Preprints of the 4th Int. Conf. on Field and Service Robotics,
pp. 293-298, July 2003.

Development of a Personal Service Robot with User-Friendly Interfaces

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

Abstract

This paper describes a personal service robot developed for assisting a user in his/her daily life. One of the important aspects of such robots is the user-friendliness in communication; especially, the easiness of user's assistance to a robot is important in making the robot perform various kinds of tasks. Our robot has the following three features: (1) interactive object recognition, (2) robust speech recognition, and (3) easy teaching of mobile manipulation. The robot is applied to the task of fetching a can from a distant refrigerator.

1 Introduction

Personal service robot is one of the promising areas to which robotic technologies can be applied. As we are facing the "aging society", the need for robots which can help human in various everyday situations is increasing. Possible tasks of such robots are: bringing a user-specified object to the user in the bed, cleaning a room, mobile aid, social interaction.

Recently several projects on personal service robots are going on. HERMES [2, 3] is a humanoid robot that can perform service tasks such as delivery using vision- and conversation-based interfaces. MORPHA project [1] aims to develop two types of service robot: robot assistant for household and elderly care and manufacturing assistant, by integrating various robotics technologies such as human-machine communications, teaching methodologies, motion planning, and image analysis. CMU's Nursebot project [10] has been developing a personal service robot for assisting elderly people in their daily activities based on communication skills; a probabilistic algorithm is used for generating a timely and use-friendly robot behaviors [9].

One of the important aspects of such robots is the user-friendliness. Since personal service robots are usually used by a novice, they are required to provide easy interaction methods to users. Since personal service robots are expected work in various environments and, therefore, it is difficult to give a robot a complete set of required skills and knowledge in advance; so teaching the robot *on the job* is indispensable. In other words, *user's assistance to a robot* is necessary and should be done easily.

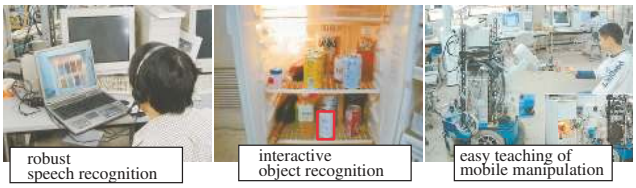


Fig. 1: Features of our personal service robot.

We are developing a personal service robot which has the following three features (see Fig. 1):

1. Interactive object recognition.
2. Robust speech recognition.
3. Easy teaching of mobile manipulation.

The following sections will describe these features and experimental results.

The current target task of our robot is fetching a can or a bottle from a distant refrigerator. The task is roughly divided into the following: (1) movement to/from the refrigerator, (2) manipulation of the refrigerator and a can, (3) recognition of a can in the refrigerator. In the third task, a verbal interaction between the user and the robot is essential to the robustness of the recognition process.

2 Interactive Object Recognition

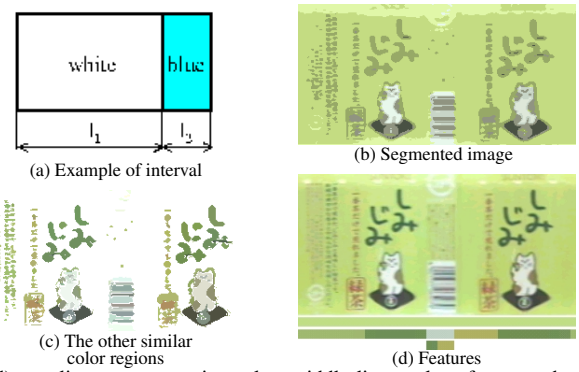
This section explains our interactive object recognition method which actively uses the dialog with a user [6].

2.1 Registration of Object Models

The robot registers models of objects to be recognized in advance. A model consists of the size, representative colors (primary features), and secondary features (the color, the position, and the area of uniform regions other than representative colors). Secondary features are used only when there are multiple objects with the same representative colors. For model registration, the robot makes a "developed image" by mosaicing images captured from eight directions while the robot rotates an object. Fig. 2 shows acquisition of developed images for two types of objects.

