Proc. 13th Int. Conf. on
Intelligent Autonomous Systems,
Padova, Italy, July 2014.

Human-Robot Collaborative Remote Object Search

Jun Miura, Shin Kadekawa, Kota Chikaarashi, and Junichi Sugiyama

Department of Computer Science and Engineering,
Toyohashi University of Technology

Abstract. Object search is one of the typical tasks for remotely-controlled service robots. Although object recognition technologies have been well developed, an efficient search strategy (or viewpoint planning method) is still an issue. This paper describes a new approach to human-robot collaborative remote object search. An analogy for our approach is *ride on shoulders*; a user controls a fish-eye camera on a remote robot to change views and search for a target object, independently of the robot. Combined with a certain level of automatic search capability of the robot, this collaboration can realize an efficient target object search. We developed an experimental system to show the feasibility of the approach.

Keywords: Human-robot collaboration, object search, observation planning.

1 Introduction

Demands for remotely-controlled mobile robots are increasing in many application areas such as disaster response and human support. One of the important tasks for such robots is object search. To find an object, a robot continuously changes its position and examines various parts of the environment. An object search task is thus roughly composed of viewpoint planning and object recognition. Although technologies for object recognition have been well developed with recent high-performance sensors and the use of informative visual features, viewpoint planning is still a challenging problem.

Exploration planning [1, 2] is a viewpoint planning for making a description of the whole workspace. Efficient space coverage is often the goal to achieve in this planning. Concerning object search, Tsotsos and his group have been developing a general, statistical framework of visual object search [3, 4]. Saidi et al. [5] takes a similar approach in object search by a humanoid. Aydemir et al. [6] utilize high-level knowledge on spatial relations between objects to select low-level search strategies. We have also developed algorithms for efficient mapping and object search in unknown environments (called *environment information summarization*). Masuzawa and Miura [7, 8] formulated this problem as a combination of greedy exploration ordering of unknown sub-regions and a statistical optimization of viewpoint planning for object verification. Boussard and Miura [9] formulated the same problem as an MDP and presented an efficient solution using LRTDP [10]. These works are for improving performance and efficiency of automatic object search.

A human operator sometimes controls or supports the robotic exploration and/or object search in a tele-operation context, where interface design is an important issue. Various design approaches are possible depending on how largely the robot controller and the operator contribute to actual robot actions. When the operator mainly controls the motion of the robot, an informative display of the remote scene is required. Fong et al. [11] proposed a sensor fusion display for vehicle tele-operation which can provide visual depth cues by displaying data from a heterogeneous set of range sensors. Suzuki [12] developed a vision system combining views from a usual camera and an omnidirectional one to provide a more informative view of a remote scene. Saitoh et al. [13] proposed a 2D-3D integrated interface using an omnidirectional camera and a 3D range sensor. Shiroma et al. [14] showed a bird's-eye view could provide a better display for a mobile robot tele-operation than a panoramic or a conventional camera.

The idea of safeguard teleoperation (e.g., [15]) is often used in which the operator gives a higher level command and the robot realizes it with keeping safety. Shared autonomy is a concept that a human and a robot collaborate by an even-contribution manner. Sawaragi et al. [16] deals with an ecological interface design for shared autonomy for a tele-operation of a mobile robot. The interface provides sufficient information for evoking the operator's natural response. These works do not suppose a high-level autonomy of the robot systems.

This paper describes a new type of human-robot collaboration in remote object search. We suppose the robot has an enough level of autonomy for achieving the task. Since the human's ability of scene recognition is usually better than those of robots, however, a human operator also observes a remote scene and helps the robot by giving advice on the target object location. An analogy of our approach is *ride on shoulders*; a boy on his father's shoulders searches for a target object and tells its location to the father, while the father is also searching for it. The boy can also understand which direction the father is focusing and/or moving. We realize this relationship by putting a fish-eye camera on a humanoid robot and make the camera's focus of attention be remotely controllable.

The rest of the paper is organized as follows. Section 2 explains the hardware and software configuration of the system. Section 3 describes an automatic object search strategy that the robot takes. Section 4 describes a camera interface for the operator and human-robot interaction in the collaborative object search. Section 5 summarizes the paper and discusses future work.

