**Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007**

Multi-cue 3D Object Recognition in Knowledge-based Vision-guided Humanoid Robot System

Kei Okada, Mitsuharu Kojima, Satoru Tokutsu, Toshiaki Maki, Yuto Mori, Masayuki Inaba

***Abstract*— A vision based object recognition subsystem on knowledge-based humanoid robot system is presented. Humanoid robot system for real world service application must integrate an object recognition subsystem and a motion planning subsystem in both mobility and manipulation tasks. These requirements involve the vision system capable of self-localization for navigation tasks and object recognition for manipulation tasks, while communicating with the motion planning subsystem. In this paper, we describe a design and implementation of knowledge based visual 3D object recognition system with multi-cue integration using particle filter technique. The particle filter provides very robust object recognition performance and knowledge based approach enables robot to perform both object localization and self localization with movable/fixed information. Since this object recognition subsystem share knowledge with a motion planning subsystem, we are able to generate vision-guided humanoid behaviors without considering visual processing functions. Finally, in order to demonstrate the generality of the system, we demonstrated several vision-based humanoid behavior experiments in a daily life environment.**

I. Introduction

Humanoid robots are expected to assist human activities in daily life. Many researches have been involved in realizing humanoid robots in a daily environment [1]–[3]. In order to achieve these tasks in real world, sensory information is essentially important. Thus the development of general-purpose autonomous sensory based behavior generation system is a very important and challenging research area.

In this paper, we introduce our knowledge based humanoid system that integrates both motion planning system and visual object recognition system. While motion planning systems [4]–[6] only output a joint angle or joint torque sequences of a robot. We are interested in systems which generate motions with appropriate visual sensing(verification or recognition) [7]–[9]. In order to achieve this, we employ the knowledge based approach. Each object in the system contains not only 3D geometric shape, but manipulation knowledge for a motion planner and visual feature knowledge for a vision based object recognition system.

Section II describes our knowledge based humanoid robot system, then in section III and IV illustrate a knowledge based motion planning system and a visual recognition system respectively. In section V? we show vision based behavior examples of humanoid robot in daily life environment.

K. Okada, M. Kojima, S. Tokutsu, T. Maki, Y. Mori and M. Inaba are with the Graduate School of Information Science and Technology, The University of Tokyo, Engineering Building No. 2, 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan k-okada@jsk.t.u-tokyo.ac.jp

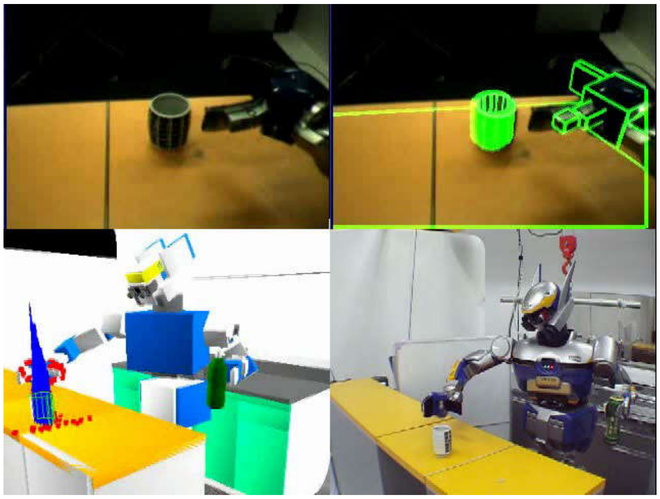


Fig. 1. Behavior example of knowledge based vision guided humanoid system

II. Knowledge-based Vision-guided Humanoid System

Fig.1 shows the concept of knowledge-based vision-guided humanoid system. The system relies on the humanoid robot programming system capable of three dimensional shape modeling [10]–[12]. The left bottom figure shows graphical viewer interface of our humanoid robot system in which all objects related to the robot task are modeled with associated knowledge. In this example, we modeled a kitchen, a bar counter, a cup and a plastic bottle which exists in the real environment shown in the right bottom figure. Top left image is robot's view and a simulated view image in model environment is super imposed to a real view image as the top right image.