(a) Original image (b) Piece image (c) Developed image
top : can bottom : square type PET bottle

Fig. 2: Procedure for constructing a developed image



(a) Example of interval (b) Segmented image

(c) The other similar color regions (d) Features

In (d), top line: representative color, middle line: color of secondary feature 1, bottom line: color of secondary feature 2

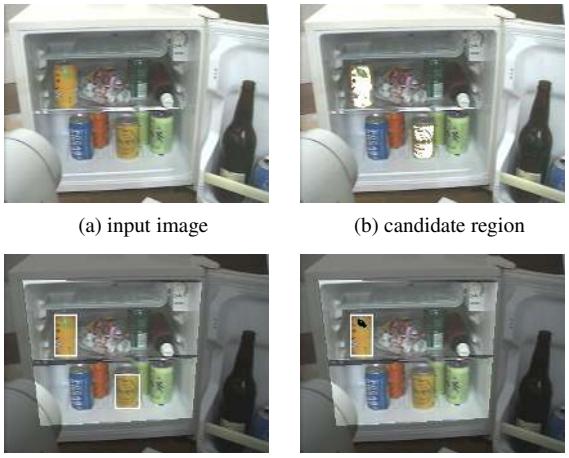
Fig. 3: Extraction of features

Since primary and secondary features depend on the viewing direction, we determine intervals of directions where similar features are observed. In the case of Fig. 3(a), for example, two intervals, I_1 (white) and I_2 (blue), are determined. If two objects are not distinguishable for an interval, it is further divided into subintervals using secondary features. Candidates for secondary features are extracted as follows. The robot first segments a developed image into uniform regions (see Fig. 3(b)) to extract primary features used for first-level intervals. The robot then extracts uniform color regions other than representative colors (see Fig. 3(c)) and records the size, the position, and the color of such regions as candidates for secondary features (see Fig. 3(d)). Secondary features of an object are incrementally registered to its model every time another object having a similar feature and being undistinguishable in several viewing directions is added to the database.

2.2 Object Recognition

The robot first extracts candidate regions for objects based on the object color which is specified by a user or is determined from a user-specified object name. Then it determines the type of each candidate from its shape; for example, a can has a rectangular shape in an image.

For each candidate, the robot checks if its size is comparable with that of the corresponding object model. If no secondary features are registered in the model, the recognition finishes with success. Otherwise, the robot tries matching using secondary features. Fig. 4 shows an example matching process. Fig. 4(c) shows two candidates are found using



(a) input image (b) candidate region

(c) can candidates. (d) recognition result. (black: secondary feature)

Fig. 4: Matching with object models.

only the primary feature (representative color). Using a secondary feature, the two candidates are distinguished.

Since the lighting condition in the recognition phase may differ from that in the learning phase, we have developed a method for adjusting colors based on the observed color of a reference object such as the door of a refrigerator [7].

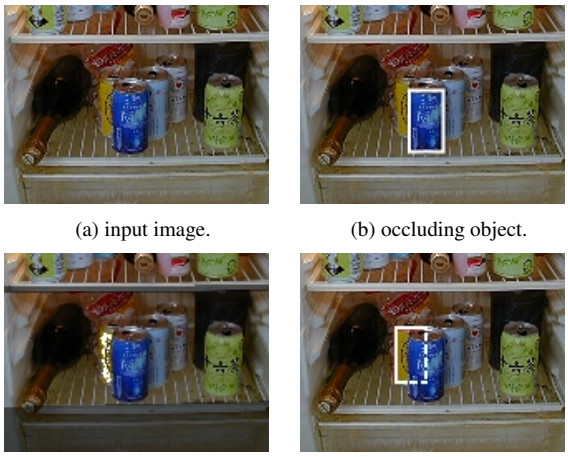
2.3 Recognition Supported by Dialog

If the robot failed to find a target object, it tries to obtain additional information by a dialog with the user. Currently, the user is supposed to be able to see the refrigerator through a remote display. We consider the following failure cases: (1) multiple object are found; (2) no objects are found but candidate regions are found; (3) no candidate regions are found due to (a) partial occlusion or (b) color change.

