Designing Laser Gesture Interface for Robot Control

Kentaro Ishii[1], Shengdong Zhao[2,1], Masahiko Inami[3,1], Takeo Igarashi[4,1], and Michita Imai[5]

[1]Japan Science and Technology Agency, ERATO, IGARASHI Design Interface Project, Frontier Koishikawa Bldg. 7F, Koishikawa 1-28-1, Bunkyo-ku, Tokyo, Japan
[2]Department of Computer Science, National University of Singapore
[3]Graguate School of Media Design, Keio University
[4]Graduate School of Information Science and Technology, The University of Tokyo
[5]Faculty of Science and Technology, Keio University kenta@designinterface.jp, zhaosd@comp.nus.edu.sg, inami@designinterface.jp, takeo@acm.org, michita@ayu.ics.keio.ac.jp

Abstract. A laser pointer can be a powerful tool for robot control. However, in the past, their use in the field of robotics has been limited to simple target designation, without exploring their potential as versatile input devices. This paper proposes to create a laser pointer-based user interface for giving various instructions to a robot by applying stroke gesture recognition to the laser's trajectory. Through this interface, the user can draw stroke gestures using a laser pointer to specify target objects and commands for the robot to execute accordingly. This system, which includes lasso and dwelling gestures for object selection, stroke gestures for robot operation, and push-button commands for movement cancellation, has been refined from its prototype form through several user-study evaluations. Our results suggest that laser pointers can be effective not only for target designation but also for specifying command and target location for a robot to perform.

Keywords: laser pointer, stroke gesture, human-robot interaction, user interface

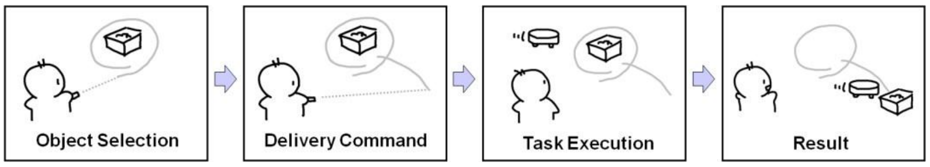

Fig. 1. Robot Control using Laser Gestures

1 Introduction

Robot technology has now advanced to the extent that many companies produce robots that not only can carry out tasks in industrial factories, but also can support everyday human activities. Today's commercial robots can already perform some of the tasks required in the home, such as vacuuming rooms [1].

These robots, however, are still limited to pre-programmed functionalities, and many important issues must be addressed to introduce more capable robots into home environments. One such issue is the question of how to give unique instructions to specify the types of robot operation, target objects, and target location. For example, when a user wants a robot to carry laundry to the veranda, the user has to give an instruction that the operation is delivery, the target object is laundry, and the target location is veranda.

A laser pointer provides a powerful alternative as a user interface for robot control. It is cheap, lightweight, points accurately, and is reliably recognized. Kemp *et al.* and Suzuki *et al.* showed that a laser pointer is useful for providing target designation for a picking-up robot [2] or a button-pushing robot [3]. However, target designation on its own is far from sufficient for a human-robot interface. The user also needs to be able to specify what to do and where to do with the target (e.g., *deliver* the laundry *to veranda*, *collect* the toys *in corner of the room*).

We extended the idea presented by Kemp *et al.* [2] and Suzuki *et al.* [3] to use a laser pointer not only for target designation but also for specifying command and target location. To achieve this, we applied gesture recognition to the laser trajectory. Fig. 1 illustrates our approach. After the user makes a gesture using the laser pointer to indicate the target object and desired command, the robot performs the task specified by the gesture. With our user interface, the user can command a robot to deliver, collect, and trash objects. The user can also instruct the robot to cancel, pause, undo, and redo a task.

This paper reports on our experience in designing an effective laser gesture interface. Using a vision-based tracking system by ceiling-mounted cameras and a remote-controlled iRobot Create, we developed a prototype robot system. We iteratively refined laser gesture sets by repeatedly testing them with first-time users. The final design of our system includes lasso and dwelling gestures for object selection, stroke gestures for robot operation, and push-button commands for movement cancellation.