Motion planner generates sequence of robot posture using these knowledge which are also used in object recognition system in order to examine if the interest object exists in real world and update its location in the model environment for a motion planner. The key feature of our system is:

1) Motion planning subsystem and object recognition subsystem share the same object representation which enables simple system design.
2) Humanoid behavior programmer does not have to explicitly consider the object recognition program. Since the framework of object recognition subsystem is so general that is able to apply any kind of objects for both manipulation and navigation.

1-4244-0912-8/07/$25.00 ©2007 IEEE.

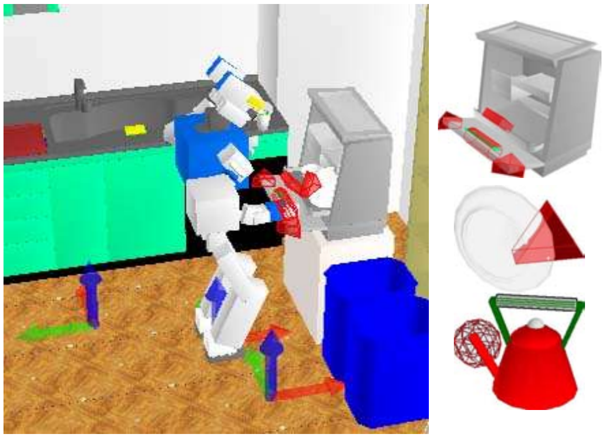


Fig. 2. An example of object and environment model with manipulation knowledge in kitchen task environment
Three arrows on the floor show spot information. Right figures show 3D shape model and handle information which is displayed as red triangles and cylinders. A red sphere in the kettle figure shows manipulation information.

III. Knowledge-based Tool Manipulation Motion Planner [6]

This section briefly describes a tool manipulation motion generation method. In contrast to the previous humanoid motion planner researchesthat focus on the planner algorithm of a large search space, we focus on knowledge representation and description for generating tool manipulation behavior. Our method relies on a manipulation knowledge [13] which is used to describe robot's motion in a simple manner. In this approach, a tool manipulation motion is described as a key frame motion of a reference coordinates which we call `manipulate` knowledge. Thus, the motion planner generates a sequence of whole body postures from the key frame motion of `manipulate`.

A. Object model with manipulation knowledge

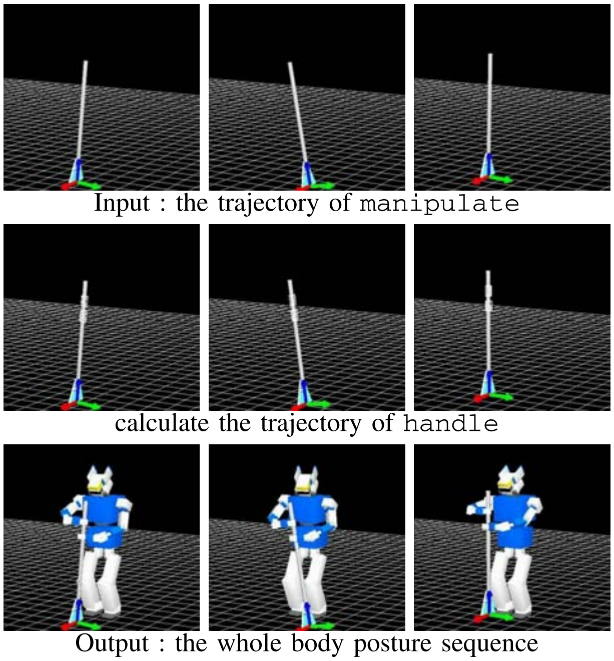
Tool manipulation motion of a humanoid robot is modeled as "a robot standing at certain position controls whole body joint angles while holding a target object".

From this viewpoint, we defined a knowledge for tool manipulation motion generation as followings. A robot is located on the `spot` coordinate to manipulate a tool, grasp the tool object according to the `handle` coordinate and constraints and manipulate a tool with reference to a `manipulate` coordinate(See Fig.2 for detail).

B. Tool manipulation motion planner using the knowledge

Fig.3 shows the procedure of the tool manipulation planner. Upper figures shows the key frame motion of the `manipulate` coordinate which is the input of this planner. Lower figures show the whole body posture sequence, which is the output of the planner. The tool manipulation planner described in this section generates complex whole body motions (sequence of 30 D.O.F. joint angles) from the simple input data (sequence of 6 D.O.F. coordinates).

