Drag-and-Drop Interface for Registration-Free Object Delivery

Kentaro Ishii, Yoshiki Takeoka, Masahiko Inami, and Takeo Igarashi

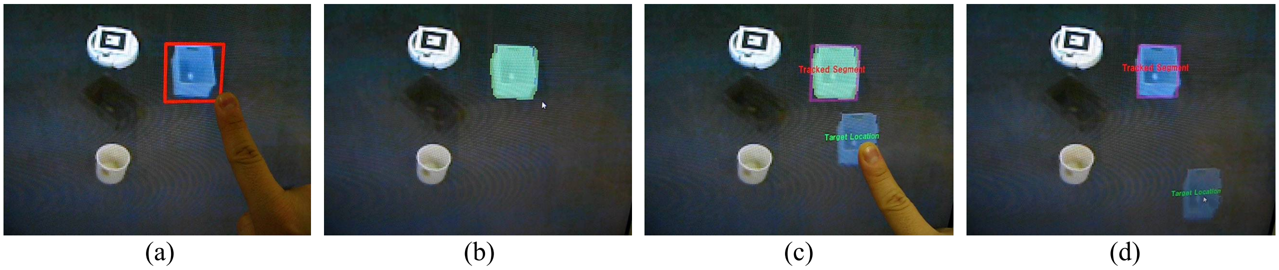

Fig. 1. Drag-and-Drop Operation by the User—(a) The user selects the target object. (b) The system highlights the selected object. (c) The user drags the object. (d) The user drops it at the target location. The robot then autonomously pushes the target object to the target location.

***Abstract*—We propose a simple drag-and-drop interface on a touch screen to give instructions to an object-delivery robot. The user uses a finger to drag a target object and drop it to a target location on the screen that shows an image of the floor provided by a ceiling camera. The system then autonomously executes a delivery task without continuous control by the user. We employ user-guided computer vision methods to identify and track an object in a scene without prior registration. We conducted a user study, which showed that the participants completed separate tasks more successfully when using our method than when using a remote controller.**

I. Introduction

TYPICAL robot-control methods can be divided into two approaches: (1) direct manual control and (2) semantic-level instructions. A typical example of the first approach is joystick control wherein the user continuously gives low-level instructions to the robot such as "move forward" and "spin left." This approach is reasonably reliable, but it requires continuous attention of the user. A typical example of the second approach is instruction by means of a natural language wherein the user gives a semantic instruction and the robot executes the task independently, without continuous control by the user. This approach seems ideal, but it suffers from various registration and recognition problems. For example, when the user says "move box no.1 to location no.1," the robot needs to know beforehand which is box no.1 and where is location no.1. Alternatively, the user might say "move this box to the corner of the room" but the robot could confuse "this box" with another box and "the corner" with a different corner.

We pursue an intermediate approach that combines the strengths of these two approaches. The user gives an instruction that has an appropriate amount of abstraction, and the system executes that task with a reasonable degree of autonomy. We apply this semi-autonomous approach to an object-delivery task. Using a finger on a touch screen with a camera view of the environment, the user intuitively "grabs" an image of a target object, "drags" it to a target location, and "drops" it there (Fig. 1). Subsequently, the robot performs the task autonomously, using histogram matching to track the location of the target object. Our method eliminates the need for continuous control, as is necessary in low-level control methods. Moreover, our method is more reliable than the one that uses semantic-level instructions because the user explicitly specifies the target object and target location in the camera view, and assists in the system recognition.

A contribution is the use of user-guided computer vision methods to achieve registration-free object identification and tracking for the purpose of the object delivery. Typical approaches for object identification and tracking attach a physical tag to each object [1]–[3] or use pattern recognition to identify a pre-registered object. However, these approaches cannot handle unknown objects. Our system requires the user to manually specify the object in the camera view by rubber banding, and the system uses this information for object identification. This strategy is also utilized for error recovery. Although our object tracking by image tracking is not perfect and the system can lose the target object, the user can re-specify the target object from the camera view on the screen. The effectiveness of a user-guided approach has been shown in GrabCut [4] and Interactive graph cuts [5]. They solved difficult segmentation problems with a few user instructions. We apply the same strategy in the camera view to identify an object to be delivered.

