as the 2D position of the hand in the image. We use this position to rotate the camera to put the selected hand in the center of the image and then by proper zooming and focusing, we compute the distance of hand to the camera.

4.3. Tracking and Camera Control

The main goal of camera tracking process is to keep the gaze on an object of interest previously located and fixed, pursuing it when the object is moving in the field of view (Crowley and Christensen, 1995). In our system, tracking also keeps the object of interest centered in the image by adequately moving the camera.

In order to use the autofocus ability of the camera, the camera must have an appropriate zoom setting to make the target large enough in the image. During depth measurement the camera control policy is to change the zoom so that the width of the face occupy approximately more than 60% of the width of the camera image.

To obtain a close-up view of a particular subject, three parameters need to be determined: the pan angle, the tilt angle, and the zoom factor. Also, for achieving high speed tracking, it is important to use both position and velocity control.

In this system we use fuzzy logic rules and reasoning to control the speed and direction of the camera. Fuzzy control has the advantage that we can do tracking without need for camera calibration or modeling.

Input values dx, dy (Fig. 5(c)) determine the difference between the current position of the hand in the image and the image center point.

$$\begin{aligned} dx &= x_m - x_0 \\ dy &= y_m - y_0 \end{aligned} \tag{7}$$

All fuzzy inputs are divided into five base membership functions: large negative (LN), medium negative (MN), zero (Z), medium positive (MP), large positive (LP). A trapezoid membership function is applied to all inputs.

The controller has four outputs: pan angle, tilt angle, pan speed and tilt speed. A triangular membership function is applied to the outputs. The inference is of Mamdani inference engine type and the center of gravity defuzzification method is implemented.

The fuzzy controller consists of a rule base for speed and direction control of the camera. It contains rules for change of tilt angle, change of pan angle and values of speed for pan and tilt change. The rules are formulated in classical logic form.

Zoom setting is performed when the hand is located in the center of the image. The Sony EVI-D30 has 1000 settings for the zoom lens position. But, because of calibration problems, we use only a few of the settings. Our first setting choice is the "Tele" position, which gives measurability over a wider range.

In order to detect features such as hands or faces in the image, the camera must have an appropriate zoom setting such as Fig. 5. However, for measuring distance, the system cannot use wide zoom and has to change the zoom to provide a closer view as in Fig. 6. Therefore, the zooming is a dynamic function in which the system selects close view when measuring the distance and select wide view when tracking.

While tracking, the camera is moved in a saccadic way in which no vision processing occurs during the camera movement.

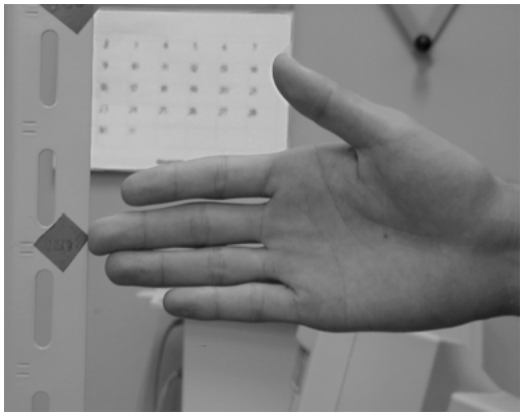

Figure 6. System zooms and automatically focuses on the detected hand.

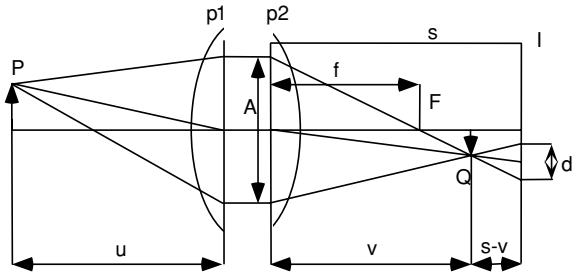

Figure 7. Image formation in a lens.