Fig.4 shows tool manipulation behavior of an actual life-sized humanoid robot by using this method.



Input : the trajectory of manipulate

calculate the trajectory of handle

Output : the whole body posture sequence

Fig. 3. Procedure of the tool manipulation planner
This planner generates complex whole body motions (sequence of 30 D.O.F. joint angles) from the simple input data (sequence of 6 D.O.F. coordinates).

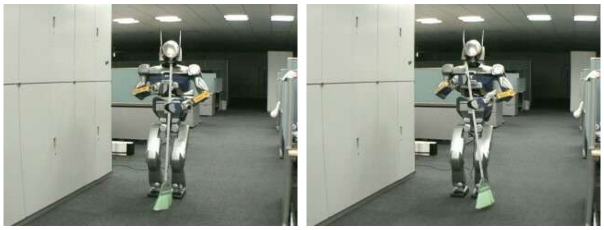


Fig. 4. Tool manipulation behaviors of a humanoid
Both sweeping and vacuuming behavior are able to generate using object models, tool manipulation knowledge and trajectories of a manipulation coordinate.

C. Motion generation programming

A humanoid behavior is programmed as following pseudo codes by the use of tool manipulation motion planner. The `manipulate` function requires the target object and the motion list with reference to the `manipulate` coordinates as its key frame motion sequence. In the function, it calculate the motion sequence of the object and calls a `:manipulate` method of a robot body to generate whole body motion. In the `:manipulate` method, it uses the `handle` knowledge to determine the reaching position and solves the whole body IK method to generate the whole body posture.

```
(defun manipulate (obj motion-list)
 (let (manip manip-list)
  (setq manip (send obj :manipulate))
  (dolist (motion motion-list)
     (push (send manip :transform motion) manip-list))
  (send *robot* :manipulate obj manip-list)))
```

IV. Knowledge based Probabilistic Visual Object Recognition

A. Object recognition with visual cue knowledge

In order to recognize objects, we defined visual cue knowledge as followings.

- `Shape` information: 3D object shape information for 3D distance information based object recognition
- `ColorHistogram` information: Color histogram of an object surface for color texture based recognition
- `VisibleEdge` information: visible(detectable) edges on an object surface for 2D edge based recognition

We adopt the Particle Filter(PF) algorithm [14], [15], which is widely used in vision based object tracking systems because of its robust characteristics.

B. Particle filter algorithm for object recognition

Particle filtering algorithm can be described as followings:

1) Measure: Calculate the weight (probability) $w_t^{(i)}$ for particle $x_t^{(i)}$ using the observation model.
2) Resample: Resample particles using their weight (probability) to generate an un-weighted approximation of $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$.
3) Update: Update particles to predict new state vector x_t by applying the process dynamics model

For the particle filter based recognition system, $\boldsymbol{x}_t$ denotes the state vector of the interest object (6 elements for a 3D position and roll-pitch-yaw rotation), and $\boldsymbol{z}^t$ the measurement vector (visual cues). The posterior distribution at the frame t is $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$ which is represented by a set of N particles (hypothesis) $\boldsymbol{x}_t = (x_t^{(1)}...x_t^{(N)})$ and their associated weights (probabilities) $\boldsymbol{w}_t = (w_t^{(1)}...w_t^{(N)})$.

Since the probability of each particle is measured as followings, calculating $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$ from the visual processing results is the vital part of the recognition system.

$$w_t = w_{t-1}\frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t)\ p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{q(\boldsymbol{x}_t|\boldsymbol{X}_{t-1},\boldsymbol{Z}_t)} = w_{t-1}\ p(\boldsymbol{z}_t|\boldsymbol{x}_t)$$

C. Multi-cue integrated visual recognition

We assume that distributions of visual cues are conditionally independent, the conditional density $p(z_k|x_k)$ can be written as the product of density function of each visual cues [16]. In this paper, three visual cues are integrated: 3D feature points, color histogram and 2D straight edge.