The remainder of this paper is organized as follows. Section 2 describes previous work on robot control, and the use of laser pointers, and stroke gesture interfaces. Section 3 introduces the prototype system we built to test the laser gesture interface. Section 4 describes the initial design of our laser gesture interface. Section 5 describes the details of our system modification process that involves four rounds of user testing and design revision. Section 6 discusses possibility and limitations of our user interface, and Section 7 concludes the paper with a brief summary.

2 Related Work

2.1 Robot Control

Previous work on interfaces for robot control can be roughly divided into two approaches: (1) direct manual control and (2) as-autonomous-as-possible. A typical example of the former is joystick control. The user continuously gives low-level instructions to the robot, such as to move forward and turn left. This approach might be necessary for performing complicated tasks in a remote location (e.g., extraterrestrial robots). The other approach is the as-autonomous-as-possible method. Typically, these methods include the use of speech- and hand-gesture recognition. These methods require only a very abstract instruction from the user (e.g., “clean here”) and the system automatically performs the task. While this might sound ideal, these high-level instructions are inherently ambiguous and often lack the robustness required for practical use.

There have been attempts to combine the two methods described above, creating systems that are more abstract than a direct form of control, but more concrete and specific than using speech and hand gestures. Perzanowski *et al.* provided multimodal interface in which the user can specify the target location by tapping a map on a PDA screen [4]. Lundberg *et al.* also proposed a PDA interface for a robot, where the user can specify the area for the robot to explore [5]. In this study, we also follow the approach to combine user control and robot autonomy.

2.2 Laser Pointer

Laser pointers can be used in several situations. Their most common use is as a pointing device to aid communication between a speaker and an audience during a presentation. Since a laser pointer can precisely point to a target location on a screen from a distance, the audience knows at which feature to pay attention. Taking advantage of this characteristic of a laser pointer, Yamazaki *et al.* developed a laserpointing device that could be operated remotely called a GestureLaser [6]. Using a mirror with an actuator and a camera, a remote operator can give instructions while checking the actual pointing position on a remote camera image.

A laser pointer can also be used as a computer input device. For ordinary computer application, the user can use a laser pointer instead of a mouse to perform actions such as point-and-click or drag-and-drop [7], [8]. For robotic application, Kemp *et al.* used a laser pointer to tell a robot which object to pick up [2]. Using a laser pointer and a mobile robot, the user can pick up an object that is some distance away because the laser points over long distances. Their prototype system enables the user to “fetch” an object while remaining seated. Suzuki *et al.* also used a laser pointer to tell a robot which button to push [3]. With a flashlight, the user can specify an approximate direction for the robot to turn its face. After that, the user can designate the target button using the laser pointer, and the robot moves toward the button accordingly. Contrary to these two works, our interface uses a laser pointer not only for target designation but also for specifying command and target location.

2.3 Stroke Gesture Interface

Stroke gestures are used to input commands for various computer applications. The most significant advantage of using stroke gestures to input commands is that the user can specify several kinds of commands using just a simple pointing device. The idea of marking menus with a pointing device [9] is now widely used as a common method to input commands in applications such as web browsers.

Sakamoto *et al.* applied stroke gesture recognition to robot instructions [10]. They used four cameras capturing top-down images of a room, which the user can monitor during instruction. Skubic *et al.* developed robotic interface by free hand stroke [11]. This interface displays sensory information and robot status such as driving or not. Igarashi *et al.* also proposed a similar technique to navigate a virtual avatar [12]. The user can navigate the avatar by drawing its path to walk through.

Our interface supports command specification based on stroke gesture recognition as well as marking menus [9] and Sketch and Run [10]. However, it is done by using a laser pointer.

3 Prototype System

In our prototype system, the user draws gestures on the floor using a laser pointer and the system executes the specified robotic task accordingly. This implementation supports three tasks: delivery, collecting, and trashing. All tasks are performed by pushing objects. The delivery task simply moves the selected objects from one place to another. The collecting task clusters the selected objects into a specified location. The trashing task moves the selected objects to a predefined trash box location. Fig. 2 shows the working environment for the prototype system.