4.4. Depth Information Using an Autofocus Camera

This system estimates the 3D position of human's hand by using the depth from focus method. Focus interpretation is a valuable alternative to stereo vision because it does not require solving correspondence for depth recovery (Xiong and Shafer, 1993).

Figure 7 shows basic image formation geometry in a lens. For a point P in the scene, the radiated light which passes through aperture A are refracted by the lens to converge at point Q. The distance of this point to the lens is determined by the lens law:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \tag{8}$$

If the image detector is located at this point, then each point on the object plane is projected to a single point on the image plane and we will have a sharp or focused image. However, if the image detector is displaced from point Q then the image of P becomes a circular image called the blur circle. The structure of this circle can be represented by a point spread function (Subbarao et al., 1993).

Most imaging systems such as standard cameras can only be in focus for one distance at a time. Surface regions at this focal distance will produce sharp images while surface regions at other distances will produce blurred images. Surfaces at different distances can be brought into focus one at a time by adjusting the power (focal length) of the optical system. In the human visual system this is accomplished by changing the shape of the lens. In camera systems, the position of one or more camera lenses is changed.

Several researchers have proposed depth cues based on focusing information (Xiong and Shafer, 1993; Subbarao et al., 1993; Krotkov, 1987; Pentland, 1987). A common technique involves focusing the object on the image detector in the camera. The distance is then determined from the camera setting. This technique is called depth from focusing (Krotkov, 1987). The other common technique is depth from defocusing, which calculates depth from the degree of image blur (Pentland, 1987).

In general, autofocusing techniques for video cameras maximize the high frequency components of an image by adjusting the focusing lens. Focused images have more high frequency components than defocused images of a scene, thus, defocused images can be described as the result of convolving the focused image with the blurring function that plays the role of a low-pass filter.

In this system we use information from the focusing ring of an autofocus camera to measure depth information. This ring provides depth information about the object in the center of the camera's field of view. By actively controlling of the camera parameters, we make the camera autofocus on the object. In this way, we can compute the distance between the object and camera using the data from the focus ring encoder. The relationship between the position of the autofocus lens and the distance of the object to the camera is formulated as a function:

$$D = F(\text{focus lens position}) \tag{9}$$

Function F is driven experimentally. Figure 8 shows such a derived function.

In this figure the vertical axis is the position of the autofocus lens and the horizontal axis is the distance of the focused object to the camera.

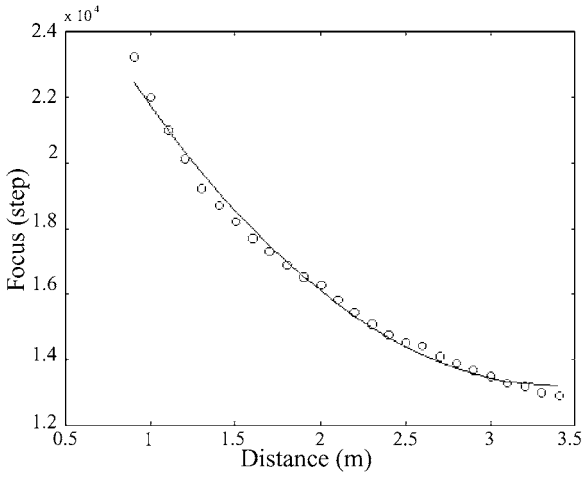

Figure 8. Calibration curve for an autofocus lens at the "Tele" position.

Position of the object in the camera-centered coordinate system can be calculated as follows:

$$(x, y, z) = (D \sin\beta \cos\alpha,\ D \cos\alpha \cos\beta,\ D \sin\alpha) \tag{10}$$

Where D is the distance from the object to the camera, α is the tilt angle and β is the pan angle of the camera. For the Sony EVI-D30, the camera tilt and pan the angle are computed from the encoder data:

$$\begin{aligned} \alpha &= 0.001977 \text{ rad/step} \\ \beta &= 0.001449 \text{ rad/step} \end{aligned} \tag{11}$$

By using information from HRPS we find the absolute position of the robot in the environment. We also have information about the position of the camera on the robot. Thus, we can compute the absolute position of the object in the room. This position is computed by transferring the camera-centered coordinate to the coordinate in the global coordinate system.