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t) = p_{point}(\boldsymbol{z}_t|\boldsymbol{x}_t)\ p_{color}(\boldsymbol{z}_t|\boldsymbol{x}_t)\ p_{edge}(\boldsymbol{z}_t|\boldsymbol{x}_t)$$

D. Multi-cue observation model

1) ***3D Feature Point:*** 3D feature points are generated through following two steps. First, the 2D feature points are generated by using the KLT feature extraction method [17], which the feature points are located by calculating the minimum eigenvalue of each 2 by 2 gradient matrix. Then, the correlation based stereo matching is applied to calculate the disparity of the feature points. Then we obtain the 3D distance of the points from the camera origin by assuming that the internal and external camera parameters are calibrated.

The likelihood is defined as

$$p_{shape}(z|x) = \exp[\frac{(\frac{1}{|P|}\sum_{p\in P} D_{point}(p, F_{ref}^{visible})^2)}{2\ \sigma_{shape}^2}]$$

where the $F_{ref}^{visible}$ is a set of visible faces from the robot's view point among the all faces of the object F_{ref}^{all}, P is a set of 3D feature points its distance from the nearest faces in $F_{ref}^{visible}$ under threshold and $|P|$ denotes the number of P. $D_{point}(p_1, p_2)$ denotes the squared 3D distance between two points. σ_{shape} is a user defined weight value.

2) ***Color Histogram:*** We use the following likelihood for the HSV color histogram.

$$p_{color}(z|x) = \exp[-\frac{B(h_{B_x}, h_{B_{ref}})^2}{2\ \sigma_{color}^2}]$$

where h_{B_x} denotes the color histogram at the interest area and $h_{B_{ref}}$ as the reference color histogram model.

The interest area is a rectangular region on the image plane. The size of this region is calculated based on the 3D object model information for each particle. Since each particle contains the position information of the 3D object model, we obtain the projected object area on the camera screen using the robot's viewpoint location information. Then the bounding box of this area becomes the interest area for histogram calculation.

The similarity between two color distributions can be measured using the Bhattacharyya coefficient [18]:

$$B(\boldsymbol{h}_1,\boldsymbol{h}_2) = [1-\sum_{b=1}^{Nb}\sqrt{h_{b,1}h_{b,2}}]^{1/2}$$

where $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are normalized histograms and $B(\boldsymbol{h}_1,\boldsymbol{h}_2)$ ranges from 0 to 1 with 0 means that the two histograms are same. σ_{shape} is also a user defined weight as above.

3) ***2D Straight Edge:*** 2D straight edge cue is used especially for artificial objects with less texture and colors. This type of object is often seen in the daily life environment such as a table, a refrigerator, a door, a drawer and so on.

First we apply the Canny edge detection method to extract reliable edges from a input image, then we extract straight edges $E^{2D}(= e^{(1)}...e^{(L)})$ as followings:

a Generate a edge between start and end points
b Calculate distance between the edge and the farest point
c Divide the edge at the point if the distance is under the threshold

The 3D edges associated with the object model E_{ref}^{3D} are projected on the image plane to obtain the 2D reference edges E_{ref}^{2D}.

To measure the similarity between E^{2D} and E_{ref}^{2D}, We first divide E_{ref}^{2D} into edge segments $e_{ref}^{(1)}...e_{ref}^{(M)}$ with fixed length. Then, for each edge segment $e_{ref}^{(m)}$, we find the nearest straight edge segment among $E^{2D}(= e^{(l)})$. For each edge $e^{(l)}$ by evaluating the distance value as

$$value(e_{ref}^{(m)}, e^{(l)}) = d\ (2 - a)$$

where

$$\begin{aligned} d &= \begin{cases} edge\_dist(e_1, e_2) & edge\_dist(e_1, e_2) \geq d_{thr} \\ 0 & edge\_dist(e_1, e_2) < d_{thr} \end{cases} \\ a &= \begin{cases} angle(e_1, e_2) & angle(e_1, e_2) \geq a_{thr} \\ 0 & angle(e_1, e_2) < a_{thr} \end{cases} \end{aligned}$$

$edge\_dist(e_1, e_2)$ is the distance between the edge e_1 and the starting point of the edge e_2 plus the distance between the edge e_1 and the end point of the edge e_2. $angle(e_1, e_2)$ is the angle between two edge segment which is calculated as $angle = \overrightarrow{e_1} \times \overrightarrow{e_2}$.