In our method, the camera can be in arbitrary view. We also use image processing techniques to perform the

All authors are with Japan Science and Technology Agency, ERATO, IGARASHI Design Interface Project, Tokyo, Japan (e-mail: kenta@designinterface.jp).

Y. Takeoka is also with Graduate School of Interdisciplinary Information Studies, The University of Tokyo, Tokyo, Japan (e-mail: takeoka@designinterface.jp).

M. Inami is also with Graduate School of Media Design, Keio University, Yokohama, Japan (e-mail: inami@designinterface.jp).

T. Igarashi is also with Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, Japan (e-mail: takeo@acm.org).

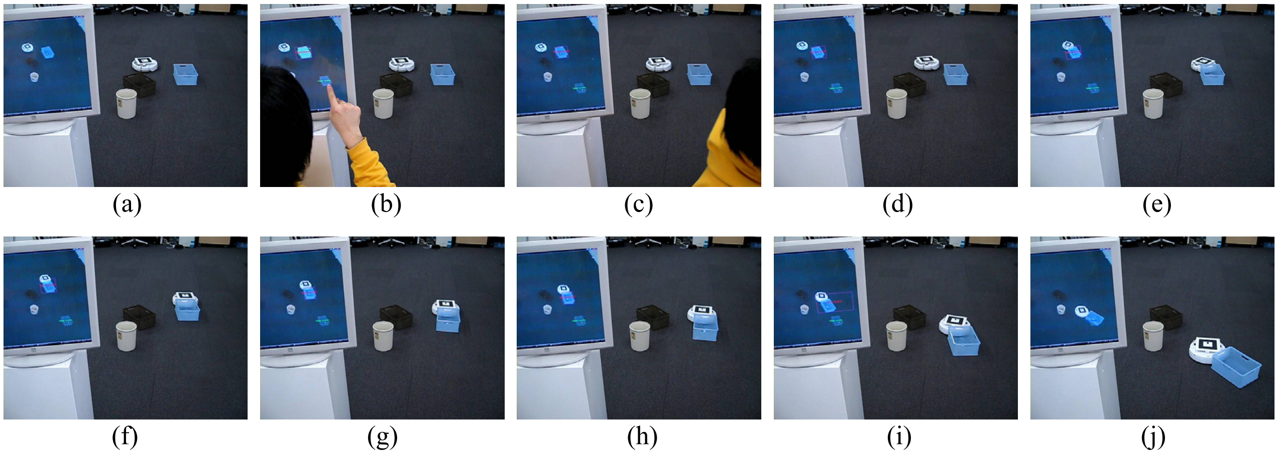

Fig. 2. Execution Sequence—(a) Initial configuration. (b) The user performs drag-and-drop operation. (c)–(j) The robot executes a delivery task while the user is absent.

automatic calibration of the system. The system automatically constructs the floor plane on the basis of the posture of a visual marker. The system uses this floor plane for controlling the robot. Because our method does not require any prior registration, with this automatic calibration feature, the setup process is very easy in any environment.

We built a prototype system using an iRobot Create [6], a ceiling camera, a host computer, and a touch screen. Then, we conducted a user study to verify the effectiveness of the proposed method. We asked the participants to complete object delivery tasks while calculating sum of two-digit numbers as distraction tasks. We compared our method with the one that used a simple remote controller, and the results showed that the participants could perform the distraction tasks better using our method. Moreover, the participants preferred our method, as revealed in a post-study questionnaire.

The advantages of our system over previous systems are summarized as follows:

- The user can command an object-delivery robot by simple and intuitive drag-and-drop interaction, and the robot can perform a delivery task with a reasonable degree of autonomy.
- We employ a semi-autonomous approach for specifying an unregistered object and for performing error recovery.
- Our system can be easily set up because the system does not require any prior registration and can automatically calibrate the working environment of the robot.

The remainder of this paper is organized as follows. Section II describes work related to our robot-control method. Section III describes important features of our method. Section IV presents a detailed implementation of our prototype. Section V describes a user study for the proposed interface. Section VI discusses limitations and future work, and Section VII concludes the paper with a brief summary.