2 Overview of the System

Fig. 1 shows the hardware and software configuration of the system. The robot we use is HIRO, an upper-body humanoid by Kawada, put on an omnidirectional mobile base. It has three types of sensors. An RGB-D camera (Kinect) on the head is used for detecting tables and object candidates. Two CCD cameras at both wrists are used for recognizing objects based on the textures. Three laser

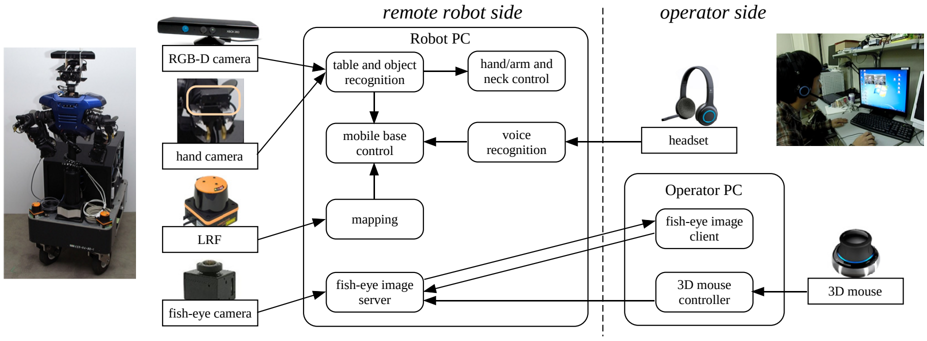


Fig. 1. System configuration.

range finders (LRFs) (UHG-08LX by Hokuyo) on the mobile base are used for SLAM (simultaneous localization and mapping).

The robot is equipped with a fish-eye camera (NM33-UVCT by Opto Inc.) for providing the operator the image of the remote scene. The operator can extract a perspective image of any direction using a 3D mouse (SpaceNavigator by 3DConnecxion Inc.) so that he/she can search anywhere at the remote site for a target object. The interface also provides the operator the state of the robot, that is, where it is and where it is searching for the target object. A headset is used for the operator to give voice commands to the robot.

The software is composed of multiple functional modules, shown by rounded rectangles in Fig. 1. Each module is realized as an *RT component* in the *RT-middleware* environment [17], which supports a modularized software development. We use an implementation of RT-middleware by AIST, Japan [18].

3 Automatic Object Search

3.1 Algorithm of automatic object search

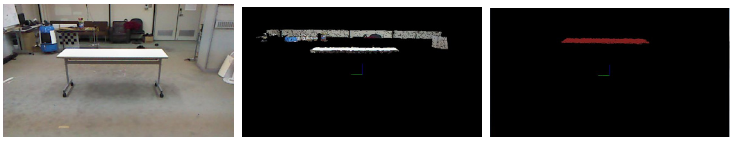
The task of the robot is to fetch a specific object in a room. Object candidates are assumed on a table. The robot thus starts from finding tables in the room, and then moves on to candidate detection and target recognition for fetching. We deal with only rectangular parallelepipeds as objects, and extract and store the visual features (i.e., color histogram and SIFT descriptors [19], explained below) of the target object in advance. The sizes of objects are also known. The algorithm of automatic object search is summarized in Fig. 2.

3.2 Object detection and recognition routines

Table detection Tables are detected from point cloud data taken by the Kinect using PCL (Point Cloud Library) [20]. Assuming that the heights of tables are

Step 1: Detect all tables in the room using the RGB-D camera by turning the neck.
Step 2: Choose the nearest and unexplored table and approach it.
  (a) If it exists, goto Step 3.
  (b) If not, go back to the initial position and report failure.
Step 3: Search for target object candidates using color histogram.
  (a) If found, goto Step 4.
  (b) If not, goto Step 2 (search another table).
Step 4: Approach candidates and recognize them using a SIFT-based recognition/pose estimation.
  (a) If the target object is recognized, grasp it and go back to the initial position.
  (b) If not, goto Step 2 (search another table).

Fig. 2. Algorithm for automatic object search.



(a) Test scene. (b) Height filtering. (c) Detected table.

Fig. 3. Table detection.

between 70 [cm] and 90 [cm], planar segments with vertical normals in that height range are detected using a RANSAC-based algorithm. Fig. 3 shows a table detection result.

Candidate detection on a table Once a table to approach is determined, the robot moves to the position with a certain distance to the table. The point cloud data is again analyzed using PCL to extract data corresponding to objects on the estimated table plane. The extracted data are clustered into objects, each of which is characterized by a Hue histogram. A normalized cross-correlation (NCC) is then calculated between the model and the data histogram to judge if an object is a candidate. Fig. 4 illustrates the process of candidate detection.