This method is quite simple and needs only the calibration of one camera parameter to obtain the relationship between the distance of the focused object to the camera and the position of the autofocus lens.

4.5. Calibration

Modern cameras have varying focus and zoom, both of which change the calibration parameters. This makes calibration of the zoom lens a complex problem.

In our system, we are considering the variation of only one camera parameter: the autofocus lens position. The position of this lens can be determined via the autofocus ring encoder. In the calibration procedure, we use a calibration pattern to search for the relationship between the distance of the focused object and the position of the autofocus lens.

The calibration pattern is a wall-mounted square array of dark and white rectangles serving as an absolute position reference. The test pattern is placed at a known distance from the camera perpendicular to the optical axis of the camera. The calibration pattern should be sharply displayed for an accurate measurement. To calibrate the camera, we aim it at the calibration pattern and then let it automatically focus on the pattern. The camera has a focus control which can displace the autofocus ring to produce a best-focused picture. For each position of the test pattern, we repeat the measurement several times and record data for the focus ring encoder. We then apply least square fitting to estimate the parameters of the model. Figure 8 shows the derived 2 dimensional calibration curve for the camera autofocus ring.

We do calibration only for some predetermined settings of the zoom lens position. These settings are mainly distributed near the "Tele" position of the lens.

Table 3. Measured distance between a hand and camera at some test points.

Depth (cm)	Measured (cm)	Error (%)
90	91	1.1
120	117	2.5
180	185	2.7
240	246	2.5
270	264	2.2
300	295	1.6
340	344	1.7

4.6. Depth Measurement Results

The proposed method provides real time hand detection and tracking. However, as the focusing ring moves slowly it takes more time for fine zooming and focusing on the detected hand. In the case of a person with small motions, the robot can locate the hand position in the room with less than 10 cm error. The measurable distance range between the robot and human is from 90 cm up to 340 cm. Table 3 lists the errors reported by the system when measuring the distance between the camera and a nearly-static human in the room.

4.7. Position Calculations

The world coordinates of the pointed object is calculated on the basis of the position of the hand and the learned size of the objects.

Position of the hand produces data only for one point in the 3D space. To relate this information to the position of objects with different size, we use an alignment algorithm which uses position of hand, learned size of object, location of nearby objects or walls, and type of objects. Most of this information is learned through communication with the human.

5. Human-Robot Interaction

We have limited our interaction system to the context of map generation in which the robot learns the map of the environment through communication with the human. As a requirement for successful interaction in

our system, it is very important to coordinate between the human and robot. The user must know the communication capability of the robot and issue the instruction appropriately. On the other hand, the robot must be able to map the user-provided information into its own action space and knowledge representation.

In this system, learning takes place in a supervised manner. For learning from interaction, the robot receives instructions from the user about the object being introduced. To understand these instructions, a common language between the robot and the user is necessary. Predefined interaction patterns, are used for exchange of information between the user and robot. We consider four main interaction patterns: introducing new object or places, guiding the robot in free space, making requests from the root, and error feedback.

The robot learns names for places and objects, their size, and their spatial relation to other objects in the environment, and makes a knowledge base from them. The learned information can be referred to by a subsequent query or other subsequent commands.

5.1. Natural Language Interface

The function of speech in our system is to select commands and gesture. The user's speech selects a command from a predefined command set. The robot's behavior controller, which is a finite state machine, manages the state of the system and execution of commands. It is responsible for planning an appropriate behaviour sequence to reach a given goal. Through speech, the user may cause robot movement, introduce an object, or make a query about learned objects.

In this system, natural language processing is currently restricted to sentences with simple structure. We use a restricted task-dependant vocabulary and grammar for speech recognizing and parsing, to increase accuracy in recognition and decrease the speech recognition latency.

To extract clean voice signal in the room environment, the human uses a wireless microphone with noise cancellation capability. The signal of this microphone is transmitted to the host computer for speech recognition. This provides user mobility in the room environment.

