The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

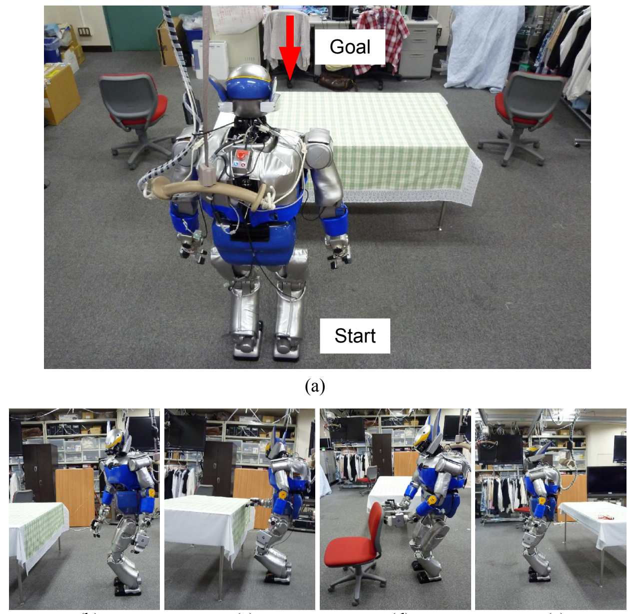
Working with Movable Obstacles Using On-line Environment Perception Reconstruction Using Active Sensing and Color Range Sensor

Yohei Kakiuchi and Ryohei Ueda and Kazuya Kobayashi and Kei Okada and Masayuki Inaba

***Abstract*— We propose a strategy for a robot to operate in an environment with movable obstacles using only on-board sensors, with no previous knowledge of the objects in that environment. Movable obstacles are detected using active sensing and a color range sensor, and when an obstacle is moved, the perception of the environment is reconstructed.**

Active sensing is defined as the classification of an object as either movable or static after the robot tries to push the object using its arm. This classification is collectively based on force sensor inputs, joint angles, and color range sensor inputs. In order to gather information from the environment, we use a color range sensor consisting of a TOF (Time of Flight) range sensor and conventional stereo cameras.

Finally, we show experimental result in the environment with movable obstacles such as a table and chairs. Humanoid robot HRP-2 detects that a chair is a movable obstacle, moves the chair to clear a path to its goal, and then reaches the goal.



(a)

(b) (c) (d) (e)

Fig. 1. Example of a scene with movable and static obstacle

I. INTRODUCTION

Robots working in nursing care, home assistance, office service, and other environments shared with humans are required to recognize changing environment and navigate by avoiding collision with fixed obstacles and moving movable obstacles out of the way to clear a path. Humanoid robot's range of abilities has been expanded to include being able to move movable objects to clear a path to its destination [1], [2]. In order to execute actions effectively in such a complex environment, on-line reconstruction of the robot's perception of the environment in response to environmental changes is needed. In addition to passive sensing, active sensing is effective for recognizing movable obstacles.

Humanoid robots that can operate in several environments using 3-D stereo vision or a range sensor have been studied [3]–[6] . Accurate 3-D sensing is necessary for recognizing complex environments. Color is also useful information for environment perception when combined with 3-D point information. In this paper, we adopt a color range sensor consisting of a TOF (Time of Flight) range sensor and conventional color stereo cameras to construct a 3-D point cloud with color.

Fig. 1 shows an example of the type of problem we address in this paper. The robot is directed to go to the goal in an environment with an obstacle (Fig. 1 (a)). The robot looks around and detects the obstacle it needs to avoid in order to prevent collision (Fig. 1 (b)). If the path to the goal is not found, the robot tries to push the obstacle in order to determine whether or not it is movable (Fig. 1 (c)). If the object is determined to be movable, the robot pushes it out of the way to clear a path to the goal (Fig. 1 (d)). Then, the robot proceeds to the goal (Fig. 1 (e)).

This paper is organized as follows. In the next section, related works are surveyed. The action strategy proposed in this paper is described in Section III. In Section IV, environment perception by the color range sensor is described. In Section V, how to detect a movable obstacle by combining information from the color range sensor, force sensors, and joint angles is described. Experimental results are presented in Section VI. and conclusions are described in Section VII .