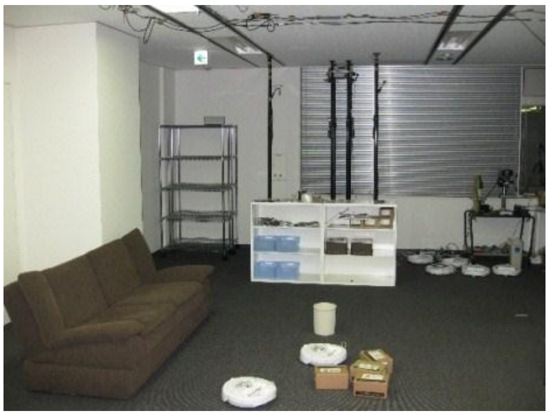
Fig. 2. Prototype Environment

Fig. 3. Laser Pointer

Fig. 4. iRobot Care

3.1 Hardware Configuration

The frontend system consists of a laser pointer held by the user and a robot on the floor that execute the tasks. The laser pointer is an off-the-shelf product with two extra buttons for moving forward/backward through a slideshow presentation (Fig. 3). The task execution robot is a developer's version of the popular home vacuuming robot (Roomba), called iRobot Create [1] (Fig. 4). It comes with a serial interface for receiving control commands. We use a Bluetooth-serial adapter to control the robot wirelessly.

The backend system consists of three cameras, a speaker, a projector, and two computers (Fig. 5). The three cameras are fixed on the ceiling and cover 4m x 3m viewing area. The speaker and projector are used to provide feedback to the user. The two computers are connected via a TCP/IP network. One computer is used for robot and object tracking and the other computer is used to recognize laser gestures, drive the robot, and generate audio-visual feedback to the user.

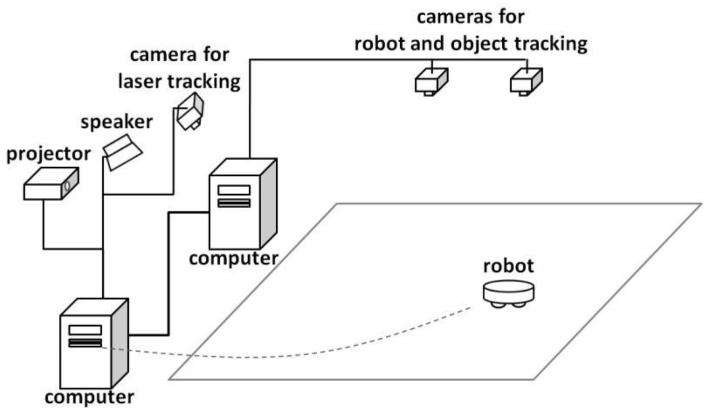

Fig. 5. Hardware Configuration

3.2 Laser Tracking

We use a simple web camera with a narrow band-pass filter for laser tracking. The narrow band-pass filter passes only light with a wavelength of the laser, shutting out all other wavelengths. Since the narrow band-pass filter passes only laser light, laser tracking is simply realized by tracking the most significant pixel on the image provided by the camera.

3.3 Robot and Object Tracking

Our system uses vision-based tag identification for robot and object tracking. 3 x 3 (9bit) 2D markers are used for visual tag tracking. An example of this design is illustrated in Fig. 6 left. In order to make the tags small, we use a proprietary tag tracking system because we are only concerned about the 2D robot motion on the floor, contrary to existing marker-based technology that can capture 3D postures, such as CyberCode [13], ARTag [14], and ARToolKit [15]. Fig. 6 right shows the results of our robot and object tracking. Each of the two cameras calculates the ID, position, and orientation of the existing tags. The coordinate system of each camera is manually calibrated with each other.

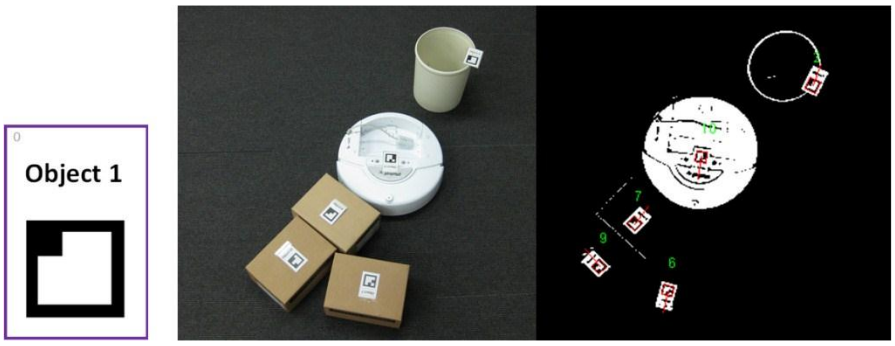

Fig. 6. Vision-based ID Tag and Tracking Results

It is important to note that visual tags may not be ideal for practical deployment because it is tedious initially to put visual tags on all objects in an environment. We envision that this will be replaced in the future by another registration-free technique such as image-based segmentation using overview camera, or probing the object by sensors on the robot.