Then we find the nearest edge segment pair $e_{ref}^{(m)}, e^{(min)}$. Let the $|value|$ the number of non-zero $value(e_{ref}^{(m)}, e^{(min)})$, the similarity of two edge segments $D_{edge}(E_1, E_2)$ becomes

$$\frac{\sum value(e_{ref}^{(m)}, e^{(min)})}{|value|}$$

Finally, we obtain the likelihood for the 2D straight edge cue as below.

$$p_{edge}(z|x) = \exp[\frac{(D_{edge}(E^{2D}, E_{ref}^{2D}))^2}{2\ \sigma_{edge}^2}]$$

E. Vision guided behavior motion generation without considering visual processing

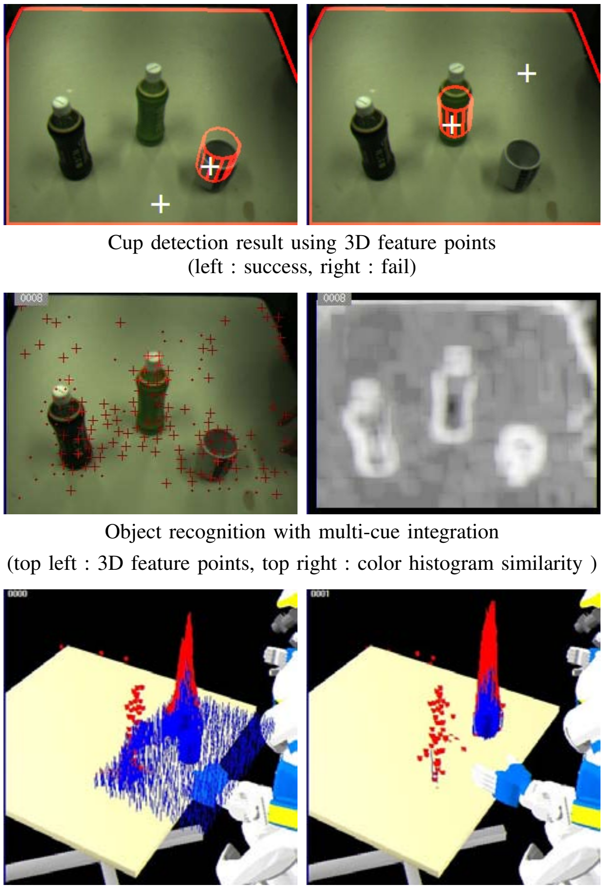
In order to evaluate a confidence of an object recognition results, we use diagonal of covariance matrix of the posterior distribution ($diag(var[\boldsymbol{x}_t])$) and height weight value($\boldsymbol{w}_t^{(i_{max})}$).

Finally we update the `manipulate` function to `manipulate-with-vision` function.

```
(defun manipulate-with-vision (obj motion-list)
 (let (manip manip-list)
  (setq manip (send obj :manipulate))
  (setq feat  (send obj :visualfeatures))
  (unless (visual-reocognition feat)
    (return-from manipulate-with-vision))
  (dolist (motion motion-list)
     (push (send manip :transform motion) manip-list))
  (send *robot* :manipulate obj manip-list)))
```

The `visual-recognition` function apply particle filter based object recognition method using `feat`(visual cue knowledge) information, evaluate a result using its confidence and update object position in a model if it has enough confidence.

Note that the function `manipulate-with-vision` takes same information as the `manipulate` function. It does not require any visual processing related arguments which is embedded in the knowledge. Therefore, the programmer does not have to consider the visual processing to generate vision-guided behavior of a robot.



Cup detection result using 3D feature points
(left : success, right : fail)

Object recognition with multi-cue integration
(top left : 3D feature points, top right : color histogram similarity)

Blue and red lines show likelihood of each particle. Blue lines indicate color histogram similarity and red is 3D features point based similarity.