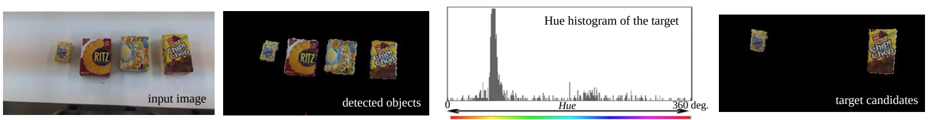


Fig. 4. Candidate detection. The rightmost object is the target.

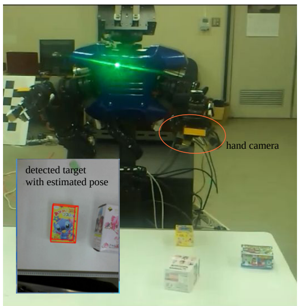


Fig. 5. Object recognition and pose estimation.

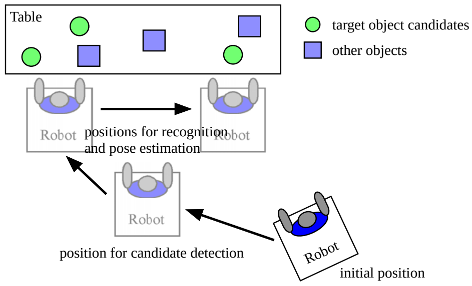


Fig. 6. Movement of the robot for detection and recognition. Two target candidates on the left are grouped and recognized from a single position.

Object recognition using SIFT Each target object candidate is verified by using a hand camera. SIFT features are extracted in each candidate object region and matched with those in the model. If the number of matched features is above a threshold, the target object is considered verified. The pose of the object can be calculated from the pairs of the 2D (image) and the 3D (model) feature positions. We use the *cv::solvePnP* function in the OpenCV library [21] for pose estimation. Fig. 5 illustrates the object recognition and pose estimation procedure.

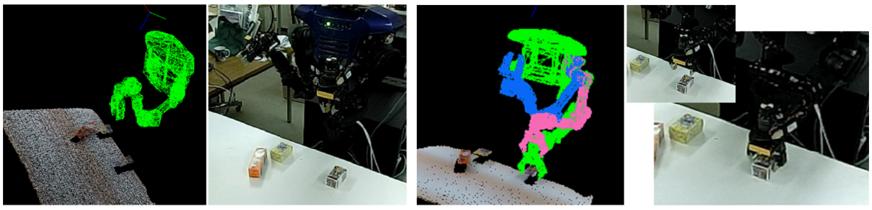
3.3 Mobile base control

The omnidirectional mobile base uses four actuated casters with a differential drive mechanism [22]. The mobile base is also equipped with three LRFs, which are used for an ICP-based ego-motion estimation provided by Mobile Robot Programming Toolkit (MRPT) [23].

In detecting candidates on a table, the robot moves to a position which has a certain relative distance (about 1 $[m]$) from the table so that the whole tabletop can be observed. In recognizing the target object, the robot approaches the table to obtain an enough number of SIFT features using the hand camera. The positions for recognition are determined considering the placements of target candidates on the table; nearer candidates are grouped to reduce the number of movements in front of the table. The robot keeps a right-angle position to the table in both observations. Fig. 6 illustrates a typical movement of the robot.

3.4 Arm and hand control

The hands of the robot is used for placing a camera above a candidate object for recognition as well as pick and place operations. In the case of recognition, the upper surface of each candidate is observed . The camera pose is defined in advance with respect to the coordinate system attached to the corresponding surface. For pick and place, we use a predefined set of grasping and approaching poses, also defined with respect to the surface coordinates. We implemented



(a) Simulation/action for observation. (b) Simulation/action for picking.

Fig. 7. Hand motion generation.

Fig. 8. Experimental scene. There are three tables and the target object is on the leftmost table with respect to the robot.

a point cloud-based collision check procedure, which can check both robot-to-object collisions and self-collisions. Fig. 7 shows examples of hand movements with collision checks.