II. RELATED WORKS

We proposed a motion planner in an environment with movable obstacles [1], in which the approach was to construct an environment manipulation task graph and solve the task using a navigation path planner and a whole-body motion planner. Stilman [2] solved the navigation planning problem among movable obstacles using the geometric model of obstacles. In this paper, we propose an action strategy in an environment with movable obstacles with no previous knowledge of the objects, such as their geometry or grasping points. Therefore, the robot needs to detect which obstacles are movable by trying to push them.

Y. Kakiuchi, R. Ueda, K. Kobayashi, K. Okada and M. Inaba are with Graduate School of Information Science and Technology The University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku, 113-8656 Tokyo, Japan {youhei,ueda,k-okada,inaba}@jsk.t.u-tokyo.ac.jp

Humanoid robots using on-line environment recognition have been researched [7], [8]. Michel [7] has proposed an on-line environment recognition system for a biped robot using on-board sensors and external optical motion capture system. In [8], a biped robot navigation in an environment with previously unknown obstacles has been researched using a pivoting range sensor for sensing the environment on-line. In this paper, the robot moves obstacles depending on the results of movable object detection, and the perception of the environment is reconstructed after each time an object is moved.

There has been research done using the fusion of a range sensor and a color camera [9], [10]. Categorization of objects in a kitchen environment has been achieved by using statistical relational learning method in [9]. In [10] localization and mapping for a mobile robot expected to explore an unknown environment has been researched.

ICP(Iterative Closest Point) is an algorithm for estimating the transformation between two point clouds [11] by minimizing the distance of closest points. A more accurate estimation has been achieved [12] by using a 3-D point cloud with color. In this paper, the color-ICP algorithm was used for detecting the displacement of the movable obstacle pushed by the robot.

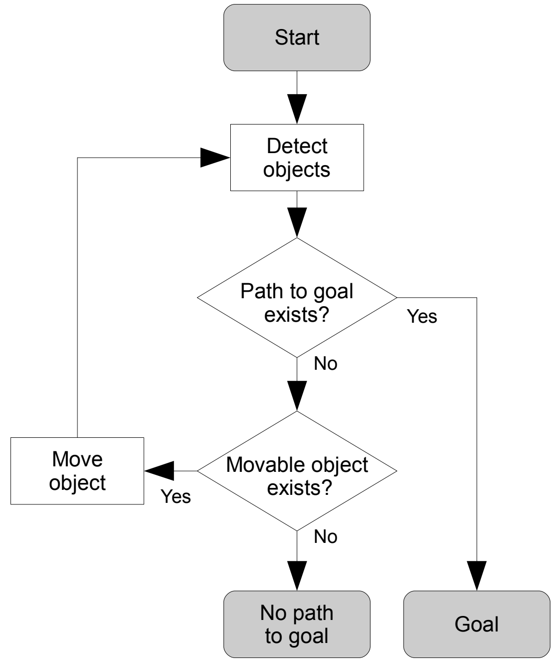


Fig. 2. Execution flow for working with movable obstacles

III. STRATEGY FOR WORKING WITH UNKNOWN OBJECTS

In this section, a strategy for working in an environment with movable obstacles is described. The proposed strategy consists of obstacle recognition using a point cloud captured by range sensor, active sensing for detecting movable obstacles, pushing a movable obstacle away, and them performing on-line reconstruction of the robot's perception of the environment.

Obstacles were detected by generating simple geometric shapes using point clouds obtained using a "looking-around" motion. Details are described in Section IV-C. "Detect Object" in Fig. 2 corresponds to this explanation.

At first, path planning with avoiding obstacles is executed. If a path to the goal is found, the robot navigates to the goal. The execution flow reaches "Goal" in Fig. 2.

When there is no path to the goal, the robot tries to detect movable objects using active sensing. "Movable object exists?" corresponds to this action. The robot starts with the obstacle nearest to its location and iterates until it finds a movable obstacle or detects that all obstacles are static objects. Details of detecting movable obstacles are described in Section V.