Fig. 5. Visual object recognition with multiple visual cues

F. Visual object recognition implementation and experiments

*1) **Experimental setup:*** We use a humanoid robot HRP2JSK equipped with calibrated stereo camera system [12]. We assume that joint angle sensors of the robot are reliable, the robot is rigid enough and robot kinematics parameters are known, thus the head position of the robot can be calculated. However, the relative coordinate from the head position to the stereo camera origin is unknown, then we also calibrated hand-eye coordinates.

In the right top image of the Fig.1, the robot posture in the model is update thorough the joint sensor information and the shape of the arm of the robot and the shape of the target in the model environment is superimposed in the real view image. This image shows that the stereo camera parameter and hand-eye coordinates are calibrated.

*2) **Object recognition experiments:*** Fig.5 shows the result of object recognition with multi visual cue. In this example, we assume the position of the object lies on the 2D plane, thus the state vector becomes two dimensional vector as $\boldsymbol{x}_t^{(i)} = (x, y)^t$ and we modeled a cup as a cylinder shape. The top figures show that there are ambiguities between plastic bottles and a cup when only 3D features points are considered, then we introduce color histogram

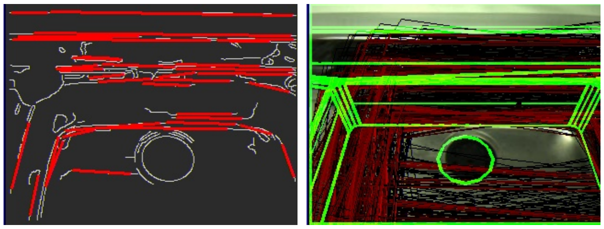


Fig. 6. A self localization example of the multi cue 3D object recognition system: Left: Edge image, The canny edge detector results and extracted straight edge segments(red lines). Right: Result image, Green super imposed sink shape is the result. Black and red lines indicate particles. Since the robot knows the kitchen object is fixed to the environment, thus a relative position of the robot from the kitchen is calculated.

based similarity. In the top right image of middle figures, the whiter pixel has higher similarity between input color image and color histogram model associated with the cup model. The bottom figures show results. Blue and red lines show likelihood of each particle. Blue lines indicate color histogram similarity and red lines indicate 3D features point based similarity. The left figure shows initial configuration and right the right figure shows the result after several iteration.

*3) **Self localization experiments:*** Fig.6 is a self localization example. In this example, only edge information is used. In the left image, white edges are results of the canny edge detector and red straight lines are extracted straight edge segments. In the right image, green super imposed sink shape is the result. Black and red lines indicate particles. Since the robot knows that the kitchen object is fixed to the environment, then a relative position of the robot from the kitchen is calculated.

V. Vision-based Humanoid Behavior Experiments

*A. **Demo scenario***

From Fig.7 to Fig.9 show the vision based humanoid behavior experiments. The scenario is as followings.

1) Grasp the cup and the plastic bottle
2) Pour the tea from the plastic bottle into the cup
3) Place the cup and the plastic bottle
4) Grasp the cup and move to the sink
5) Open the water outlet and wash the cup

To achieve this task, the robot has to recognize the cup, the plastic bottle, the sink and water flow. In Fig.7 and Fig.8, the robot detect the position of the cup and the bottle to grasp them. In the Fig.8, blue colored lines on the counter in lower figures represent the particle and its weight. In the lower left figure, there are no tall line, that means no particle has high confidence. On the other hand, in the lower right figure, the particles around the cup position have high confidence.

In the Fig.9, the robot detect current position using the method described in Fig.6, then the robot detect the position of the water flow to wash the cup. The water flow is modeled as a cylinder and 3D feature points are use to calculate the similarity.