3.4 Robot Control

Using the information collected by the tag tracking system, the robot control module controls the robot based on the results from the gesture recognition module. Since all tasks are comprised of pushing operations, the robot control module creates a path procedure and sends low-level control commands such as “move forward” and “spin left.” To complete an atomic pushing operation, the robot first moves behind the target object, adjusts its orientation toward the target location, and then moves to the target location. The system also instructs the robot to avoid other objects during its movement, tracking the robot and object positions during operation. In case the object slips off from the robot during pushing, the system orders the robot to retry from moving behind the object.

3.5 Audio-Visual Feedback to the User

A sound speaker is placed at the corner of the prototype environment for the user to receive sound feedback. Sound feedback works to understand gesture recognition results. In addition, we use poles to place a projector near the ceiling, projecting toward the floor (Fig. 7). The projector gives visual feedback on the laser pointer trajectory while the user is drawing a stroke. Visual feedback enables the user to understand laser tracking result (Fig. 8).

Fig. 7. Projector on Ceiling

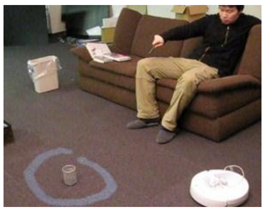
Fig. 8. Visual Feedback

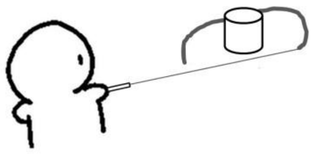
Fig. 9. Object Selection on Laser Trajectory

4 Initial Laser Gesture Design

4.1 Instruction Procedure

The following steps summarize the robotic instruction procedure:

1. Select Object
2. Input Gesture Command

Since all tasks require target objects, the user needs to specify the target objects before inputting a command.

Object selection in initial design is performed by a lasso gesture (Fig. 9). We choose lasso gesture because the user can select single or multiple objects in the same manner.

4.2 Command Gestures

We designed command methods as single-stroke gestures. Fig. 10 shows stroke gestures representing delivery, collecting, and trashing, as well as the gesture to cancel a selection. We consulted the existing works on stroke gesture interfaces [16], [17], [18] to design our initial set of stroke gestures.

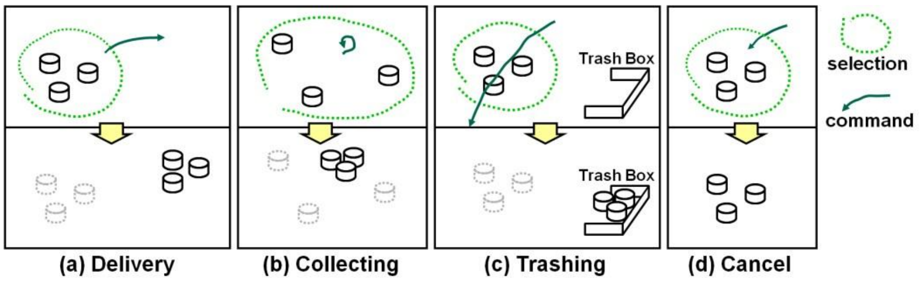

Fig. 10. Command Gestures

The delivery command tells the robot to move the selected objects to a target location. When the user draws a line from inside the selection lasso to another area, the system interprets it as a delivery command and the robot moves the objects to the end of the line (Fig. 10(a)). The collecting command tells the robot to collect the selected objects at the target location inside the selection lasso. When the user draws a circle inside the selection lasso, the system interprets it as a collecting command and the robot collects the objects in the center of the circle (Fig. 10(b)). The trashing command tells the robot to throw selected objects into a predefined trash box. When the user draws a stroke across the selection lasso, the system interprets it as a trashing command, and the robot moves the objects to the trash box (Fig. 10(c)). We also provide a cancel command. When the user draws a line across the selection lasso from outside to inside, the system interprets it as a cancellation of the selection and returns to the initial state (Fig. 10(d)).