If no movable obstacle are found, the execution flow reaches "No path to goal" in Fig. 2. When a movable obstacle is found, the robot moves the obstacle. Then, the execution flow returns to "Detect Object".

Fig. 3 shows an example of an environment in which a table is a static object, and two chairs are movable objects. "S" indicates the starting location of the robot. "G" indicates the goal. Circled number from 1 to 3 indicate the objects set in this environment. The path planner described in this paper uses the rapidly-exploring random tree (RRT) path planning algorithm. In collision detection between the robot and obstacles, the robot is assumed to be a bounding cylinder.

The robot executes actions in this environment as follows. The path to the goal is not found in this setting because distance between the table and the chair is shorter than the diameter of the cylinder assumed to be the robot. The sequence for detecting movable obstacles is then executed as follows. The robot moves to a suitable position for trying to push the obstacle, pushes the obstacle, and then detects movement of the obstacle.

The robot push the obstacle forward and reconstructs a perception of obstacles. If a path to the goal can be found, the robot reaches the goal.

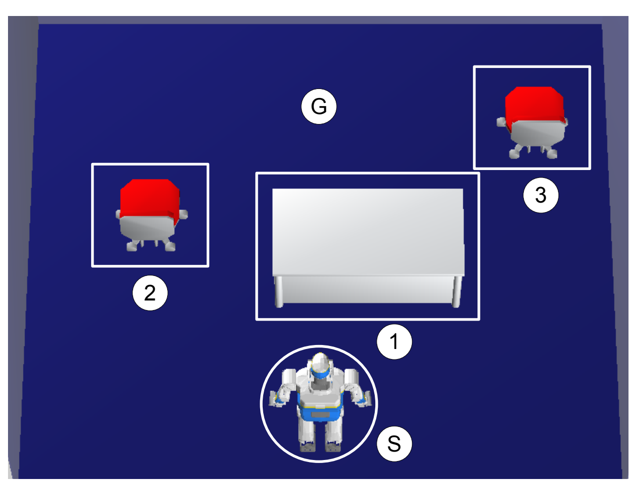


Fig. 3. Example of an arrangement of obstacles in an environment

IV. RECOGNITION OF THE ENVIRONMENT WITH COLOR RANGE SENSOR

A. Color Range Sensor

The color range sensor (shown in Fig. 4) used in this paper consists of a TOF range sensor (Swissranger, SR-4000) and two conventional IEEE1394 color cameras(Pointgray, Flea2). The focal length of the color cameras is 4 mm, and the base length of stereo cameras is 110 mm. The focal length of the camera is selected to be wider than the range sensor. The horizontal view angle of the range sensor is 43.6 deg and the vertical view angle is 34.6 deg. A 3-D point cloud, with color, that has 25,344 (176 x 144) points can be constructed at more than 10 frames per second using this color range sensor.

The range sensor is placed in between the stereo cameras. This alignment allows the estimation of the color corresponding to a point using images from the two cameras. Therefore, errors corresponding to the background side can be reduced.

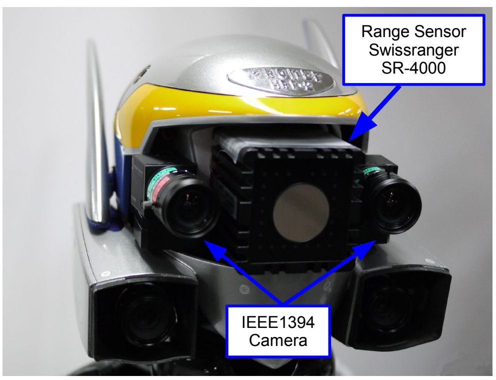


Fig. 4. The color range sensor, which consists of a range sensor and color cameras, is fixed at the head of HRP-2

Camera calibration is executed in two steps. First, intrinsic camera parameters of the stereo cameras and the range sensor are calibrated. In the second step, external parameters of the range sensor and the stereo cameras are calibrated simultaneously by showing the robot a checkerboard (shown in Fig. 5).