In this dialog, it is important for the robot to generate *good* questions which can retrieve an informative answer from the user. We here explain case (3)-(a) in detail. In this case, the robot asks the user an approximate position of the target like: "I have not found it. Where is it ?" Then the user may answer: "It is behind *A*" (*A* is the name of an occluding object). Using this advice, the robot first searches for object *A* in the refrigerator (see Fig. 5(b)). Then it searches both sides of the occluding object for regions of the representative color of the target object and extracts its vertical edge corresponding to the object boundary (see Fig. 5(c)). Finally the robot determines the position of edges on the boundary of the other side using the size of the target object (see Fig. 5(d)).

3 Robust Speech Recognition

Many existing dialog-based interface systems assume that a speech recognition (sub)system always works well. However, since the dialog with a robot is usually held in environments where various noises exist, such an assumption is difficult to be made. There is another problem that a user, who



(a) input image. (b) occluding object.

(c) region of target object. (d) recognition result.

Fig. 5: Recognition of occluded object.

is usually not an expert of robot operations, most probably uses words which are not registered in the robot's database. Therefore, the dialog system has to be able to cope with speech recognition failure and unknown words [11].

3.1 Overview of the Speech Recognition

We use IBM's ViaVoice as a speech recognition engine. Fig. 6 shows an overview of our speech recognition system. We first apply a context-free grammar (CFG)-based recognition engine to the voice input. If it succeeds, the recognition result is sent to an image recognition module. If it fails to identify some words due to, for example, noise or unknown words, the input is then processed by a dictation-oriented engine, which generates a set of probable candidate texts. Usually in a candidate text, some words are identified (i.e., determined to be registered ones) and the others are not. So the unidentified words are analyzed to estimate their meanings, by considering the relation to the other identified words. For example, if an unidentified word has a similar pronunciation to a registered word, and if the category (e.g., the part of speech) is acceptable considering the neighboring identified words, the robot supposes that the unidentified word is the registered one, and generates a question to the user to verify the supposition. The robot uses probabilistic models of possible word sequences and updates the model through the dialog with each specific user.

3.2 Estimating the Meaning of Unidentified Words

We consider that an unidentified word arises in the following three cases: (1) a known word is erroneously recognized; (2) an unknown word is uttered which is a synonym of a known word; (3) noise is erroneously recognized as a word. In addition, we only consider the case where one or consecutive two unidentified word(s) exist in an utterance. The robot evaluates the first two cases (erroneous recognition or unknown word) and selects the estimation with the highest evaluation value. If the highest value is less than

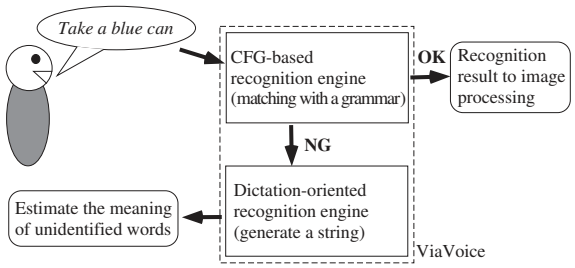


Fig. 6: Speech recognition system.

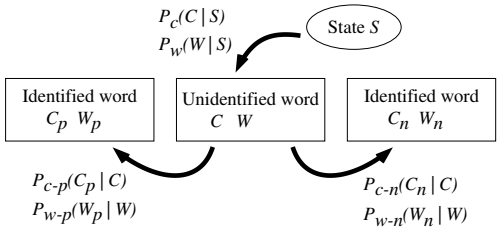


Fig. 7: Estimation of category C and word W

a certain threshold, the unidentified word is considered to come from noise.