In this system the verbal input is converted to a text string by a commercially available speech recognition system. The textual string is then parsed by our natural language processing system and is translated into robot commands.

For producing linguistic feedback, we use a dialog manager module which helps the robot in learning, in reporting errors, or answering queries in a linguistic way.

5.2. Dialogue Category

We have a variety of commands and statements, which help in providing information about the environment or asking requests from the robot. Usually there is enough redundancy in each linguistic input to allow the user to speak to the robot with as much flexibility as possible.

Each command has a speech component, which defines the action, and it may have a gesture modifier, which provides complementary information for that command. Deciding command category is very important because every command starts a behavior in the robot and a larger command set requires a larger behavior set in the robot too.

The main set of languages supported by our system includes: *motional* commands, *teaching* statements, *object modifiers*, *fuzzy* commands and *knowledgebase* query and update.

5.2.1. Motional Commands. The user uses *motional* commands such as: *Move*, *Go direct*, *Stop*, *Turn left*, etc., to guide the robot through a pathway or to a position where a new object can be introduced. Each of these commands causes the robot to plan for its navigation and start a new activity.

5.2.2. Teaching Statements. The user teaches the robot about places or the position of objects during a dialog starting with a sentence like:

"I teach you a TV"

There are four different methods for introducing an object. The robot asks the user to explicitly define the teaching method and then follows the dialogue for each method to get necessary information to complete its knowledge about the object.

5.2.2.1. Teaching Objects by Pointing to the Centre of the Object. This method is used for teaching small and medium sized objects such as TV sets. In this method, the user points to the object by his hand (Fig. 9) and introduces the object through proper dialog. This method has the advantage that the robot does not have to move to the object while it is being introduced. However, this approach involves more problems with recognition and understanding. This command cannot

Figure 9. User introducing a new object to the robot.

be understood unless the robot sees the human in its field of view and the pointing hand is beside or over the object. The user must guide the robot to the proper position before giving such commands; otherwise the robot will report the error after its failure to find the human. The robot tracks the pointing hand and uses hand position as an estimation of object position in the room.

5.2.2.2. Teaching Objects by Pointing to two Corners of an Object. This method is used for large objects such as tables and doors. This method differs with the previous method in the number of points necessary for the introduction of objects. Here the user points to two corners of a table, one by one. The robot follows a predefined dialogue pattern to ask necessary information about the object.

5.2.2.3. Teaching Objects by Referring to the Robot Position. In this method the user introduces the position of objects with reference to the robot's current position. For example:

"TV is in front of you"

In this statement, the location of the new object is estimated according to the position of the robot and the learned size of object. The robot uses HRPS to find its position and then uses an estimation algorithm to estimate an object position from other information learned via interaction.

The size of unknown objects are described by object modifiers such as:

"TV has medium size"

The robot has a predefined understanding of size descriptors.

5.2.2.4. Teaching Objects by Referring to Already Learned Objects. In this method, the user teaches new objects with reference to already learned objects. For example:

"TV is on the Table"

Here the table should be an object in the robot's knowledge base, else the robot will report an error. The size and other attributes of the new object are learned by proper dialog.

5.2.3. Request. The user may then make requests of the robot. For example:

"Go to the TV"

The robot uses its learned information to make a plan for this request. This kind of reference is very useful and helps in directing the robot to the destination in more natural ways.

5.2.4. Fuzzy Commands. The user uses a set of *fuzzy* commands to move the robot to a specified position in the room. It happens that when directing the robot toward a specific location, fine positioning becomes very difficult due to error in the robot's direction perception. In this case it is necessary to give commands such as "*a little more to right*". For these fuzzy commands the robot has to memorize the previous command and adjust its behavior using a membership function as a function of distance or rotation done in previous motional or fuzzy commands.