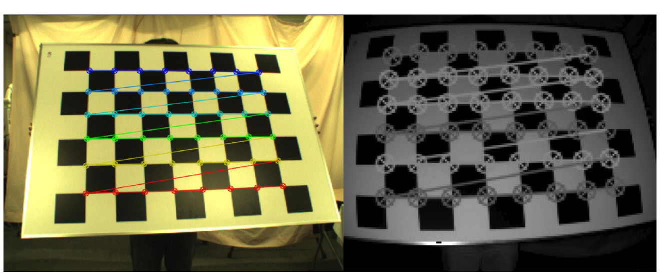


Fig. 5. A checkerboard as seen by the left camera of stereo cameras (left), and the intensity image seen by the range sensor (right)

Fig. 6 shows an example of a 3-D point cloud with color captured by the color range sensor located on the robot head and the image captured by the left camera.

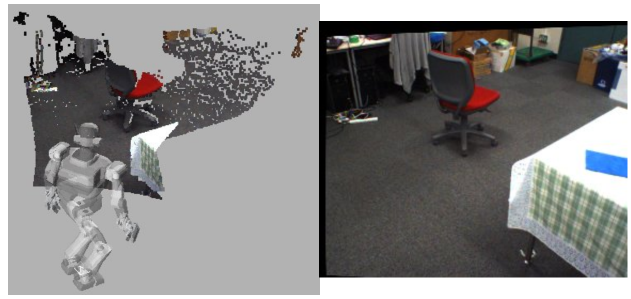


Fig. 6. Constructed color point cloud(left), and the corresponding image captured by the left camera(right)

B. Floor Detection

The view angle of the color range sensor is too narrow for a humanoid robot to recognize the environment around it. Thus, the color range sensor is panned and tilted by swinging the head. Fig. 7 shows 15 point clouds gathered at tilt angles of 45, 30 and 20 degree, for each pan angle of 50, 25, 0, -25, and -50 degree.

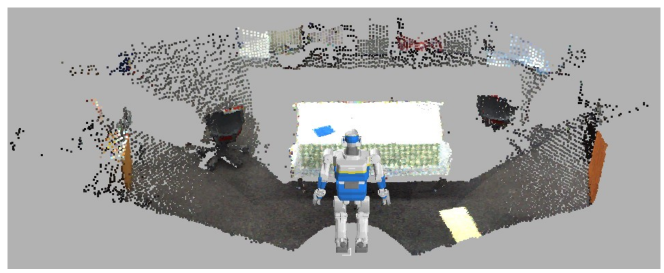


Fig. 7. Gathered 3D point cloud of the room. 15 point clouds are shown in this figure.

Removing floor points from gathered point clouds is necessary for detecting obstacles on the floor. The floor detecting method in this paper is based on the assumption that the place where robot is standing is the floor of the environment. From this assumption, points that have similar color and that have similar elevation to known floor's points directly in front of the robot are assumed to be the floor.

$P(z)$, Gaussian distribution of the height of the floor, is defined as follows:

$$P(z) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-(\frac{z-\mu_z}{2\sigma_z})} \quad (1)$$

where z is z component of the points, σ_z is the variance of known floor's points, and μ_z is the average of known floor's points.

$P(\boldsymbol{c})$, Gaussian distribution of the color-vector of the floor, is defined as follows:

$$P(\boldsymbol{c}) = \frac{1}{(2\pi)^{3/2}\sqrt{|\boldsymbol{\Sigma_c}|}} exp[-\frac{1}{2}(\boldsymbol{c}-\boldsymbol{\mu_c})\boldsymbol{\Sigma_c}^{-1}(\boldsymbol{c}-\boldsymbol{\mu_c})^t] \quad (2)$$

where $\boldsymbol{c}$ is the RGB color vector, Σ_c is the variance-covariance matrix of color-vector of points, $\boldsymbol{\mu_c}$ is the mean-vector of color-vector of points.

The likelihood of points being floor points is defined as $P(z)P(\boldsymbol{c})$. Fig. 8 shows the floor detection result. In this case, the points, which were on a square side length of 400 mm in front of the robot, were used as known floor's points.

Green points indicate detected points as the floor. Colored cloth put on the floor, to the right of the robot, as shown in Fig. 8, is not detected as the floor.