The problem of estimating the meaning of an unidentified word is formulated as finding the registered word W with the maximum probability, given state S, context γ, and a text string R generated by the dictation-oriented engine. State S indicates a possible state in the dialog such as the one where the robot is waiting for the user's first utterance or the one where it is waiting for an answer to its previous question like "which one shall I take ?" Context γ is *identified* words before and after an unidentified one under consideration.

Fig. 7 illustrates the estimation of category C and word W using the probabilistic models:

- $P_{c-p}(C_p|C)$ is the probability that C_p is uttered just before the utterance of C.
- $P_{c-n}(C_n|C)$ is the probability that C_n is uttered just after the utterance of C.
- $P_{w-p}(W_p|W)$ is the probability that W_p is uttered just before the utterance of W.
- $P_{w-n}(W_n|W)$ is the probability that W_n is uttered just after the utterance of W.

For case (1) (i.e., erroneous recognition of a registered word), we search for the word $\hat{W}$ which is:

$$\hat{W} = \arg\max_{W} P(W|S, \gamma, R). \quad (1)$$

For case (2) (i.e., use of a synonym of a registered word), we search for the word $\hat{W}$ which is:

$$\hat{W} = \arg\max_{W} P(W|S, \gamma). \quad (2)$$

We here further examine eq. (1) only due to the space limitation. Eq. (1) is rewritten as:

$$P(W|S,\gamma,R)=\frac{\sum_C\{P(W|S,\gamma,C)P(C|S,\gamma)\}P(R|W,S,\gamma)}{\sum_W P(W,R|S,\gamma)} \simeq \frac{\sum_C\{P(W|S,\gamma,C)P(C|S,\gamma)\}P(R|W)}{\sum_W P(W,R|S,\gamma)} \quad (3)$$

where $\sum_C$ indicates the summation for categories C whose probability $P(C|S,\gamma)$ is larger than a threshold, and $\sum_w$ indicates the summation for words W belonging to the categories. Eq. (3) is obtained by considering that a recognized text R depends almost only on word W; $P(R|W)$ is called a *pronunciation similarity*.

An example of successful recognition of an unidentified word is as follows. A user asked the robot to take a blue PET bottle, by uttering "AOI (blue) PETTO BOTORU (PET bottle) WO TOTTE (take)". The robot however first recognized the utterance as "OMOI KUU TORABURU WO TOTTE". Since this includes unidentified words, the robot estimates their meanings using the above-mentioned method, and reached the conclusion that "OMOI" means "AOI" and "KUU TORABURU" means "PETTO BOTORU".

The recognition result of unidentified words are fed back to the system to update the database and the probabilistic models [11].

4 Easy Teaching of Mobile Manipulation

Usually service robots have to deal with much wider range of tasks (i.e., operations and environments) than industrial ones. An easy, user-friendly teaching method is, therefore, desirable for such service robots. Among previous teaching methods, direct methods (e.g., the one using a teaching box) are intuitive and practical but requires much user's effort, while indirect methods (e.g., teaching by demonstration [5, 4]) are easy but still needs further improvement of the robot's ability for deployment.

We, therefore, use a novel teaching method for a mobile manipulator which exists in between the above two approaches. In the method, a user teaches the robot a nominal trajectory of the hand and its *tolerance* to achieve a task. In this teaching phase, the user does not have to explicitly consider the structure of the robot but teaches the movement of the hand in the object-centered coordinates. The tolerance plays an importance role when the robot generates an actual trajectory in the subsequent playback phase; although the nominal trajectory may be infeasible due to the structural limitation, the robot can search for a feasible one within the given tolerance. Only when the robot fails to find the feasible trajectory, the robot plans a movement of the mobile base; that is, the redundancy provided by the mobile base acts as another tolerance in trajectory generation. Since the robot autonomously plans a necessary movement of the base, the user does not have to consider whether the movement is needed. The teaching method is well intuitive and does not require much effort from a user. In addition, it does not assume a high recognition and inference ability of the robot because the nominal trajectory given by the user has much information for planning feasible motions; the robot does not need to generate a feasible trajectory from scratch.

