POMDP-based action planning for the recognition of occluded objects with Humanoid robots

Masato Tsuru
Graduate School of Engineering Science, Osaka University, 1-3, Machikaneyama, Osaka, Japan
E-mail: tsuru@hlab.sys.es.osaka-u.ac.jp

Pierre Gergondet
CNRS-JRL, AIST 1-1-1, Umezono, Tsukuba, Japan

Tomohiro Motoda
Graduate School of Engineering Science, Osaka University

Adrien Escande
CNRS-JRL, AIST

Eiichi Yoshida
Intelligent Systems Research Institute, AIST, 1-1-1, Umezono, Tsukuba, Japan

Ixchel G. Ramirez-Alpizar
AIRC, AIST, 2-3-26, Aomi, Koto-ku, Tokyo, Japane

Weiwei Wan
Graduate School of Engineering Science, Osaka University

Kensuke Harada
Graduate School of Engineering Science, Osaka University, AIRC, AIST

Abstract

In this paper, we present a high-layer motion planner which plans humanoid robot actions to search for object models in the robot workspace. To overcome the occlusion problem, our proposed method plans to get different perspectives of the object. POMDP (Partially Observable Markov Decision Process) is used to determine the observation pose of a robot. The planner then builds a comprehensive point cloud of by merging the point clouds gathered from different positions. The point cloud is compared to a 3D model of the object to estimate the pose of the real object.

Keywords: *humanoid robot, POMDP, planning, point cloud, occlusion, 6DoF registration*

1. Introduction

Recently, the necessity for humanoid robots, to assist us in our daily life, has been increasing. In such situations, since robots usually have to detect specific objects by themselves, object search and recognition for humanoids is of great importance for robotics researchers.

We propose a method to improve object recognition and localization capabilities for humanoid robots. Our method uses POMDP (Partially Observable Markov Decision Process) to plan the view pose of a robot to scan

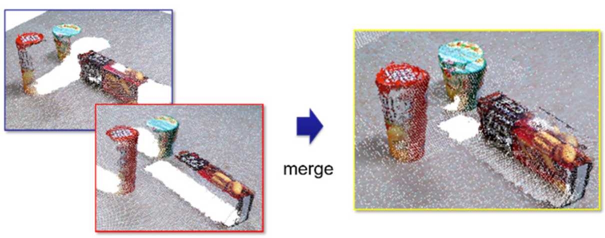

Fig. 1. By merging two point clouds which obtained from different directions, our robot relaxes the problem coming from occlusion.

the object and its environment by using a 3D depth sensor, and create a merged point cloud derived from different points of view. To do it, we set one task, searching an object. We give a target object as a 3D model to the robot, then the robot try to find it from a certain region. The robot obtains 3D point cloud data through its head camera, Xtion, and detects the target object by point cloud matching. However, in the event of occlusion, the robot must recognize the object using incomplete information, which is not always possible. To avoid said problem, our robot moves to another position to get additional point cloud (Fig.1).

It is important to choose a second robot position or viewpoint, which maximizes the information gathered from point cloud data gathering. To minimize the number of times the robot must change position to gather a satisfactory cloud point of the object and its environment. Therefore, we propose a high-layer system to compute the necessary viewpoints and robot positions required to gather a complete point cloud representation of the object. The system analyzes gathered point cloud and estimates the direction which can fill the missing point cloud. Thanks to POMDP controller, our robot doesn't choose a same view direction many time. After changing its position several times and finding the object, the robot tries to grasp the target.

2. Related Works

Saidi et al.[1] searched a target object with autonomous humanoid robot. They also point out the difficulty of object searching task, and solved it with a walking strategy. They used heuristics to plan the next action in unknown environments. We improves the strategy of search with POMDP. Our work also deals with object grasping after a successful recognition is performed.

Kim and Likhachev[2] implemented POMDP with PR2, a mobile robot. They tackled the occlusion problem in 3D point cloud. Their robot detected objects with changing its position and height of torso. The target object was estimated from a small partial point cloud that was hidden by other objects. Finally, they set Breadth-First Search as a baseline and proved the efficiency of POMDP through several experiment. In this research, we apply this method to a humanoid robot HRP-2 that performs stability problems or complex posture problems.

Foissotte et al.[3] proposed a method which selects the best posture to observe a target object, within humanoid stability constraints. In that research, the target location is fixed, and humanoid robot is not allowed to change its position dynamically. The robot can only take one step, change its head position and orientation properly, and fold out its arms to keep stability. The viewpoint is selected by the amount of information. Their method was tested in a simulation environment with OpenGL, and evaluate the observable area with considering occlusions. They proved the efficiency of the planned action for humanoid; however, they didn't generate joint trajectories and consider the dynamical stability in changing posture. In our research, we extend this research and allow the robot to change its standing position and posture. Thanks to it, less difficult postures are generated and it is easy to realize.