3.5 Automatic object search experiment

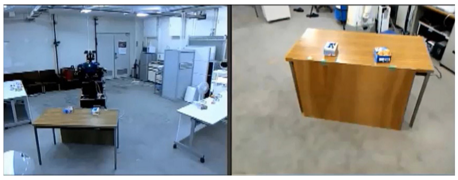
We performed automatic object search experiments. Fig. 8 shows the experimental scene. The robot examines tables in the right-to-left order because the rightmost table is the nearest to the initial position. Since the target object is on the leftmost table from the robot, it at least searches every table for candidate detection.

Fig. 9 shows snapshots of an automatic object search. The search process is as follows. After detecting three tables in the room (Step 1), the robot first moved to the rightmost one to find a candidate (Step 2). Since the candidate was not the target (Step 3), the robot moved to the center table where no candidates were found (Step 4). It then move to the leftmost one to find a candidate (Step 5). Since the candidate was recognized as the target, the robot picked it up (Step 6) and brought it to the initial position (Step 7).

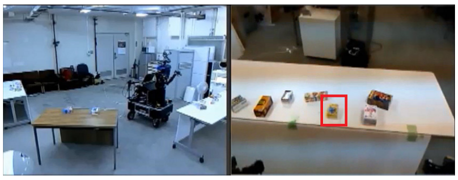
(a) Step 1: Detect three tables.

(b) Step 2: Move to the first table and find a candidate.

(c) Step 3: Fail to recognize (no target object).

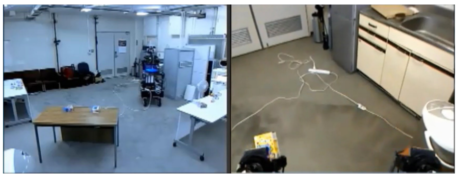


(d) Step 4: Move to the second table and find no candidates.



(e) Step 5: Move to the last table and find a candidate.

(f) Step 6: Succeed to recognize and grasp the target.



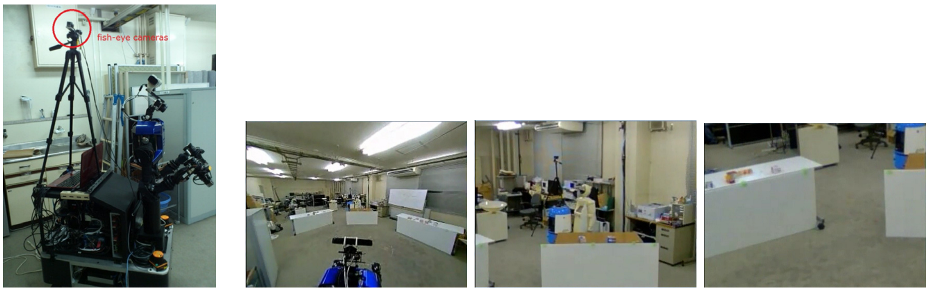
(g) Step 7: Go back to the initial position.

Fig. 9. Snapshots of an experiment of automatic object search. The views from the robot are shown on the right at each step.

4 Collaborative Object Search

A child on his father mentioned in Sec. 1 is the analogy for the collaborative object search in this work. The child does not walk by himself but looks around to independently search for a target object, and once he finds it, he tells his father of the location of the target. This is a kind of interruption to father's action, and the father takes the advice and moves to it.

To provide an independent view to the operator, a fish-eye camera is put on the mobile base and is made controllable to the operator. The operator changes the view for searching for the target and gives verbal advice to the robot.



(a) Setting of the fish-eye camera. (b) View the remote scene and the robot. (c) Zoom up the center table in (b). (d) Zoom up the left table in (b).

Fig. 10. Fish eye camera and examples views.

4.1 Fish-eye camera-based interface

The fish-eye camera is set at the rear of the robot as shown in Fig. 10(a). This setting enables the operator to view not only the remote scene but also the robot's state (see Fig. 10(b)). The camera has a function of extracting an arbitrary part of the fish-eye image and converting it to a perspective image. The operator can thus control the pan/tilt/zoom of a *virtual camera* using the 3D mouse. Fig. 10(b)-(d) show examples of images taken from the same robot position.

4.2 Voice command-based instruction

The operator uses voices to instruct the robot to take a better action than the robot's current one. Since the task (i.e., target object search) is simple, instructions used are also simple enough to be easily used by the operator. Table 1 summarizes the voice instructions and the corresponding robot actions to be invoked.