II. Related Work

Autonomous navigation methods utilizing simple interactions have been proposed. A pointing interface on a PDA screen proposed by Lundberg *et al.* [7] and a free-hand stroke interface on a tablet PC proposed by Skubic *et al.* [8] allowed the robot to navigate to the target location without continuous control by the user. Similarly, we apply a simple drag-and-drop method to an object-delivery task.

Rouanet *et al.* demonstrated a user-guided object identification method for a user to teach new words to a robot [9]. Using a handheld device, the user can view what the robot sees through a video image from the camera on the robot. The user informs the robot to focus on an object by encircling the object on the robot camera image. The robot then acquires a more focused template of the target object. They showed that the user-guided approach by encircling the focused object enhanced the object identification process required for teaching new words. This method was useful for object registration, while in our study we focus on registration-free task execution.

We share the objective of controlling a robot more implicitly as found in the work by Zhao *et al.* [10]. The user does not directly give a command to the robot, rather issues a command based on an object. In their prototype implementation, paper cards identified by visual markers provided instructions to robots. These cards were placed at arbitrary locations so that the user could assign a robot task, such as “deliver this object there.” Our system is similar in that it provides a way to specify arbitrary locations to which an object needs to be moved. Unlike their system, our system can deliver unregistered objects.

III. System Features

This section describes important features of our system. First, we describe the user interface and autonomous execution following user commands. Then, we explain the user assistance of our semi-automatic approach. Finally, we

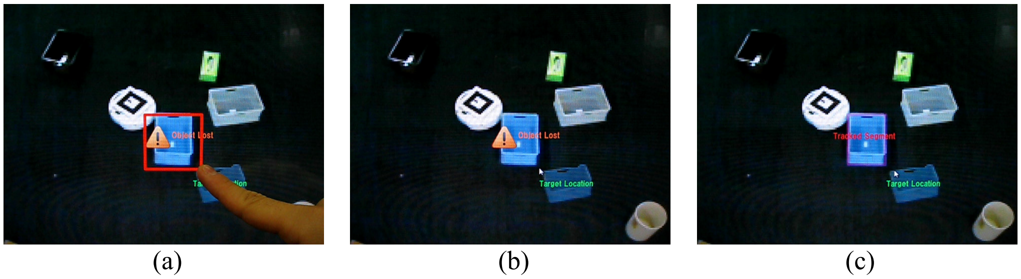

Fig. 4. Recovery from an Object Lost—The system provides a warning when it loses the target object. The user then reselects the object and the system resumes the task.

explain the automatic calibration of the system that enables easy setup.

A. Drag-and-Drop Interface for Object Delivery

We apply the established drag-and-drop method used in modern graphical user interfaces (GUIs) to the real-world task of object delivery. The user uses a finger to select a target object on a touch screen, "drags" it to a target location, and "drops" it to initiate the delivery operation (Fig. 1). When the user specifies the region enclosing a target object by rubber banding, the system recognizes the enclosed image segments as the target object. The image segments are also highlighted to confirm that the system recognizes the object before the user drags it to the target location (Fig. 1(b)). As the object is dragged, a translucent copy of the selected image segments is displayed along the dragging path (Fig. 1(c)). When the user drops the object, the robot starts the delivery task. The translucent copy remains at the target location while the robot executes delivery. This visual feedback provides a useful preview of the result (Fig. 1(d)). When the object is selected and dropped, the interface also provides audio feedback to the user to indicate the system state.

Fig. 2 shows the execution sequence of the system. After receiving the user instructions, the robot begins to deliver the object by pushing it according to the instruction. The user can perform other activities while the robot performs the task in the background. When the robot delivers the object, the interface makes a sound and displays a message notifying completion of the task.

B. User-Guided Object Identification and Tracking

To deliver an object, the robot needs to know location of the object. We use a graph-based image-segmentation technique [11] to extract the target object from a camera image. However, the image segmentation cannot be perfect because the system cannot estimate which segments comprise the target object (Fig. 3), whereas this task is easy for the user. In our system, the user informs the system about the object that the user has in mind. When the user provides an input to select an object (Fig. 1(a)), the system interprets that the enclosed segments comprise the target object.