4.3 Task Control Gestures

We also designed task control gestures so that users could adjust the execution of tasks after having given some commands. Using task control gestures, users can cancel, pause, undo, and redo a task (Fig. 11).

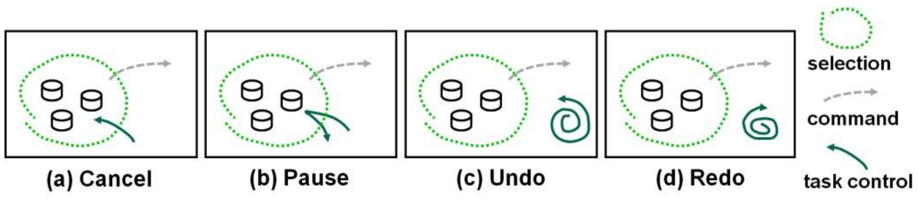

Fig. 11. Task Control Gestures

Task cancellation uses the same gesture as selection cancellation (Fig. 11(a)). When a task is cancelled, the robot stops performing the task immediately and leaves the objects as they are. Pause is used to suspend a robot's activity temporarily (e.g., to stop robot noise when answering a phone call). If the stroke enters the selection lasso and returns to the outside, the system pauses in the current task (Fig. 11(b)). After the pause, the robot resumes its task when the same gesture is recognized. Since our application only changes the position of real-world objects, the system realizes undo by restoring the object position from task history. If the stroke forms a double counter-clockwise circle, the system starts to undo the previous task (Fig. 11(c)). If no task history is stored, the system does not recognize the undo command. Redo is an inverse function of undo. The user can re-order the robot to redo the task that it has been instructed to undo. If the stroke forms a double clockwise circle, the system starts to redo the previously undone task (Fig. 11(d)). If there is no undone task, the system does not recognize the redo command.

4.4 Design of Audio-Visual Feedback

To ensure that the user understands how the system recognizes each gesture, it is important that users receive appropriate feedback. In addition to drawing the laser trajectory as described in section 3.5, the system emits sound and changes the color of the trajectory when the system recognizes a stroke gesture. The system emits a different sound for selecting objects, and for each command and task-control gestures. The trajectory color is white before recognition, and yellow after recognition.

5 Iterative Refinement based on User Study

5.1 Overview

The interface introduced in the previous section was tested and refined by first-time users. We invited 10 participants (9 females, 1 male; 3 single, 7 married, 2 with kids; age range: 27 to 39 years old, mean: 31) who identified as having a non-science educational background, as we assumed that home users would be the target audience of our system. We had 3 participants in the first round, 3 participants in the second round, 2 participants in third round, and 2 participants in the fourth round. Although we had a small number of participants in each round, we intended to evaluate the interface qualitatively rather than quantitatively.

In each round, participants could only refer to a user's manual for instructions on how to use the system, and could not ask questions to the experimenter. We took this policy because in the real world, the home user can only refer to a user's manual of a product for information on the system.

Each participant was given the following tasks to accomplish in ten minutes:

- Move a trash box to the sofa where the participant is sitting
- Cluster four toy boxes, which are placed around the room.
- Move a cardboard to trashing area.

It should be noted that we did not specify the destination for the cluster in the second task. This may have caused various completion times among the study participants.

The time restraint was set on the belief that real-world users tend to give up on a product that cannot be understood in less than ten minutes. Since it took about two minutes for experts to finish these tasks, the participants could have the remaining eight minute to understand system behavior.

The following describes the course of each user study:

1. The experimenter explains the aims of this study and user tasks to the participant.
2. The experimenter gives the user's manual to the participant.
3. After the participant has read the user's manual, the experimenter starts 10 minute trial session. If the participant finishes all tasks before 10 minutes, the experimenter stops the trial session. During the trial session, the experimenter records the session by videotape.
4. After the trial session, the participant fills out a questionnaire and the experimenter interviews the participant.

5.2 The First Round

The first round used the initial design of laser gesture interface described in Section 4. Specifically, all instructions were given as stroke gestures. All visual feedback was given as lines that indicate the trajectory of laser movement, which color changes from white to yellow when recognized.