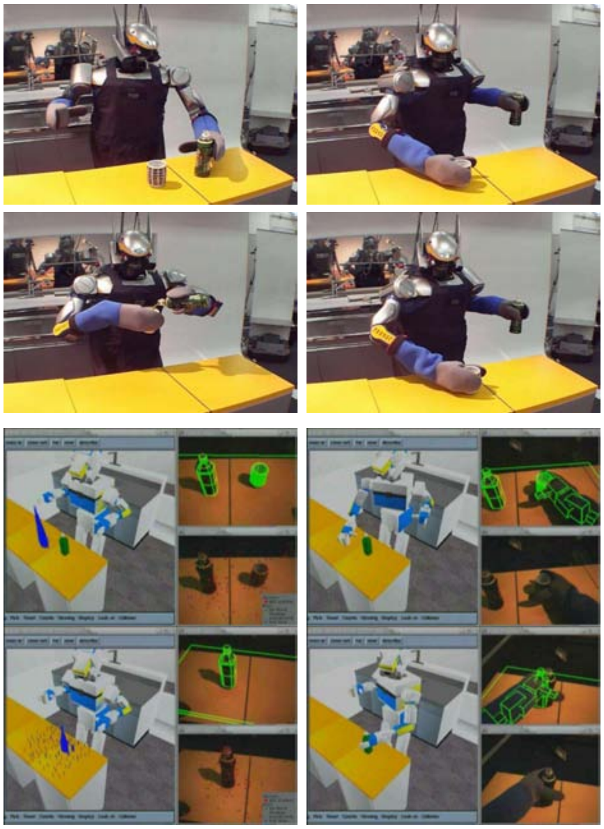


Fig. 7. Humanoid behavior examples in daily life environment based on visual object recognition: The robot recognize the cup and the plastic bottle to grasp and pour the tea.

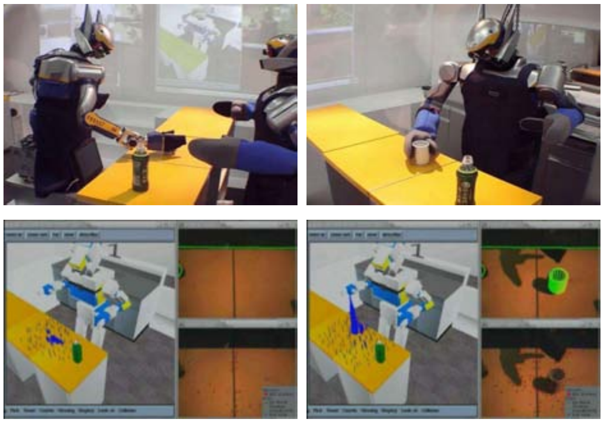


Fig. 8. The robot recognizes the cup on the counter. The blue colored lines on the counter in lower figures represent the particle and its weight. In the lower left figure, there are no tall line, that means no particle has high confidence. On the other hand, in the lower right figure, the particles around the cup position have high confidence.

*B. **Knowledge data base for the experiments***

In order to archive this experiment, we modeled 5 objects in demo environment. For an object manipulation, we modeled a cup and a plastic bottle with manipulation knowledge and visual cue knowledge which are a 3D shape and a color histogram(Fig.5). Water flow is also modeled using 3D shape knowledge(Fig.9). For a navigation task, a bar

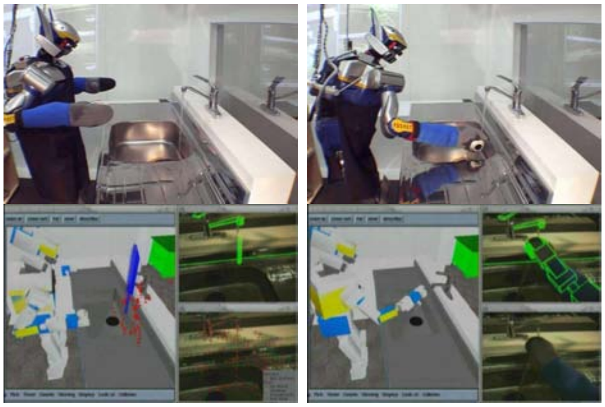


Fig. 9. The humanoid robot detect the position of the water flow to wash the cup. The water flow is modeled as a cylinder and 3D feature points are use to calculate the similarity.

counter and a kitchen sink are described with visible edge knowledge(Fig.6).

Currently, these visual knowledge are modeled manually. A 3D shape and a visual edge features are strong enough to be used as an absolute reference. We use HSV color space for histogram based recognition, since HSV is robust to the illumination changes. However throughout the experiments, we have to change the histogram set when the color of the table changes or spot lights are added.

VI. Conclusion

This paper presents an object recognition subsystem of our knowledge-based vision-guided humanoid robot system. The humanoid behavior experiment is a part of final demo of "The Real-world Information System Project" which was performed several times in front the domestic and international medias without failure. This demo indicates the robustness of our system.