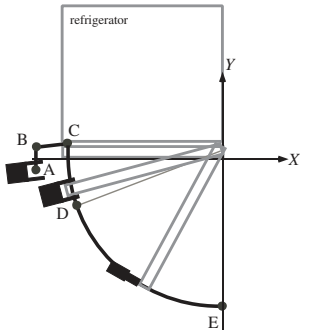


Fig. 8: A nominal trajectory for opening a door.

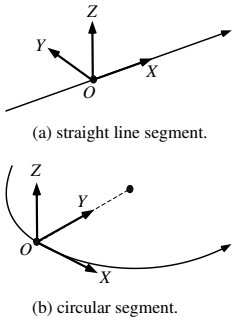


Fig. 9: Coordinates on segments.

The following subsections explain the teaching method, using the task of opening the door of a refrigerator as an example.

4.1 Nominal Trajectory

A nominal trajectory is the trajectory of the hand pose (position and orientation) in a 3D object-centered coordinate system. Among feasible trajectories to achieve the task, a user arbitrarily selects one, which can easily be specified by the user. To simplify the trajectory teaching, we currently set a limitation that a trajectory of hand position is composed of circular and/or straight line segments.

Fig. 8 shows a nominal trajectory for opening a door, which is composed of straight segments AB and BC and circular segments CD and DE set on some horizontal planes; on segment CD, the robot roughly holds the door, while on segment DE, the robot pushes it at a different height. The axes in the figure are those of the object-centered coordinates. On the two straight segments, the hand orientation is parallel to segment BC; on circular segment CD, the hand is aligned to the radial direction of the circle at each point; on circular segment DE, the hand tries to keep aligned to the tangential direction of the circle.

4.2 Tolerance

A user-specified trajectory may not be feasible (executable) due to the structural limitation of the manipulator. In our method, therefore, a user gives not only a nominal trajectory but also its tolerance. A tolerance indicates acceptable deviations from a nominal trajectory to perform a task; if the hand exists within the tolerance over the entire trajectory, the task is achievable. A user teaches a tolerance without explicitly considering the structural limitation of the robot. Given a nominal trajectory and its tolerance, the robot searches for a feasible trajectory.

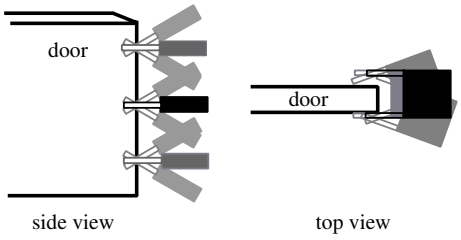


Fig. 10: An example tolerance for opening the door.

A user sets a tolerance to each straight or circular trajectory using a coordinate system attached to each point on the segment (see Fig. 9). In these coordinate systems, a user can teach a tolerance of positions relatively intuitively as a kind of the *width* of the nominal trajectory. Fig. 10 shows an example of setting a tolerance for circular segment CD, which is for opening the door, in Fig. 8.

4.3 Generating Feasible Trajectories

The robot first tries to generate a feasible trajectory within a given tolerance. Only when the robot fails to find a feasible one, it divides the trajectory into sub-trajectories such that each sub-trajectory can be performed without movement of the base; it also plans the movement between performing sub-trajectories.

4.3.1 Trajectory Division Based on Feasible Regions

The division of a trajectory is done as follows. The robot first sets via points on the trajectory with a certain interval (see Fig. 11). When generating a feasible trajectory, the robot repeatedly determines feasible poses (positions and orientations) of the hand at these points (see Sec. 4.3.2).