It turned out that this initial design caused a log of failures. None of the three participants in this round reported to understand the system behavior. We observed they failed to complete any of the tasks, being confused and frustrated. The interview revealed the most fundamental problem with the initial design was that the participants failed to cancel the robot actions. We often observed the participants panicked when the robot started to move against their intention and they were unable to stop it. The interview also revealed the visual feedback was problematic. The initial design did not distinguish the color of the first stroke (object selection) and the second stroke (command). The participants often got confused if the system was in the first or second step.

5.3 The Second Round

Based on the results in the first round, the following revisions were made to the system:

- We changed cancellation command from a gesture to a secondary button (originally designed for returning to a previous presentation slide) on the laser pointer.
- We changed the color of the strokes in the visual feedback. The object selection stroke is shown in yellow and command stroke is shown in blue in the revised design.

The result of the second design proved to be more successful than the first. Of the three participants, two reported that they managed to understand the system behavior, with one of the two successfully completing all the three tasks in 8m33s. We observed that the easy cancellation greatly contributed to this improvement. The participants repeatedly cancelled their mistakes and tried again, which helped steady learning. The third user, who was unable to finish the tasks, did not understand the system, and tried to input command gestures before object selection.

In the interview, the participants reported a problem with the second design. They found it difficult to make the "delivery command", which required the user to start a stroke inside of the first lasso, because the laser did not appear until they pressed the laser emitting button. From this, we learned that in general, stroke gestures using a laser pointer are ineffective at "starting" at a specific position because the pointer is invisible in the beginning.

5.4 The Third Round

Based on the results in the second round, the following revisions were made to the system:

- Since it was difficult for the user to start a command stroke inside of the object selection lasso, we decided not to take the starting point of a stroke into account. A closed stroke is recognized as object collection regardless of the position (Fig. 12(b)). An open stroke that crosses the selected objects is recognized as trashing (Fig. 12(c)), and other open strokes are recognizes as object delivery (Fig. 12(a)). In addition, pause command of task control gestures was replaced by using the third button (originally designed for forwarding a presentation slide) on the laser pointer.
- We revised the visual feedback of the object selection. The previous design showed the trajectory of the object selection lasso drawn by the laser pointer. The revised version simply shows a spot light on the selected target objects (Fig. 12). This helps the user to distinguish object selection and command specification phases.

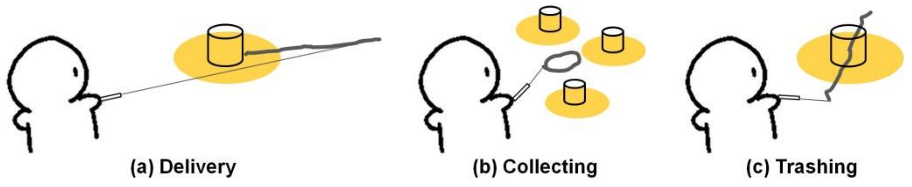

Fig. 12. Command Gestures after Object Selection (Revised for Third Round)

Both participants successfully understood the system behavior and completed all the three tasks in 6m10s and 9m30s respectively. As with the second round study, we observed the participants frequently used cancellation button on the laser pointer to cancel selection and stop unexpected executions. The participants in the third round achieved the study goal that the user gets accustomed the system behavior in under ten minutes.

Although this study was successful, there were still areas of improvement for our system. One participant, who finished the tasks earlier, commented that lassoing for object selection was somewhat bothering. The participant suggested using dwelling behavior of laser to select an object, and commented that direct dwelling gesture can be more intuitive selection method than lasso gesture.

5.5 The Fourth Round

Based on the results in the third round, the following revisions were made to the system:

- We added a direct dwelling gesture on the object for object selection in addition to lasso selection. After making one second dwelling gesture, the object under the laser is selected. The user can select a single object directly and quickly. After lasso selection, the user also can add an object using another dwelling selection. In addition, we added a function to remove an object from the selected objects by using a dwelling gesture. In other words, the user can toggle the selection status of an object.

Both two participants successfully understood the system behavior and completed all the three tasks in less than ten minutes. Both participants finished their tasks earlier than the earliest participant in the third round, with 3m54s and 5m22s respectively.