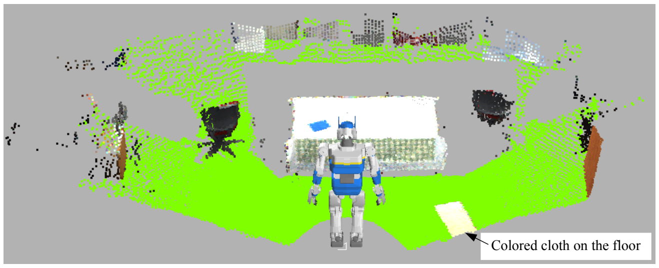


Fig. 8. Floor detection result. Green points are indicated as the floor.

C. Obstacle Detection

A simple labeling method is used for the clustering of point clouds. An elevation map is generated from point clouds excluding points detected as the floor. The elevation map has square grid cells. The cell size is 40 mm on a side. In order to eliminate measuring error, cells that contain a quantity of points lower than a certain threshold are removed. Each labeled point cloud is converted into a prism in order to reduce the cost of collision detection. The prism has the height of highest points in the same labeled cell and the shape of the convex hull of the points projected onto the xy-plane.

Fig. 9 shows labeled prisms. Gathered point clouds are clustered into prisms, and each prisms is treated as a obstacle. Some points outside of the prisms are detected as measurement errors.

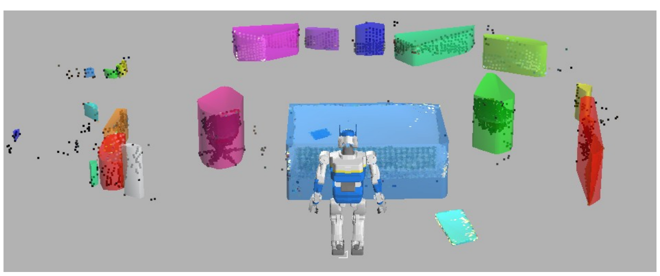


Fig. 9. Labeled prisms generated from point clouds

V. DETECTION OF MOVABLE OBSTACLES BY ACTIVE SENSING

The results from the three types of sensor output are used for movable obstacles detection. A 6-axis force-torque sensor is placed at each wrist, potentio sensors at each each arm joints and the color range sensor are used.

Fig. 10 shows pushing motion. The robot moves its hand horizontally away from itself in order to push the obstacles in front of it. Before detecting obstacles, the robot navigates to a suitable position from which to push the object by using the results of obstacle detection. The height at which the robot extends its hand toward the object varies depending on the height of the detected obstacle.

Fig. 10. Left: Pushing a static obstacle (table), Right: Pushing a movable obstacle (chair)

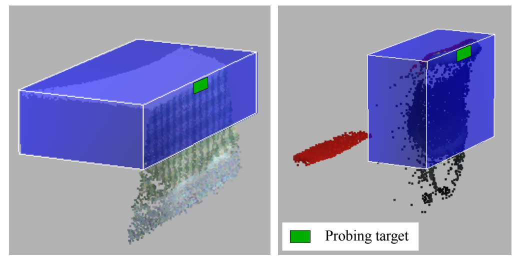


Fig. 11. Detection result of pushing target

Fig. 11 shows detection result of the pushing target. The green square indicates the target for pushing. The target is the top of the vertical surface facing the robot.

Whether or not the robot touches the obstacle is detected by the 6-axis force-torque sensor. When the sensed force is greater than a certain threshold, the robot stops pushing, and the obstacle is determined to be a static obstacle. Fig. 12 shows the results of the sensing force from three cases. The force shown in Fig. 12 is the norm of a 3-axis force. In the case of the robot touching an obstacle and the obstacle moving, a certain amount of force is needed against the friction force. The robot touched the obstacle at about 1.2 s as indicated by the green line in Fig. 12. In the case of robot touching a static obstacle, the robot stops moving its hand. The pushing motion was stopped when the sensed force exceeded 20 N as indicated by the blue line in Fig. 12. In the case of robot not touching anything, sensed values were roughly zero as indicated by the red line in Fig. 12.