The key feature of our vision system is:

1) Very robust visual object recognition system based on multi-cue integration and particle filter based stochastic approach.
2) The developed system is able to utilized for both navigation tasks and manipulation tasks by using movable/fixed knowledge of the object.
3) The object recognition subsystem and the motion planning subsystem are tightly connected and integrated by sharing the same object and environment knowledge. This feature makes possible to automatically generate visual-guided behaviors.

The key challenge in humanoid robotics research is integration of various kinds of systems. Thus the design concept or developmental methodology of integrated system is one of the important issues. In this paper, we presented knowledge centered integration of vision and motion subsystems. This approach enables subsystem to perform effectively by communicating each other through shared knowledge.

Limitations of the system are: 1) Currently we manually modeled knowledges, development of manipulation and visual knowledge acquisition behavior is required. 2) The system does not recognize other robots or humans. Human or robot activities recognition and integration with object recognition system are required.

References

[1] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent ASIMO: System overview and integration. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*, pages 2478–2483, 2002.

[2] R. Dillmann, P. Steinhaus, and R. Becher. ARMAR II - A Learning and Cooperative Multimodal Humanoid Robot. *International Journal on Humanoid Robotics*, 1(1):143–156, 2004.

[3] R. Ambrose, S. Askew, W. Bluethmann, and M. Diftler. Humanoids Designed to do Work. In *In Proceedings of the IEEE International Conference on Humanoid Robots (Humanoids 2001)*, pages 173–180, 2001.

[4] F. Gravot, R. Alami, and T. Simeon. Playing with several roadmaps to solve manipulation problems. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 2311–2316, 2002.

[5] J.J. Kuffner and K. Nishiwaki and S. Kagami and M. Inaba and H. Inoue. Motion planning for humanoid robots. In *In Proceedings of 11th International Symposyum on Robotics Research (ISRR'03)*, page 20, 2003.

[6] Kei Okada, Takashi Ogura, Atsushi Haneda, Junya Fujimoto, Fabien Gravot, and Masayuki Inaba. Humanoid Motion Generation System on HRP2-JSK for Daily Life Environment. In *2005 IEEE International Conference on Mechatronics and Automation (ICMA05)*, pages 1772–1777, 2005.

[7] R.C.Bolles. Verification Vision for Programmable Assembly. In *Proceedings of 5th International joint Conference on Artificial Intelligence*, pages 579–575, 1977.

[8] J. Miura and K. Ikeuchi. Task-Oriented Generation of Visual Sensing Strategies in Assembly Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):126–138, 1998.

[9] Kei Okada, Mitsuharu Kojima, Yuichi Sagawa, Toshiyuki Ichino, Kenji Sato, and Masayuki Inaba. Vision based behavior verification system of humanoid robot for daily environment tasks. In *2006 6th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)*, pages 7–12, 2006.

[10] T. Matsui. Multithread object-oriented language euslisp for parallel and asynchronous programming in robotics. In *Workshop on Concurrent Object-based Systems, IEEE 6th Symposium on Parallel and Distributed Processing*, 1994.

[11] M.Inaba, S. Kagami, F. Kanehiro, Y. Hoshino, and H. Inoue. A Platform for Robotics Research Based on the Remote-Brained Robot Approach. *The International Journal of Robotics Research*, 19(10):933–954, 2000.

[12] K. Okada, T. Ogura, A. Haneda, D. Kousaka, H. Nakai, M. Inaba, and H. Inoue. Integrated System Software for HRP2 Humanoid. In *Proc. of International Conference on Robotics and Automation (ICRA'04)*, pages 3207–3212, 2004.

[13] Kunikatsu Takase. Skill of intelligent robot. In *Proc of 6th International Joint Conference on Artificial Intelligence,(IJCAI1979)*, pages 1095–1100, 1979.

[14] Genshiro Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, March 1996.

[15] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[16] Jan Giebel, Dariu Gavrila, and Christoph Schnörr. A bayesian framework for multi-cue 3d object tracking. In *ECCV (4)*, pages 241–252, 2004.

[17] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[18] Patrick Perez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 661–675, London, UK, 2002. Springer-Verlag.