Although the robot can turn or move an arbitrary number of degrees or centimeters in any direction, there are occasions which utterances like "*move a little more*" are more natural and easier for humans to issue. Fuzzy commands solve the problems caused by difficulties in recognizing numbers in speech recognition systems. Such context dependant commands need the previous activity to be memorized and replanned if necessary. The validity scope of these commands is limited to previous motional commands.

5.2.5. Knowledge Update. The ability to update learned knowledge plays an important role in our interaction system because with the current implementation, the human provides the main information for the robot and if there is any mistakes made by the human or if there is any change in the room arrangement which cannot be detected autonomously, we need a means of communication to update the knowledge base.

5.2.6. Error Reporting. The robot uses an error reporting system to produce an error message when it

cannot realize a linguistic command or visual gesture, or when there is any missing object in its knowledge base. This bi-directional communication mainly helps the human to do its actions in a way appropriate for robot understanding.

5.3. *Sample Dialog*

The following dialog between a human and robot illustrates how the new object is introduced to the robot: (*H* is for Human and *R* is for Robot)

H: I teach you a TV.
R: Is it a TV?
H: Yes.
R: Where is the TV?
H: Follow my hand.
R: How many points will you use to introduce it?
H: One point.
R: Put your hand on the TV.
H: I did.
R: I found it. What is the size of TV?
H: Medium.
R: Which side did you point out?
H: Left.
R: Who is the owner of this TV?
H: Saito.
R: I learned TV.

Although in our system, the human always starts the interaction by issuing commands and queries, the robot takes control of the dialog to learn necessary information out of interaction.

5.4. *Dialogue Manager*

The most commonly used and simplest dialogue management technique is state-based (Spiliotopoulos et al., 2001). In this technique, the possible dialogues are represented by a series of states. At each state, the system asks the user for specific information. The system may generate a response to the user, or it may call a procedure to use implemented internal behavior. The structure of the dialog is predefined, and at each state the user is expected to provide particular inputs. This makes the user's utterances easier to predict, leading to faster development and more robust systems at the expense of limited flexibility in the structure of the dialogues (Spiliotopoulos et al., 2001). In our system the main dialogs are used in teaching procedure and error reporting or correction.

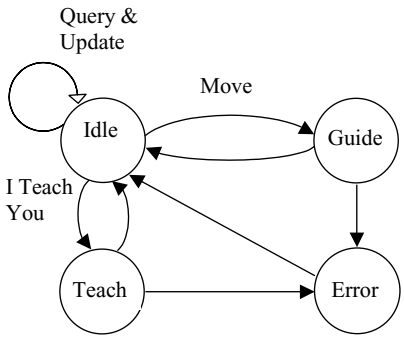

Figure 10. Basic states of interaction system.

5.5. *State Controller*

The robot's controller consists of a simple state machine with basic states shown in Fig. 10. In the "Idle" state, the robot waits for the human to start the interaction. The human's command or statements change the state to a new one. The state changes to the "Guide" state when the human issues commands to make robot movement. In the "Teach" state, the robot starts a detailed dialog to learn necessary information from the human. The "Error" state handles necessary dialog to inform the human about the cause of the error and the change of the interaction state.

6. Map Generation and Navigation

In the map generation process, the user uses a combination of commands and statements to guide the robot to the location of interesting objects in the room, or to a position from which the robot can detect the user's pointing hand.

We use a geometrical map for representing the room environment. HRPS provides exact data for robot positioning and map generation and makes the global coordinate system available. We also use an attributed topological map, which stores characteristics of learned objects and pathways between them. For each object, we store attributes such as: size, front side, owner, room name, list of nearby objects in the left, right, up, down, front and back sides of object, and object type. Initially the map is empty and only the size of the room and position of the walls are prior-known items. Free locations of the map are considered as pathways unless an object is introduced or the sensory system detects an obstacle in the environment.