3-D point clouds with color were used for estimating obstacle displacement by means of the color-ICP algorithm [12]. Fig. 13 shows the point cloud of the chair at the position before pushing and the position after it was moved. The colors of the chair points were obtained using color-ICP. Fig. 14 shows the results of estimated displacement using color-ICP. Red points shows the point cloud of the chair after it was moved. Green points shows the point cloud of the chair at the original position. Blue points shows estimated result using color-ICP.

In this case, calculated displacement was (189.3, 13.6, -

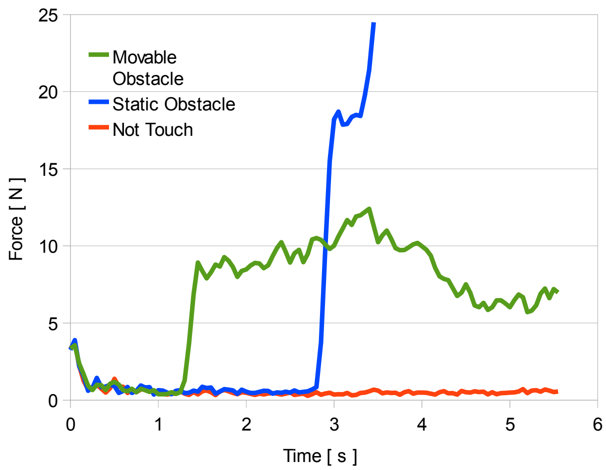


Fig. 12. Force sensing results of each case

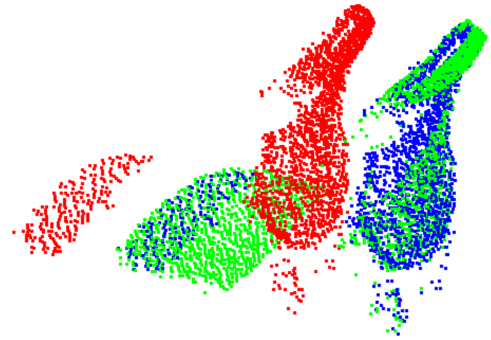


Fig. 14. ICP result, Red: moved points, Green: original points, Blue; moved points translated with ICP result

0.4) [mm] in Cartesian coordinates, and the rotation was a yaw angle of 2.5 deg, a pitch angle of 0.5 deg, and a roll angle of 0.2 deg. At the same time, the distance from the points at which the robot first touched the chair to the points at which the robot stopped pushing was 182.2 mm in the x-direction. The displacement detected by two sensors was 7.1 mm. This was a sufficient enough result for detecting obstacle movement.

In the case of pushing a static obstacle, the displacement estimated by the ICP algorithm was less than 10 mm.

VI. EXPERIMENTAL RESULT

The proposed strategy was demonstrated with humanoid robot HRP-2 in a room with two chairs (movable) and a table (static). The goal was set at the opposite side of the table. The chairs were set as obstacles in the path to the goal. The green cross in Fig. 15 indicates the goal. Fig. 15 shows the experimental result of navigating to the goal in an environment with movable obstacles by pushing obstacles out of the way on the path to the goal. In this experiment, the position of the robot was tracked using HRP-2's built-in walk pattern generator.

As described in Section III, HRP-2 looked around and gathered point clouds from 15 views in 53 s. Obstacles were detected and path planning was executed in 74 s. The nearest obstacle (table) was found. HRP-2 walked to the front of it in 106 s, pushed the table, and, using hand position information and former and latter point cloud information, confirmed that the table was not moved. The table was detected as a static obstacle in 122 s. Next, HRP-2 walked to the back of the nearest chair in 222 s, pushed the chair, and confirmed that the chair was moved. The chair was detected as a movable obstacle in 244 s. Then, HRP-2 moved the chair 800 mm forward. In this experiment, the direction and distance of robot's pushing a obstacle were preliminarily determined. HRP-2 looked around, detected obstacles and found the path to the goal. Finally, HRP-2 reached the goal in a total of 384 s. HRP-2 successfully reached the goal by pushing the chair out of the way on the path to the goal.