3. Proposed Method

3.1. *Overview*

Our system overview is shown in Fig. 2. In this research. We discrete the searching space to simplify the POMDP formulation, and list up all of feasible viewpoints. From that simplification, the robot can search only in a pre-limited area. First, the robot gets point cloud from a randomly selected position in the list. Then, each viewpoint is evaluated by its probability to get new information, depending on the obtained point cloud. The robot executes the highest scored observation action and gets a new point cloud from another direction. The system will merge the new point cloud with the previously obtained one by using the ICP (Iterative Closest Point) algorithm, based on the robot position and orientation. After every observation, the POMDP controller re-calculates every viewpoint score, judging from known point cloud. The robot continues its search until the object pose is fully recognized.

In every step, the robot estimates the object's position and checks that the ICP error. In the case of small errors, the object pose is considered to be known and a grasp of the object is attempted. In grasp planning, robot solves Inverse-Kinematics of right arm, from its shoulder to the hand and checks the stability of the posture. And an approaching motion of the right arm is also generated automatically.

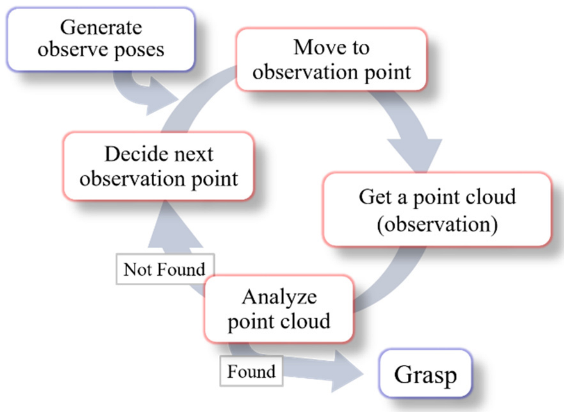

Fig. 2. Basic structure of proposed system

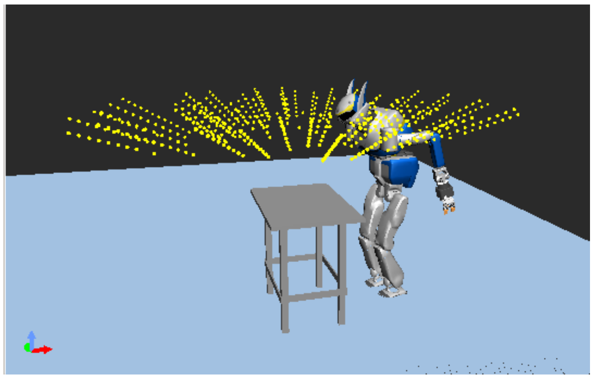

Fig. 3. Generated viewpoint candidates and one example of the postures. Every yellow viewpoint has its observation posture, which are confirmed stability and joint angle limits.

3.2. *Posture Generation*

Because our method needs a lot of posture candidates, we generate a lot of observation postures at first step. As a prime of this research, the searching environment is known roughly. So, we set the target area on the table. Then, we define view directions toward the area. After that, viewpoints, which corresponds to robot head positions and orientations, are created for each 5 cm on the view direction lines. Our planner calculates the whole-body Inverse Kinematics to place the robot head to each viewpoint. If the solution is stable, we add the new one position to a feasible action list. Doing this process, the robot obtained several observation postures like Fig. 3.

3.3. *Point Cloud Processing*

In this research, we use a feature-based point cloud recognition for detecting the target object by merging point clouds. We use CVFH (Clustered Viewpoint Feature Histogram) descriptor[3], which can be used to recognize an object from even partially observed point cloud. We give the 3D model of the object to CVFH before starting experiment. CVFH allows the robot to initially estimate the object location. Then, our algorithm calculates a new robot pose to obtain the missing point cloud data. If there are no hints, even POMDP must decide the next action almost randomly and it is not desirable for us. The robot then proceeds to change its position across its workspace, gathering several point clouds from different directions, finally merging the gathered data into a single large point cloud. Finally, we use the ICP algorithm to fit the object 3D model to the multidirectional point cloud. After acquiring a new set of point clouds from a given viewpoint, our algorithm merges it with the previous sets and uses ICP to try to match the 3D model of the searched object with the merged set of point clouds. ICP returns a score that quantifies the matching between the point cloud and the 3D model. If the score is high enough, the algorithm considers the object pose to be known and uses the position and orientation given by ICP to plan a grasp motion of the object.