5.6 The Final Design and Execution Sequence

In our final design, object selection is realized by lasso and dwelling gestures. If the user draws a lasso in the initial state, the system selects objects in the lasso. The user can also use dwelling gesture to select and remove an object. The system provides three commands for object control and distinguishes them by stroke gesture recognition. Closed strokes are recognized as object collection. Open strokes that cross the selected objects are recognized as trashing and other open strokes are recognized as object delivery. The user can cancel or pause by pushing the extra buttons on the laser pointer.

Fig. 13 shows an execution sequence of the system. The user first selects the upper three objects using lasso selection (Fig. 13(b)). After that, the user adds the lower object using dwelling selection (Fig. 13(c)). When the user inputs the collecting command, the robot starts the task to collect the objects that have been selected (Fig. 13(d)). If the robot finishes the task, the visual feedback turns off (Fig. 13(k)) and the robot returns to its initial position (Fig. 13(l)).

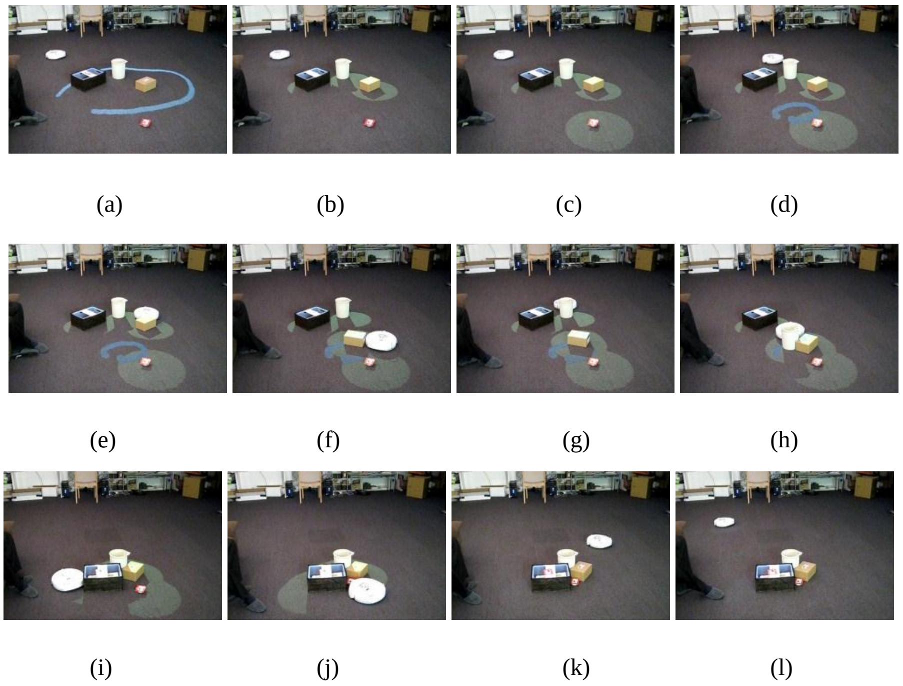

Fig. 13. Execution Sequence

6 Discussion

Through our user study, we observed the extent that a laser pointer can be used as a robot-controlling interface, and concluded that the laser pointer is good at staying at a position or jumping to another position, but not good at starting a stroke at a specific position. Starting a stroke at a specific position was one of the main difficulties in making command and task-control gestures in our initial design. However, we observed that it is still possible to use gestures to distinguish commands if the system is only concerned with the middle or end position of the path, as verified in the successful completion of tasks in the third and fourth round.

The study also showed that visual feedback is essential, especially for the first-time users. If it is not well designed, the user cannot understand the behavior of the system. Although visual feedback is essential, the light provided by the projector potentially has an occlusion problem. Designing another occlusion-free feedback is one of the future modifications in this system.

Easy cancellation methods are important for robot applications. Since the robot affects the objects in the real world, it is necessary to have an emergency stop button. In the initial design, we did not provide this feature. In the initial and second design, the users often made mistakes in specifying commands due to the difficulty in starting at a specific position. In such cases, easy cancellation especially works. Easy cancellation also helps first-time users learn the system behavior faster. In the final design, we do not use the stroke gesture interface for cancellation; however, this interface still works for specifying other commands as well as designating targets.