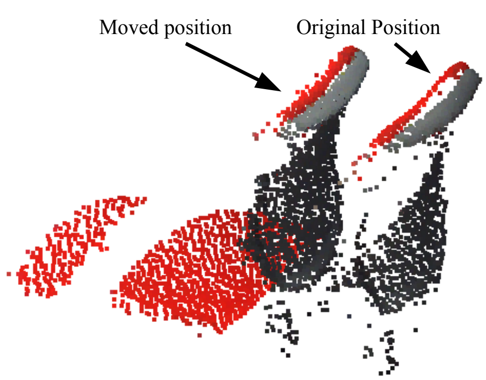


Fig. 13. Original and moved color point cloud of the chair

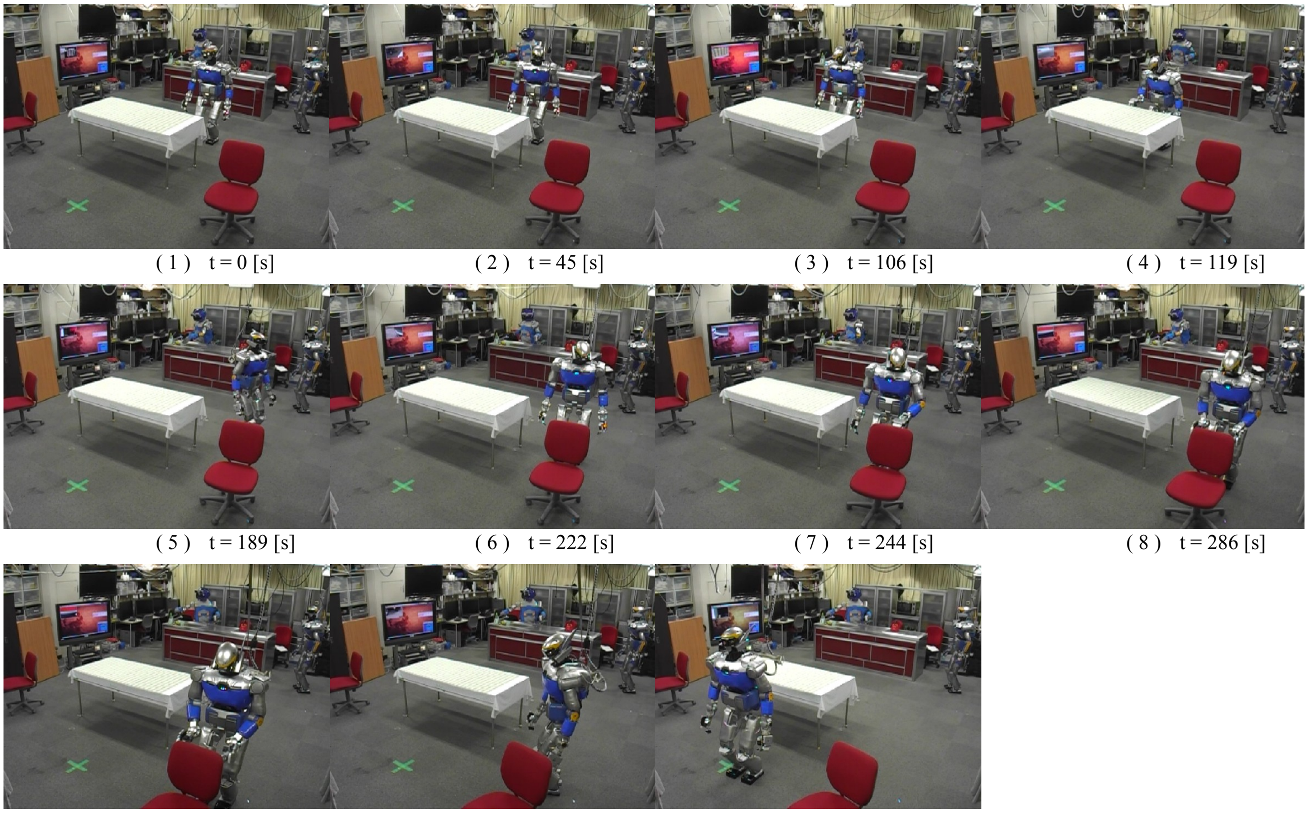
VII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper, we have proposed the action strategy in an environment with movable obstacles without prior knowledge of which objects were movable or which objects were not movable. In this case, the robot needed to detect which obstacles were movable and reconstructs its perception of the environment. We have also shown that the color range sensor is effective for detecting the floor and for clustering obstacles. In order to detect a movable obstacles, active sensing is necessary. Active sensing is achieved by pushing obstacles and then using the combined output of color range sensor, force sensor, and joint angles for accurate detection.