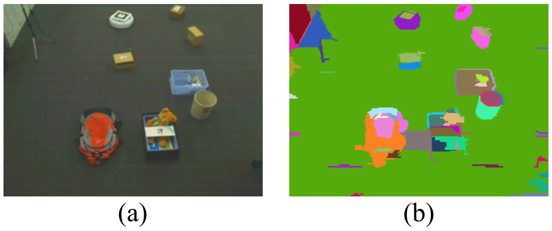

Fig. 3. Segmentation Result—(a) Original image. (b) Segmented image.

In addition, object tracking by image matching cannot be perfect because lighting conditions may change when the robot moves the target object. To solve this problem, the user needs to explicitly reselect a lost target object. In case the tracking module loses track of the object, the system makes a sound and displays a message asking the user to reselect the object by rubber banding. When the user reselects the object, the system renews the reference image and resumes delivering the object to the target location (Fig. 4).

C. Automatic Calibration

Our system uses a visual marker-detection technique by NyARToolkit [12], which is a Java implementation of ARToolKit [13]. NyARToolkit can compute the 3D posture of a marker. Using this information, the system constructs a model of the floor plane and calculates the relative positions of the robot and objects in a floor-plane coordinate system (Fig. 5). This floor-plane model functions as the calibration between the camera image and the 2D coordinates of the floor where the robot moves and works. In our current implementation, system calibration is performed the first time the marker appears in the camera image. All the user has to do is simply run the program and place the robot within the camera view.

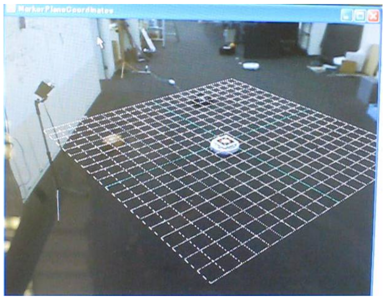

Fig. 5. Floor-Plane Construction by the 3D Posture of the Marker on the Robot

IV. Implementation Details

A. Hardware Configuration

Our system consists of a touch screen, a camera, a robot, and a computer (Fig. 6). The user of the system inputs instructions, and receives visual and audio feedback by means of the touch screen with a speaker. We use an off-the-shelf web camera and attach it to the ceiling. The camera covers an area of 6 m × 4 m. The object-delivery robot, called iRobot Create [6], is a developer's version of the popular Roomba home-vacuuming robot. A visual marker is attached to the robot to enable location and orientation tracking and floor-plane construction, as described in Section III-C (Fig. 7). The robot comes with a serial interface for receiving control commands. We use a Bluetooth serial adapter to control the robot wirelessly. The touch screen, camera, and robot are connected to a computer that processes the captured image and user inputs, drives the robot, and generates user feedback.

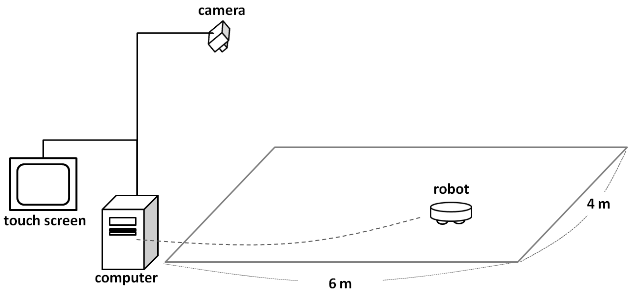

Fig. 6. Hardware Configuration

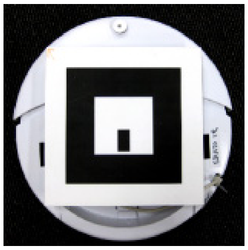

Fig. 7. iRobot Create with Vision-based Marker

With our prototype, the average completion time for 5 m delivery is 20 seconds, and the position error of delivered object is less than 50 cm. Note that the completion time varies among trials because of the possibility of the object getting deviated from the direction of pushing, and the accuracy depends on the distance of the camera from the floor.

B. Object Tracking

This system tracks a target object using histogram-based image matching with an active search method [14]. The system computes a color histogram of pixels in the selected segments, and searches for a region that has the most similar color histogram. Fig. 8 shows color histograms of the selected objects. Different objects have differently colored histograms.