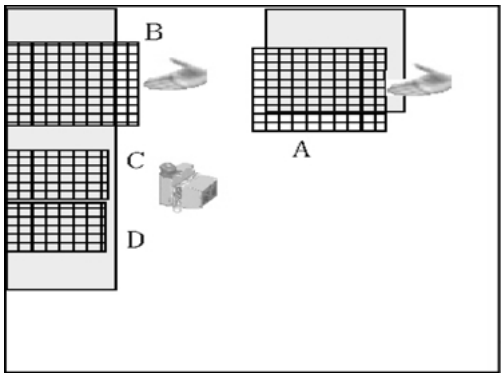

Figure 11. A sample map of the room learned through human-robot interaction. The real world map is shown in gray and the map generated by the robot is shown by dashed lines.

Map generation is affected partly by accuracy of the robot's position and the hand position measurement, and mainly by the robot's representation of objects and the object alignment algorithm. Due to the fact that the robot's knowledge about the size of the objects is inaccurate and that errors exist in hand positioning, there would be a mismatch between the world imagined by the robot and the real world (Fig. 11). Thus, the robot will need to be able to revise its internal representation when facing new facts about the environment. Doing this revision is possible through obstacle sensing and also knowledge-base updating via interaction with the human.

We have supposed that all objects can be represented by a square shape in a 2D map and are parallel to the walls. We apply an alignment algorithm which uses a simple object model, world model, hand direction, front side of object, object type, and hand position information to estimate the position of objects in the room. This algorithm produces objects parallel to the walls and side centered to the detected hand's position or in front of the current location of the robot. Object sizes are extended to fill small gaps between different objects or objects and walls.

Figure 11 shows a sample map generated by this method. The objects A and B are positioned by measuring the position of the hand. Their sizes are specified by the user to be objects with medium size. The object C is positioned by indirect statements. In this case the user introduced a small sized object in front of the robot. Object D is introduced to be on the right side of known object C and to have a small size. The user can introduce a stack of objects in the same position, if necessary.

The produced map is obviously a rough representation of the real world with references to the learned objects. Successful navigation with this map requires careful utilization of sensory data.

6.1. Path Planning

Path planning is a fundamental issue in robotics. The purpose of a path planner is to compute a path, i.e., a continuous sequence of configurations that leads the robot to its goal. In our system, the existence of an absolute world model allows for automatic path planning and execution, and for subsequent route revisions in the event that a new obstacle is encountered.

The whole path of the mobile robot is expressed by an assembly of straight line segments and turning angles. The configuration of the robot position is specified by a 3D vector, whose elements are the 2D position of the front wheel and the orientation of the robot. The size of the robot is considered by setting a margin around each obstacle. The size of the margin is set to half of the robot width.

When the parser translates a motional command, it provides the destination to the robot. Once the goal has been specified, the robot performs both physical action (actual motion) and cognitive actions (reasoning) to reach the goal. Planning is done in a classical way creating detailed sequences of actions to be broken down to the lowest level of action and executed until some problem arises. The robot searches for the destination in the map and plans a route from its current location to the nearest side of the destination object. The route is planned by the direct path algorithm, in terms of navigation behaviors. Then, the planned behavior sequence is executed and if nothing unplanned happens (no dynamic obstacle is avoided) the robot reaches a position in front of its destination.

In the interaction system, path planning is initiated by a command such as "Go to the TV". Here the robot searches for the referenced object in its knowledge base and then checks for its topological relationship.

Included within the topological map is the vertical relationship between objects. Such a relationship is shown in Fig. 12.

If the selected object is not directly located on the ground, then the robot searches for objects under it with connection to the room floor. For the example in Fig. 12, the robot chooses the position of the table and also the front side of the TV to determine the destination point in the path planning process (Fig. 13).

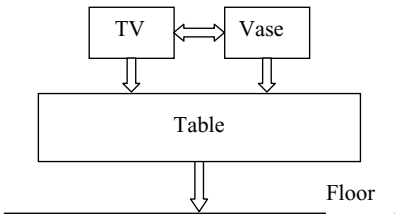

Figure 12. Topological relationship between objects.

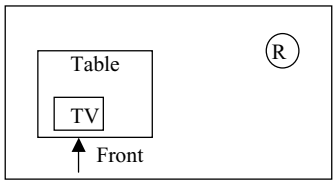

Figure 13. Front sides of the objects are used to determine the destination point.