Table 1. Voice instructions

Voice instruction	Robot action
"Hiro" (name of robot)	Stop action
"Left table"	Look at the table on the left
"More to the left"	Look at the table next to the left one
"Right table"	Look at the table on the right
"More to the right"	Look at the table next to the right one
"Come back"	Come back to the initial position
"Search there"	Move to the table in front of the robot for search

4.3 Collaborative search experiments

Fig. 11 shows snapshots of a collaborative object search when the target exists in the scene. After detecting three tables in the room (Step 1), the robot moved to the rightmost one. While this movement, the operator found the target on the leftmost table and said "Hiro" to stop the robot (Step 2). The operator then said "left table" and the robot looked at the center table (Step 3). Since the target is on the table at the left side of the current one[1], the operator further said "more to the left" and the robot looked at the leftmost table (Step 4). Then the operator said "search there" and the robot moved to that table (Step 5). As the robot found a candidate, it approached the table (Step 6). Since the candidate was recognized as the target, the robot picked it up (Step 7) and fetched it to the initial position (Step 8).

Fig. 12 shows snapshots of a collaborative object search when the target does not in the scene. After detecting three tables, the robot moved to the rightmost one and further approached it because a candidate was found (Step 1). Recognition using a hand camera failed (Step 2). While the robot was moving to the next table, the operator noticed that there were no target objects in the room and said "Hiro" to stop the robot (Step 3). The robot stopped searching and came back to the initial position (Step 4).

4.4 Comparison of automatic and collaborative search

In collaborative search, the operator observes the remote scene from a distant place through a fish-eye camera and a display and, once he finds the target, he interrupts the robot and instructs the place to search (or orders to stop search). Appropriate operator's advice keeps the robot from examining tables without targets, thereby reducing the total cost of search.

We compared the automatic and the collaborative object search in terms of the total search time, the number of tables examined for candidates, and that of candidates examined for target recognition. Table 2 summarizes the comparison results. The collaborative search is more efficient than automatic one by the timely advice from the operator to the robot.

5 Conclusions and Future Work

This paper describes a novel type of human-robot collaboration for object search. Analogy to the child-on-the-shoulder case, the operator examines the remote scene through a camera on the robot, by which he can observe the state of the robot as well as the scene, and gives timely advice to the robot. We have implemented an experimental system and shown the collaborative object search is more efficient than automatic one in several preliminary experiments.

[1] Note that the operator was able to see which table the robot is looking at through a fish-eye camera

(a) Step 1: Detect three tables.

(b) Step 2: While the robot moves to the first table, the operator finds the target objects and orders it to stop.

(c) Step 3: The operator says “left table” and the robot looks at the center table.

(d) Step 4: The operator says “more to the left” and the robot shift the focus on the leftmost one.

(e) Step 5: The operator says “search there” and the robot move to the table.

(f) Step 6: Detect a candidate and approach to the table.

(g) Step 7: Recognize the target and pick it up.

(h) Step 8: Go back to the initial position.

Fig. 11. Snapshots of an experiment of collaborative object search. The views from the robot are shown on the right at each step.

The current system deals with only object on tables. Extending the space to search to various locations (e.g., in the shelf) is desirable. This will require increasing voice instructions so that various places can be specified. Communications between the operator and the robot could be more interactive, since the current communication is unidirectional (from operator to robot). The robot may want to actively ask about probable place of a target object or ask the operator to examine some place where the robot thinks objects probably exists. Such kinds of interactions, which may be observed in actual child-father interactions are expected to make the collaborative search much more efficient.

(a) Step 1: The robot approaches to the first table to recognize the candidate.

(b) Step 2: Recognition fails.

(c) Step 3: While the robot moves to the second table, the operator notices that no target exist and says "Hiro" to stop the robot.

(d) Step 4: The robot stops searching and comes back to the initial position.

Fig. 12. Snapshots of another experiment of collaborative object search when the target object does not exist in the scene. The views from the robot are shown on the right at each step.

Table 2. Comparison of automatic and collaborative search.