7 Conclusion

This paper explored the possibility of using a laser gesture interface to give instructions to a mobile robot. The user makes a gesture using a laser pointer, specifying target objects, commands, and target locations. The system then recognizes the gesture and executes the specified task by driving the robot. We implemented a prototype system by combining a robot moving on the floor and cameras on the ceiling. Using the system, we iteratively improved the interface design by conducting user studies. The results revealed the importance of easy and reliable cancellation methods, the difficulty of starting a gesture at a specific location, and the importance of understandable visual feedback. Based on the results of iterative studies, we added the dwelling gesture for object selection, eliminated considering the starting point of strokes, changed the laser gesture cancellation function to a push button function, and revised the visual feedback from stroke-based to status-based.

The main limitations of the current implementation are in using vision-based tags for object tracking and visual feedback occlusion problem. To overcome these limitations, our future work includes object tracking by other registration-free techniques, such as image-based segmentation using an overview camera, or probing objects by sensors on the robot; and designing occlusion-free feedback.

References

1. iRobot Corporation: Home Page, http://www.irobot.com
2. Kemp, C.C., Anderson, C.D., Nguyen, H., Trevor, A.J., Xu, Z.: A Point-and-Click Interface for the Real World: Laser Designation of Objects for Mobile Manipulation. In: 3rd ACM/IEEE International Conference on Human-Robot Interaction, pp. 241--248, (2008) 3. Suzuki, T., Ohya, A., Yuta, S.: Operation Direction to a Mobile Robot by Projection Lights. In: IEEE Workshop on Advanced Robotics and its Social Impacts, TPO-2-2, (2005)

4. Perzanowski, D., Schultz, A.C., Adams, W., Marsh, E., Bugajska, M.: Building a Multimodal Human-Robot Interface. IEEE Intelligent Systems, vol. 16, no. 1, pp. 16--21, (2001)
5. Lundberg, C., Barck-Holst, C., Folkeson, J., Christensen, H.I.: PDA interface for a field robot. In: 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 2882--2888, (2003)
6. Yamazaki, K., Yamazaki, A., Kuzuoka, H., Oyama, S., Kato, H., Suzuki, H., Miki, H.: GestureLaser and GestureLaser Car: Development of an Embodied Space to Support Remote Instruction. In: Sixth European Conference on Computer Supported Cooperative Work, pp. 239--258, (1999)
7. Olsen, D.R., Nielsen, T.: Laser pointer interaction. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 17--22, (2001)
8. Kirstein, C., Muller, H.: Interaction with a Projection Screen using a Camera-Tracked Laser Pointer. In: International Conference on MultiMedia Modelling, pp. 191--192, (1998) 9. Kurtenbach, G., Buxton, W.: User Learning and Performance with Marking Menus. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 258--264, (1994)

10. Sakamoto, D., Honda, K., Inami, M., Igarashi, T.: Sketch and Run: A Stroke-based Interface for Home Robots. In: 27th International Conference on Human Factors in Computing Systems, pp. 197--200, (2009)
11. Skubic, M., Anderson, D., Blisard, S., Perzanowski, D., Schultz, A.: Using a hand-drawn sketch to control a team of robots. Autonomous Robots, vol. 22, no. 4, pp.399--410, (2007)
12. Igarashi, T., Kadobayashi, R., Mase, K., Tanaka, H.: Path Drawing for 3D Walkthrough. In: 11th Annual ACM Symposium on User Interface Software and Technology, pp. 173-174, (1998)
13. Rekimoto, J., Ayatsuka, Y.: CyberCode: Designing Augmented Reality Environments with Visual Tags. In: DARE 2000 on Designing Augmented Reality Environments, pp. 1--10, (2000)
14. Fiala, M.: ARTag, a Fiducial Marker System Using Digital Techniques. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 590--596, (2005)
15. ARToolKit Home Page, http://www.hitl.washington.edu/artoolkit/
16. Baudel, T.: A Mark-Based Interaction Paradigm for Free-Hand Drawing. In: 7th Annual ACM Symposium on User Interface Software and Technology, pp. 185--192, (1994)
17. Landay, J.A., Myers, B.A.: Interactive Sketching for the Early Stages of User Interface Design. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 43--50, (1995)
18. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: A Sketching Interface for 3D Freeform Design. In: 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 409--416, (1999)