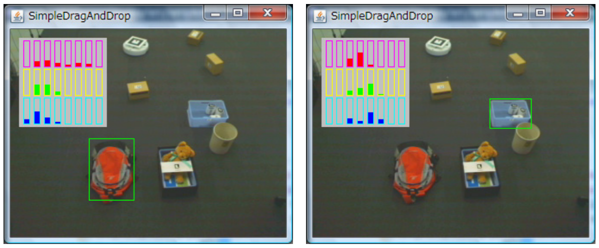

Fig. 8. Color Histograms—Different objects have differently colored histograms.

Fig. 9 shows the result of object tracking and indicates that this system tracks the object even if it is rotated. The user drags the blue box and drops it as described in Section III-A. This image segment is used as the reference image for object matching.

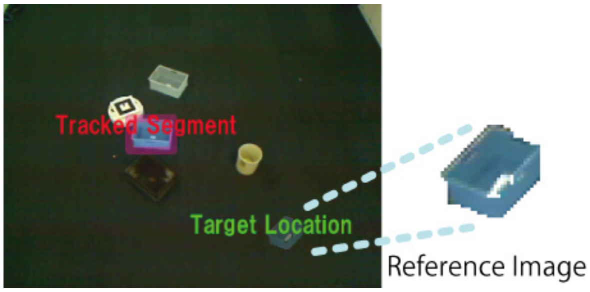

Fig. 9. Tracking Result—The user views the left image on the display. The real object appears at the left, above the center of the image, while a translucent proxy of the object appears at the bottom right of the image. The translucent proxy is also used for the reference image for image tracking.

C. Robot Tracking and Control

Robot tracking is performed by recognizing the visual marker on the robot from the camera view, as described in Section III-C. Using robot position and orientation information and object-position information, the robot-control module controls the robot by sending low-level control commands to the robot such as "move forward" and "spin left." To complete an atomic pushing operation, the robot (1) moves behind the target object, (2) adjusts its orientation toward the target location, and (3) pushes the target object to the target location. By tracking the positions of the robot and the object, the system is able to instruct the robot to reposition itself behind the object in case the object slips off in front of the robot during a pushing operation.

V. User Study

We conducted a user study to show the effectiveness of the semi-autonomous control method. We chose to compare with remote controller method as it and our proposed method shared the feature of easy setup. The aim of this comparison was to evaluate the extent to which participants could perform personal tasks while the robot performs the delivery task. To compare the two interfaces, each participant controlled the robot to deliver an object using each robot control system, while simultaneously calculating sum of two-digit numbers. After this comparison, each participant was asked to answer a questionnaire. Each participant was also interviewed about both the systems to obtain qualitative evaluations.

A. Study Setup

The study was conducted on ten participants: two were females, and eight were males, and their ages ranged from 21 to 24 years. Two of them had engineering-related majors, and none of them had ever used the proposed system before.

First, the participants were explained about the usage of both control systems, and followed by a practice session to provide a better understanding of the systems. Then, a two-minute test session started, wherein the participants repeatedly controlled the robot to deliver an object using each interface, while simultaneously providing oral answers to a series of the additions of two-digit numbers. We instructed the participants to deliver the object to one of four target locations forming a square, and start the next delivery in a clockwise direction immediately after the previous delivery was completed. A single addition was displayed on another display near the participant at a time. The additions were changed each time the participant answered, irrespective of whether or not the answer was correct. The session was recorded, and the visuals were reviewed to count both the total number of answers and the number of correct answers in the two-minute session. Fig. 10 shows one of the scenes of the user study. We alternated the order of use of the control systems by the participants to counterbalance the results.