For each via point, the robot calculates a region on the floor in the object coordinates such that if the mobile base is in the region, there is at least one feasible hand pose. By calculating the intersection of the regions, the robot determines the region on the floor where the robot can make the hand follow the entire trajectory. Such an intersection is called a *feasible region* of the task (see Fig. 12).

Feasible regions are used for the trajectory division. To determine if a trajectory needs division, the robot picks up one via point after another along the trajectory and repeatedly updates the feasible region. If the size of the region becomes less than a certain threshold, the trajectory is divided at the corresponding via point. This operation continues until the endpoint of the trajectory is processed. Fig. 13 shows example feasible regions of the trajectory of opening the door shown in Fig. 8. The entire trajectory is divided into two parts at point V; two corresponding feasible regions are generated.

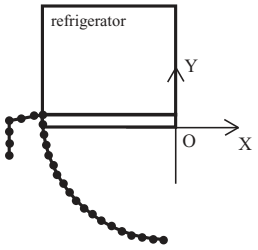


Fig. 11: Via points.

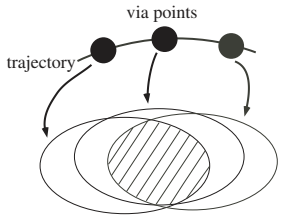


Fig. 12: Calculation of a feasible region for via points.

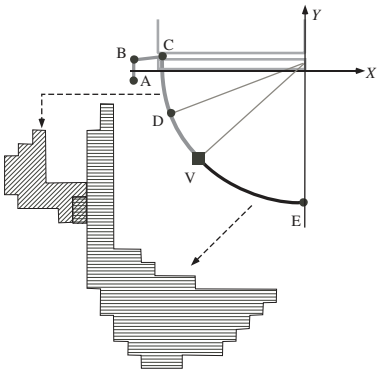


Fig. 13: Example feasible regions.

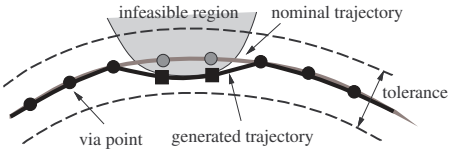


Fig. 14: On-line generation of a feasible trajectory.

4.3.2 On-line Trajectory Generation

A feasible trajectory is generated by iteratively searching for feasible hand poses for a sequence of via points. This trajectory generation is performed on-line because the relative position between the robot and manipulated objects varies each time due to the uncertainty in the movement of the robot base. The robot estimates the relative position before trajectory generation. The previously calculated trajectories can be, however, used as guides for calculating the current trajectory; all trajectories are expected to be similar to each other as long as the uncertainty in movement is reasonably limited.

Fig. 14 illustrates how a feasible trajectory is generated. In the figure, small circles indicate via points on a given nominal trajectory; two dashed lines indicate the boundary of the tolerance; the hatched region indicates the one where the robot cannot take the corresponding hand pose due to the structural limitation. A feasible trajectory is generated by searching for a sequence of hand poses which are in the tolerance and near to the given via points (two squares indicate selected via points). The bold line in the figure in-

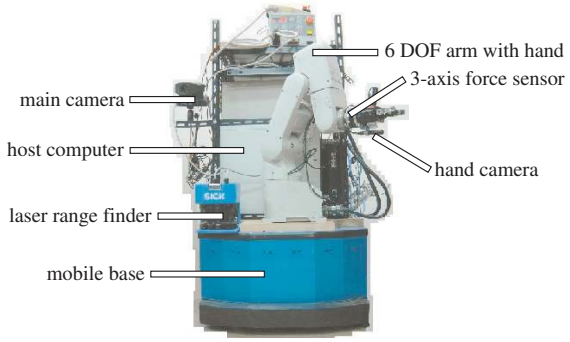
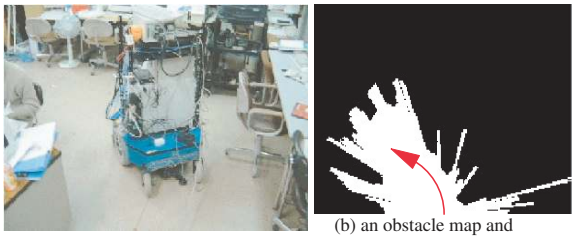


Fig. 15: Our service robot.