Humanoid robot HRP-2 succeeded in reaching the goal by moving a movable obstacle. We have shown that humanoid robots can execute actions in a complex complex environment with unknown obstacles by means of detecting movable obstacles with active sensing.

There are some problems with detecting objects with no previous knowledge. Segmentation with simple labeling fails when objects are located too close. Also, there might be objects that are not supposed to be touched, such as fragile or valuable object.



(1) t = 0 [s] (2) t = 45 [s] (3) t = 106 [s] (4) t = 119 [s]

(5) t = 189 [s] (6) t = 222 [s] (7) t = 244 [s] (8) t = 286 [s]

(9) t = 296 [s] (10) t = 340 [s] (11) t = 384 [s]

Fig. 15. The result of navigating to the goal indicated by green cross with detecting and moving movable obstacles

B. Future Works

In this paper, planning for obstacle reposition is not considered. Rearrangement planning by mobile robots as described in [13] is needed even for humanoid robots in order for the robot to move obstacles more effectively.

Some problems remain for executing rearrangement by humanoid in a real environment. One supposed problem of rearrangement in a real environment is avoiding the collision between manipulated objects and obstacles in an environment. We think that using collision avoidance planning together with visual perception and a active sensing method as trying to move objcts in order to detect a collision is effective to that problem.

REFERENCES

[1] K. Okada, A. Haneda, H. Nakai, M. Inaba, and H. Inoue, "Environment Manipulation Planner for Humanoid Robots Using Task Graph That Generates ActionSequence," in *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 1174–1179.

[2] M. Stilman, K. Nishiwaki, S. Kagami, and J. J. Kuffner, "Planning and executing navigation among movable obstacles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 820–826.

[3] F. Kanehiro, T. Yoshimi, S. Kajita, M. Morisawa, K. Fujiwara, K. Harada, K. Kaneko, H. Hirukawa, and F. Tomita, "Whole body locomotion planning of humanoid robots based on a 3d grid map," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 1072–1078.

[4] F. Saïdi, O. Stasse, K. Yokoi, and F. Kanehiro, "Online object search with a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1677–1682.

[5] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "3d perception and environment map generation for humanoid robot navigation," *The International Journal of Robotics Research*, vol. 27, no. 10, pp. 1117–1134, 2008.

[6] N. Kwak, K. Yokoi, O. Stasse, and T. Foissotte, "3d grid and particle based slam for a humanoid robot," in *IEEE/RAS International Conference on Humanoid Robotics(Humanoids 09)*, 2009, pp. 62–67.

[7] P. Michel, J. E. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade, "Online environment reconstruction for biped navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 3089–3094.

[8] J. E. Chestnutt, Y. Takaoka, K. Suga, K. Nishiwaki, J. Kuffner, and S. Kagami, "Biped navigation in rough environments using on-board sensing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3543–3548.

[9] Z. C. Marton, R. B. Rusu, D. Jain, U. Klank, and M. Beetz, "Probabilistic categorization of kitchen objects in table settings with a composite sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 4777–4784.

[10] J. V. Miro and G. Dissanayake, "Robotic 3d visual mapping for augmented situational awareness in unstructured environments," in *Proceedings of the International Workshop on Robotics for Risky Interventions and Surveillance of the Environment (RISE 2008)*, 2008, pp. 1–13.

[11] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," in *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, 2001, pp. 145–152.

[12] A. Johnson and S. B. Kang, "Registration and integration of textured 3-d data," in *Proceedings of InternationalConference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97)*, 1997, pp. 234 – 241.

[13] J. Ota, "Rearrangement planning of multiple movable objects by a mobile robot," *Advanced Robotics*, vol. 23, pp. 1–18(18), 2009.