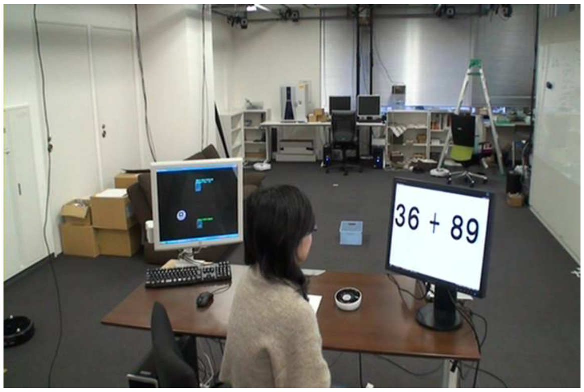

Fig. 10. Comparison Study Setup

After using both control systems, the participants filled out a questionnaire. The participants used a five-point scale to rate four topics: “understandability,” “learnability,” “intuitiveness,” and “incorporation.” We defined “incorporation” as “how well the system can be incorporated into other activities.” Futher, each participant was interviewed for approximately 45 minutes about both the systems they had used.

B. Results and Observations

Table I shows the average number of answers for the distracting additions, as well as the number of correct answers. In both cases, the participants scored higher when using the proposed system. We performed the Wilcoxon signed-rank test, and found significant differences (p < 0.01) between the two methods in both scores.

TABLE I
AVERAGE NUMBER OF ANSWERS

	Answers	Correct Answers
Proposed System	30	28
Remote Controller	19	18
	(p < 0.01)	(p < 0.01)

Fig. 11 shows the results of the questionnaire. Compared to the remote-control system, the proposed system received higher score in every question. In particular, “incorporation” had a large difference in scoring, thus indicating that the proposed method is suitable when the users need to perform other tasks. We performed the Wilcoxon signed-rank test, and found significant differences (p < 0.05 in “learnability” and “intuitiveness;” and p < 0.01 in “incorporation”) between the two methods in all questions except “understandability.”

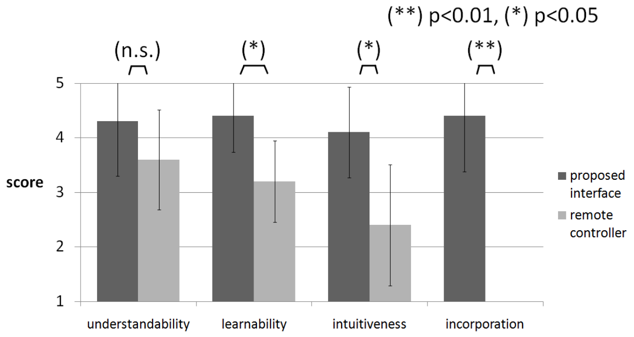

Fig. 11. Questionnaire Result

Although time and accuracy to complete each task may provide valuable information, we could not obtain meaningful results, mainly because we did not provide any direction about delivery accuracy to the participants. As a result, precision of positioning the object in four desired locations in the controller method varied from participant to participant. Some participants spent more time and achieved more accuracy with the controller method compared to our proposed method, while other participants spent less time and achieved less accuracy.

In the post-study interview, the participants provided many positive feedbacks for the proposed system. They primarily liked the simplicity of the proposed interface and the fact that they could perform their own tasks after issuing instructions to the robot.

Most participants reported that they liked the real-time view on being asked about the view provided by the ceiling camera. Only one participant experienced difficulty because of the inability to judge the exact distance when using the camera view.

There were also suggestions for improvement. Most of the participants wanted to know the estimated time to complete a delivery. Although we understand the reason for this request, it is difficult to estimate a completion time because the round robot pushes the target object in current delivery system, and the robot can swerve off the object. One possible solution for estimating the completion time is to use a more functional robot, such as one with a grasping mechanism.

One participant requested to be informed repeatedly within a certain time period about the progress of delivery, and preferred audio feedback to a visual message that requires looking at the touch screen.

Two participants complained that the object orientation changed from its initial position, as shown in the translucent preview. Because the current implementation uses histogram matching, which does not preserve orientation information, it is difficult to maintain the stability of object orientation.

A participant, who was not an engineering student, wanted to specify the robot path as well as select a target object and location. We did not anticipate that a user unaware of the details of our system would request for such a feature.

VI. Limitations and Future Work

We used an image-based technique to track objects in our prototype system. However, we will consider using physical tracking tags or prior registration in our method to make it easier to complete the delivery task without losing the object. Although one of the advantages of our method is its ability to handle unknown objects, our method will be more reliable in terms of object tracking for objects with such physical tracking tags or prior registration.