Case 1: the target object exists in the room.

method	search time	# of table examined	# of candidates examined
automatic	5 min. 20 sec.	3	2
collaborative	3 min. 33 sec.	1	1

Case 2: the target object does not exist in the room.

method	search time	# of table examined	# of candidates examined
automatic	4 min. 47 sec.	3	2
collaborative	3 min. 36 sec.	1	1

References

1. A.A. Makarenko, S.B. Williams, F. Bourgault, and H.F. Durrant-Whyte. An Experiment in Integrated Exploration. In *Proceedings of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 534–539, 2002.
2. R. Martinez-Cantin, N. de Freitas, R. Brochu, J. Castellanos, and A. Doucet. A Bayesian Exploration-Exploitation Approach for Optimal Online Sensing and Planning with a Visually Guided Mobile Robot. *Autonomous Robots*, Vol. 27, pp. 93–103, 2009.
3. Y. Ye and J.K. Tsotsos. Sensor Planning for 3D Object Search. *Computer Vision and Image Understanding*, Vol. 73, No. 2, pp. 145–168, 1999.
4. K. Shubina and J.K. Tsotsos. Visual Search for an Object in a 3D Environment Uing a Mobile Robot. *Computer Vision and Image Understanding*, Vol. 114, pp. 535–547, 2010.

5. F. Saidi, O. Stasse, K. Yokoi, and F. Kanehiro. Online Object Search with a Humanoid Robot. In *Proceedings of 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1677–1682, 2007.
6. A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt. Search in the Real World: Active Visual Object Search Based on Spatial Relations. In *Proceedings of 2011 IEEE Int. Conf. on Robotics and Automation*, pp. 2818–2824, 2011.
7. H. Masuzawa and J. Miura. Observation Planning for Efficient Environment Information Summarization. In *Proceedings of 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 5794–5780, 2009.
8. H. Masuzawa and J. Miura. Observation Planning for Environment Information Summarization with Deadlines. In *Proceedings of 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 30–36, 2010.
9. M. Boussard and J. Miura. Observation planning for object search by a mobile robot with uncertain recognition. In *Proceedings of the 12th Int. Conf. on Intelligent Autonomous Systems*, 2012. F3B.5 (CD-ROM).
10. B. Bonet and H. Geffner. Labeled RTDP: Improving the Convergence of Real-Time Dynamic Programming. In Enrico Giunchiglia, Nicola Muscettola, and Dana S. Nau, editors, *Proceedings 13th Int. Conf. on Automated Planning and Scheduling (ICAPS-2003)*, pp. 12–31. AAAI, 2003.
11. T. Fong, C. Thorpe, and C. Baur. Advanced Interfaces for Vehicle Teleoperation: Collaborative Control, Sensor Fusion Displays, and Remote Driving Tools. *Autonomous Robots*, Vol. 11, pp. 77–85, 2001.
12. S. Suzuki. A Vision System for Remote Control of Mobile Robot to Enlarge Field of View in Horizontal and Vertical. In *Proceedings of 2011 IEEE Int. Conf. on Robotics and Biomimetics*, pp. 8–13, 2011.
13. K. Saitoh, T. Machida, K. Kiyokawa, and H. Takemura. A 2D-3D Integrated Interface for Mobile Robot Control using Omnidirectional Images and 3D Geometric Models. In *Proceedings of 2006 IEEE/ACM Int. Symp. on Mixed and Augmented Reality*, pp. 173–176, 2006.
14. N. Shiroma, N. Sato, Y. Chiu, and F. Matsuno. Study on Effective Camera Images for Mobile Robot Teleoperation. In *Proceedings of 13th IEEE Int. Workshop on Robot and Human Interactive Communication*, pp. 107–112, 2004.
15. T. Fong, C. Thorpe, and C. Baur. A Safeguarded Teleoperation Controller. In *Proceedings of 2001 IEEE Int. Conf. on Advanced Robotics*, 2001.
16. T. Sawaragi, T. Shiose, and G. Akashi. Foundations for Designing an Ecological Interface for Mobile Robot Teleoperation. *Robotics and Autonomous Systems*, Vol. 31, pp. 193–207, 2000.
17. N. Ando, T. Suehiro, and T. Kotoku. A Software Platform for Component Based RT System Development: OpenRTM-aist. In *Proceedings of the 1st Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR '08)*, pp. 87–98, 2008.
18. OpenRTM. `http://openrtm.org/openrtm/en/`.
19. D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.
20. Point Cloud Library. `http://pointclouds.org/`.
21. OpenCV. `http://opencv.org/`.
22. Y. Ueno, T. Ohno, K. Terashima, H. Kitagawa, K. Funato, and K. Kakihara. Novel Differential Drive Steering System with Energy Saving and Normal Tire using Spur Gear for Omni-directional Mobile Robot. In *Proceedings of the 2010 IEEE Int. Conf. on Robotics and Automation*, pp. 3763–3768, 2010.
23. MRPT. `http://www.mrpt.org/`.