3.4. *POMDP*

The POMDP represents the main contribution of this paper. POMDP (Partially Observable Markov Decision Process) is extended and generalized from MDP (Markov Decision Process), and is used in partially observable environments. In POMDP, the position and orientation of an object is represented by a probability distribution. The system updates the probability distribution (it is often called as belief) by every ICP result depending on its accuracy score. If there are some similar objects, the CVFH + ICP usually propose multiple candidates. In such case, the POMDP changes its probability distribution for each candidate. And then, the POMDP evaluates every action by virtual action. In the virtual action process, the POMDP system simulates all of possible observations and evaluates these qualities and quantities. That means, for every feasible viewpoint, by assuming all of the candidates are true and observing them virtually, the system calculates unknown area that the camera may observe. These virtual observation scores are multiplied with the probability distribution, and then, the robot executes observation from the highest scored viewpoint.

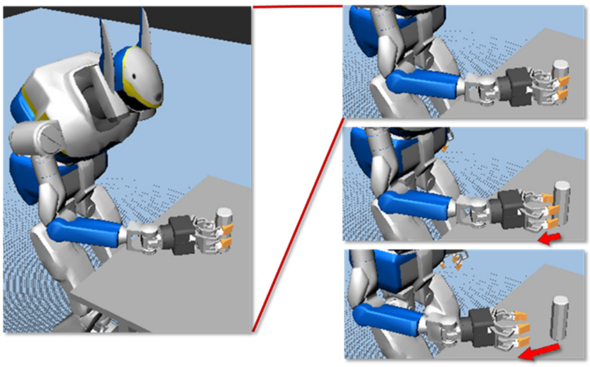

Fig. 4. Generation of grasping motion. The final grasping pose is first generated, and then the approaching trajectory is generated.

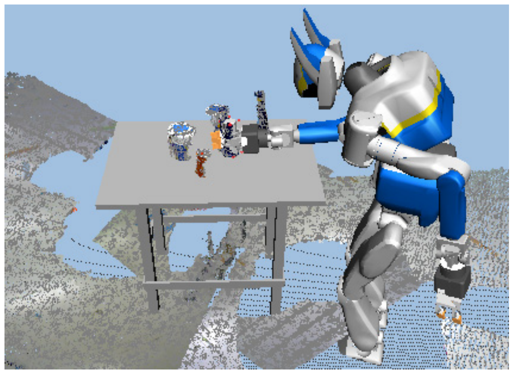

Fig. 5. The robot finally identified the can and grasped it. To disturb the recognition of the can, a flashlight and a case of wet wipe were also placed on the table.

3.5. *Grasp Planning*

After the algorithm detects the object pose, we use grasp planning to make the robot pick-up the object. The 3D model is given for searching, so we use a model-based grasping method. Harada et al.[4] realized a method of general grasping motion generation with approximating the target object by cylinders. They also discussed about an approach direction in grasping with considering open and close of hand. Our system implements said method to realize automatic grasp motion generation. The system approximates the target 3D model by cylinders or boxes. Then it generates grasping hand poses and checks each stability. And an approaching motion is planned automatically by withdrawing the hand, with relying to the robot hand's coordinate system (Fig. 4).

4. Simulation Experiment

Experiment was done with Kinect, and we moved it to viewpoints planned by POMDP. The HRP-2 robot in simulator knows the 3D model of the target object. The target of this experiment is to identify the position of a can in a cluttered environment. Our proposed system generated about 250 observation postures surrounding the table. The first posture was selected randomly since there are no prior information about the object position. After the first observation, the planner decided a second viewpoint for the robot in order to get additional cloud point data from a different perspective. In our simulations, the robot changed its position three times before identifying the object. Finally, the robot grasped the can, as seen in Fig. 5.

5. Conclusions and Future Works

We proposed an observation motion planning framework for a biped humanoid robot based on POMDP to compensate missing cloud point information, in cluttered environments.

However, our algorithm does not take apply the cost to the robot movement into account when planning the viewpoints. In future work we will implement the costs to robot movement to reduce the walking distance of a robot.

6. References

1. F. Saidi, O. Stasse, K. Yokoi and F. Kanehirot, "Online object search with a humanoid robot," 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, 2007, pp. 1677-1682.
2. Sung-Kyun Kim and M. Likhachev, "Planning for grasp selection of partially occluded objects," 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, 2016, pp. 3971-3978.
3. T. Foissotte, O. Stasse, A. Escande and A. Kheddar, "A next-best-view algorithm for autonomous 3D object modeling by a humanoid robot," Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots, Daejeon, 2008, pp. 333-338.
4. A. Aldoma et al., "CAD-model recognition and 6DOF pose estimation using 3D cues," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, 2011, pp. 585-592.
5. K. Harada, K. Nagata, T. Tsuji, N. Yamanobe, A. Nakamura, and Y. Kawai, "Probabilistic approach for object bin picking approximated by cylinders," in Proc. IEEE Int'l Conf. Robotic and Automation (ICRA), May 2013, pp. 3742–3747