The robot in our system cannot work outside the view of the camera. We plan to add cameras to expand the field of view in the future. If an additional camera is placed to partially provide an overlapping view and detect the robot's marker simultaneously, the system can obtain a homography of the two camera views as the robot travels from the view of one camera to that of another. This does not harm the easy setup feature of our system. The robot can automatically travel around the environment, and the first time two or more cameras sees the marker simultaneously, the system can calibrate coordinates among the cameras.

The robot used in our prototype is rudimentary, and can move only at floor level and push an object along the floor. However, the same drag-and-drop interface can be applied to advanced robots. For example, by using a robot that can grasp an object, our drag-and-drop interface could be used to place an object into a container. We hope to apply our interface in such robots to complete more complex and interesting tasks in the future.

VII. Conclusion

This paper proposed a simple drag-and-drop touch screen interface that instructs a robot to deliver an object. Our method uses image-processing techniques to provide registration-free and automatic calibration of the system. The user assists the system operation by providing information regarding the image segments of the target object. The system autonomously drives the robot after a delivery task is assigned to the robot. Users can perform their personal activities while the robot executes the task in the background. Our user study showed that unlike the use of a remote controller, the use of our semi-autonomous system does not interfere with a user's personal activities.

Acknowledgment

We thank Manfred Lau for his valuable suggestions, and Sorahiko Nukatani for his help to make the video.

References

[1] Ubisense. Precise Real-Time Location Systems and GIS Consulting by Ubisense [Online]. Available: http://www.ubisense.net

[2] Y. Nishida, H. Aizawa, T. Hori, N. H. Hoffman, T. Kanade, and M. Kakikura, "3D Ultrasonic Tagging System for Observing Human Activity," in Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems, vol. 1, pp. 785–791.

[3] R. Raskar, P. Beardsley, J. van Baar, Y. Wang, P. Dietz, J. Lee, D. Leigh, and T. Willwacher, "RFIG Lamps: Interacting with a Self-Describing World via Photosensing Wireless Tags and Projectors," in Proc. 31st Int. Conf. Computer Graphics and Interactive Techniques, 2004, pp. 406–415.

[4] C. Rother, V. Kolmogorov, and A. Blake. ""GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts," in Proc. 31st Int. Conf. Computer Graphics and Interactive Techniques, 2004, pp. 309–314.

[5] Y. Y. Boykov and M.-P. Jolly. "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," in Proc. 8th Int. Conf. on Computer Vision, 2001, vol. 1, pp. 105–112.

[6] iRobot. iRobot Corporation: Home Page [Online]. Available: http://www.irobot.com

[7] C. Lundberg, C. Barck-Holst, J. Folkeson, and H. I. Christensen, "PDA interface for a field robot," in Proc. 2003 IEEE/RSJ Int. Conf. Intelligent Robots and Systems, vol. 3, pp. 2882–2888.

[8] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz, "Using a hand-drawn sketch to control a team of robots," Autonomous Robots, 2007, vol. 22, no. 4, pp. 399–410.

[9] P. Rouanet, P.-Y. Oudeyer, and D. Filliat, "An integrated system for teaching new visually grounded words to a robot for non-expert users using a mobile device," in Proc. of 9th IEEE- RAS Int. Conf. Humanoid Robots, 2009, pp. 391–398.

[10] S. Zhao, K. Nakamura, K. Ishii, and T. Igarashi, "Magic Cards: A Paper Tag Interface for Implicit Robot Control," in Proc. of 27th International Conf. Human Factors in Computing Systems, 2009, pp. 173–182.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," Int. J. Computer Vision, 2004, vol. 59, no. 2, pp. 167–181.

[12] NyARToolkit. NyARToolkit for Java.en - NyARToolkit [Online]. Available: http://nyatla.jp/nyartoolkit/wiki/index.php?NyARToolkit%20for%20Java.en

[13] ARToolKit. ARToolKit Home Page [Online]. Available: http://www.hitl.washington.edu/artoolkit/

[14] H. Murase and V. V. Vinod, "Fast visual search using focused color matching - active search," System and Computers in Japan, 2000, vol. 31, no. 9, pp. 81–88.