7. Conclusion

In this work, we have demonstrated multi-modal interaction between a human and a home robot in the context of map generation. Our work is a multi-disciplinary research, composing of research in several fields such as robotics, machine vision, natural language understanding, and machine learning. We divided the current research into several sections. Techniques of each one were presented.

The generated map provides very important information about the room environment, information that no current sensor is able to provide. In spite of low accuracy in the position of objects in this map, this information is essential for common understanding about the environment between the robot and human. Use of interaction in the map generation system is a basic contribution to sharing human knowledge and providing intelligence for robots with as much ease as possible.

We introduced a method for object localization, which is an on-demand localization method, useful for locating any prior-detected object.

Our positioning system, HRPS is an easy and practical system, which makes the global world coordinate system available. This has a big influence in all parts of the system.

In the natural language processing domain, problems exist with speech recognition technology. We have adapted a limited grammar and dictionary based on our specific task. Furthermore, we believe that speech is the most essential means of interaction between the human and robot and needs to be investigated further.

References

Agah, A. 2000. Human interactions with intelligent systems: Research taxonomy. *Computers and Electrical Eng*., 27(1):71–107.

Aggarwal, J., Ghosh, J., Nair, D., and Taha, I. 1996. A comparative study of three paradigms for object recognition: Bayesian neural networks and expert systems. In *Advances in Image Understanding*, IEEE Computer Society Press, pp. 241–262.

Alford, W.A., Rogers, T., Wilkes, D.M., and Kawamura, K. 1999. Multi-agent system for a human-friendly robot. In *Proc. of the 1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC '99)*, Tokyo, Japan, pp. 1064–1069.

Arai, T. and Nakano, E. 1983. Development of measuring equipment for location and direction (MELODI) using ultrasonic waves. *Trans. ASME, Journal of Dynamic Systems, Measurement and Control*, 105:152–156.

Beom, H.R. and Cho, H.S. 1995. Mobile robot localization using a single rotating sonar and two passive cylindrical beacons. *Robotica*, 13:243–252.

Bischoff, R. and Jain, T. 1999. Natural communication and interaction with humanoid robots. In *2nd International Symposium on Humanoid Robots*, Tokyo, Japan, pp. 121–128.

Borenstein, J. and Feng, L. 1994. UMBmark—A method for measuring, comparing, and correcting dead-reckoning errors in mobile robots. The University of Michigan, Technical Report UM-MEAM-94-22.

Crowley, J.L. and Christensen, H.I. 1995. *Vision as Process*. Springer-Verlag, Berlin.

Farebrother, R.W. 1998. *Linear Least Squares Computations*. Marcel Dekker, New York.

Horn and Schmidt, G. 1995. Continuous localization of a mobile robot based on 3d-laser-range-data, predicted sensor images, and dead-reckoning. *Robot. Auton. Syst.*, 14:99–118.

Kim, J.H. and Cho, H.S. 1992. Real time determination of a mobile robot's position by linear scanning of a landmark. *Robotica*, 10:309–319.

Kleeman, L. 1992. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning. In *IEEE International Conference on Robotics and Automation*, Nice, France, pp. 2582–2587.

Koh, K.C., Kim, J.S., and Cho, H.S. 1994. A position estimation system for mobile robots using a monocular image of a 3-D landmark. *Robotica*, 12:431–441.

Krotkov, E. 1987. Focusing. *Int. Journal of Computer Vision*, 1(3): 223–238.

Matsui, T., Asoh, H., Fry, J., Motomura, Y., Asano, F., Kurita, T., Hara, I., and Otsu, N. 1999. Integrated natural spoken dialogue system of Jijo-2 mobile robot for office services. In *Proceedings of AAAI-99*, Orlando, FL.

Oestreicher, L., Hüttenrauch, H., and Severinsson-Eklund, K. 1999. Where are you going little robot?, Prospects of human-robot interaction. Position paper for the *CHI '99 Basic Research Symposium*, Pittsburgh, USA.