(a) movement of the robot. (b) an obstacle map and a planned movement.

Fig. 16: Obstacle avoidance.

Fig. 17: Fetch a can from a refrigerator.

dicates the generated feasible trajectory. In the actual trajectory generation, the robot searches the six dimensional space of hand pose (position and orientation) for the feasible trajectory.

During executing the generated trajectory, it is sometimes necessary to estimate the object position. Currently, we manually give the robot a set of necessary sensing operations for the estimation.

5 Manipulation and Motion Experiments

Fig. 15 shows our personal service robot. The robot is a self-contained mobile manipulator with various sensors. In addition to the above-mentioned functions, the robot needs an ability to move between a user and a refrigerator. The robot uses the laser range finder (LRF) for detecting obstacles and estimating the ego-motion [8]. It uses the LRF and vision for detecting and locating refrigerators and users. Fig. 16 shows a collision-avoidance movement of the robot. Fig. 17 shows snapshots of the operation of fetching a can from a refrigerator to a user.

6 Summary

This paper has described our personal service robot. The feature of the robot is a user-friendly human-robot interfaces including interactive object recognition, robust speech recognition, and easy teaching of mobile manipulation.

Currently the two subsystems, object and speech recognition and teaching of mobile manipulation, are implemented separately. We are now integrating these two subsystems into one prototype system for more intensive experimental evaluation.

Acknowledgment

This research is supported in part by Grant-in-Aid for Scientific Research from Ministry of Education, Culture, Sports, Science and Technology, and by the Kayamori Foundation of Informational Science Advancement.

References

[1] Morpha project, *http://www.morpha.de/*.

[2] R. Bischoff. Hermes – a humanoid mobile manipulator for service tasks. In *Proc. of FSR-97*, pp. 508–515, 1997.

[3] R. Bischoff and V. Graefe. Dependable multimodal communication and interaction with robotic assistants. In *Proc. of ROMAN-2002*, pp. 300–305, 2002.

[4] M. Ehrenmann, O. Rogalla, R. Zöllner, and R. Dillmann. Teaching service robots complex tasks: Programming by demonstration for workshop and household environments. In *Proc. of FSR-2001*, pp. 397–402, 2001.

[5] K. Ikeuchi and T. Suehiro. Toward an assembly plan from observation part i: Task recognition with polyhedral objects. *IEEE Trans. on Robotics and Automat.*, Vol. 10, No. 3, pp. 368–385, 1994.

[6] Y. Makihara, M. Takizawa, Y. Shirai, J. Miura, and N. Shimada. Object recognition supported by user interaction for service robots. In *Proc. of ICPR-2002*, pp. 561–564, 2002.

[7] Y. Makihara, M. Takizawa, Y. Shirai, and N. Shimada. Object recognition in various lighting conditions. In *Proc. of SCIA-2003*, 2003. (to appear).

[8] J. Miura, Y. Negishi, and Y. Shirai. Mobile robot map generation by integrating omnidirectional stereo and laser range finder. In *Proc. of IROS-2002*, pp. 250–255, 2002.

[9] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*, Vol. 42, No. 3-4, pp. 271–281, 2003.

[10] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Magaritis, M. Montemerlo, J. Pineau, J. Schulte, and S. Thrun. Towards personal service robots for the elderly. In *Proc. of WIRE-2000*, 2000.

[11] M. Takizawa, Y. Makihara, N. Shimada, J. Miura, and Y. Shirai. A service robot with interactive vision – object recognition using dialog with user –. In *Workshop on Language Understanding and Agents for Real World Interaction*, 2003 (to appear).