Pentland, A.P. 1987. A new sense for depth of field. *IEEE Trans. Patt. and Machine Intell. PAMI*, 9:522–531.

Schultz, A.C. and Adams, W. 1998. Continuous localization using evidence grids. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 2833–2839.

Shiry, S., Nakata, Y., Takamori, T., and Hattori, M. 2000. Human detection and localization at indoor environment by home robot. In *IEEE SMC Conference*, Nashville, USA.

Shiry, S., Tani, T., Takamori, T., and Hattori, M. 1999. A new home robot positioning system (HRPS) using IR switched multi ultrasonic sensors. In *IEEE SMC Conference*, Tokyo, Japan.

Spiliotopoulos, D., Androutsopoulos, I., and Spyropoulos, C.D. 2001. Human-robot interaction based on spoken natural language dialogue. In *European Workshop on Service and Humanoid Robots* (Servicerob 2001), Santorini, Greece.

Subbarao, M.T., Choi, S., and Nikzad, A. 1993. Focusing techniques. *Journal of Optical Engineering*, 32(11):2824–2836.

Torrance, M. 1994. Natural communication with robots. Masters Thesis, Massachusetts Inst. of Tech.

Xiong, Y. and Shafer, S. 1993. Depth from focusing and defocusing. Tech. Report CMU-RI-TR-93-07, Robotics Institute, Carnegie Mellon University.

Yang, J., Lu, W., and Waibel, A. 1998. Skin-color modeling and sdaptation. In *Proceedings of ACCV'98*, Vol. II, pp. 687–694.

Yang, J. and Waibel, A. 1996. A real time face tracker. In *Proc. of WACV*, Sarasota, Florida, USA.

Yin, X.M., Guo, D., and Xie, M. 2001. Hand image segmentation using color and RCE neural network. *Robotics and Autonomous System*, 34(4):235–250.

Saeed Shiry Ghidary received his B.E. degree in Electronics and M.Sc. in Computer Architecture from Amir Kabir University, Tehran in 1991 and 1994, respectively. He was a faculty member of Shahed University, Tehran from 1994 to 1997. He received his Ph.D. in Intelligent Artificial Systems from Kobe University, Japan in 2002.

He is currently a postdoctoral fellow at GMD-JRL research laboratories. His research interests include human-computer interaction, intelligent robotic systems, vision, and artificial intelligence.

Yasushi Nakata received his B.E. from Engineering Department, Kobe University in 2000. He graduated from Graduate School of Science and Technology, Kobe University in 2002. His research interests include vision and speech processing.

Hiroshi Saito received his B.E. from department of computer and systems engineering, Kobe University in 2001. Now he is a graduate student at Takamori-Tadokoro Laboratory, Kobe University. His research interests include recognition of real world with various sensors and multimodal interaction between a robot and a human.

Motofumi Hattori is a research associate of department of computer and systems engineering, faculty of engineering, Kobe University since 1993. He received his first master degree from Graduate School of Science, and the second master degree from Graduate School of Engineering, Kobe University in 1991 and 1993 respectively. He received the degree of Dr. Eng. from Kobe university in 2000. The title of his doctor thesis is "Analyses of emotions of Bunraku puppet's actions manipulated by an expert of puppet manipulation". His research interests include description, analysis, and design of humanoid's motions and mobile robots which cooperate with human in the home.

Toshi Takamori received the M.S. degree in 1967 from Kobe University, and the D.U. degree in 1971 from Universite de Toulouse, the D.E. degree in 1967 from Osaka University. He was an associate professor from 1975 to 1983 and a professor from 1983 to 1992 in Dept. of Instrumentation Engineering, Kobe University. He has been a professor in the Department of Computer and Systems Engineering at Kobe University since 1992. His research interest is in new actuators and robotics. He received Hydraulic and Pneumatic Technology Foundation Award in 1993. He is a member of IEEE, JSME, RSJ, SICE, and JSPE.