

Sketch and Run: A Stroke-based Interface for Home Robots

Daisuke Sakamoto^{1&3}, Koichiro Honda^{1&3}, Masahiko Inami^{2&3}, Takeo Igarashi^{1&3}

¹Graduate School of Information
Science and Technology,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku,
Tokyo, 113-0033 Japan

²Graduate School of Media
Design, Keio University (KMD)
4-1-1 Hiyoshi, Kohoku-ku,
Yokohama 223-8526, JAPAN

³JST, ERATO, IGARASHI
Design UI Project, 1-28-1,
Koishikawa, Bunkyo-ku, Tokyo,
112-0002 Japan

{sakamoto | honda | inami}@designinterface.jp, takeo@acm.org

ABSTRACT

Numerous robots have been developed, and some of them are already being used in homes, institutions, and workplaces. Despite the development of useful robot functions, the focus so far has not been on user interfaces of robots. General users of robots find it hard to understand what the robots are doing and what kind of work they can do. This paper presents an interface for the commanding home robots by using stroke gestures on a computer screen. This interface allows the user to control robots and design their behaviors by sketching the robot's behaviors and actions on a top-down view from ceiling cameras. To convey a feeling of directly controlling the robots, our interface employs the live camera view. In this study, we focused on a house-cleaning task that is typical of home robots, and developed a sketch interface for designing behaviors of vacuuming robots.

Author Keywords

stroke-based interface, sketching interface, stroke gesture, home robot, human robot interaction

ACM Classification Keywords

H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces - Interaction styles; I.2.9. ARTIFICIAL INTELLIGENCE: Robotics - Operator interfaces

INTRODUCTION

Significant progress has been made in robotics technology in recent years. In research environments, it is already common to see humanoid robots walking around with biped legs [5], and some of them are capable of using tools designed for humans such as knives and brooms [9]. In the commercial domain, robots are no longer limited to factory

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2009, April 4–9, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-246-7/09/04...\$5.00.

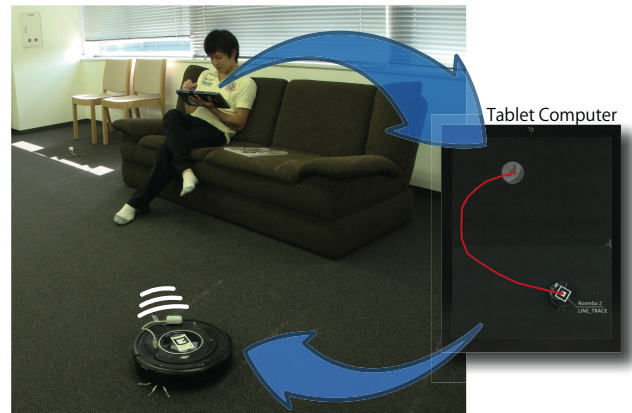


Figure 1: Controlling a home robot with a sketching interface: A user sketches the robot's movements on a tablet computer. The robot then proceeds along the sketched movements in the room.

environments and are moving into home environments, as signified by the success of vacuuming robots.

However, the old-time dream of having a housekeeping robot at home has not really become a reality. The main reasons may be cost and hardware capabilities, but we believe that the lack of vision on how to “control” advanced robots is also a significant limiting factor. One may think that simple vacuuming does not require much control, but in reality, individual users have their own needs and usually want to specify when, where, and how to clean, which is not really well supported by the current systems. Furthermore, if we have an advanced humanoid robot at home, the problem of controlling it is much more serious. Typical demonstrations of these robots assume almost autonomous behaviors, but perfect autonomy is difficult because users have their own preferences and needs, which are hard to estimate beforehand.

The above observation led to our efforts to explore better user interfaces to control robots. The goal is to develop a new form of user interface where the user can give specific and detailed instructions to home robots, such as where to go and what to do, while avoiding low-level controls such

as “move forward” and “turn left.” Natural languages are a popular option. However, in many cases, a graphical representation is much more direct and intuitive for communicating spatial information. Therefore, we examine the possibility of using a graphical representation to control robots in this work.

This paper reports on our initial experiment to build a specialized graphical interface for giving spatial information and controlling home robots. The main idea is to use sketching on a hand-held computer screen showing a top-down camera view as the user interface (Figure 1). We chose this configuration for several reasons. First, sketching on a top-down view can directly provide positional information (which can be more ambiguous with natural language). Second, sketching naturally supports asynchronous control because sketching implies an instruction staying in the environment and executed late. Third, this configuration frees the user from tedious calibration or setup because the user-interaction space (live video) is identical to the robot-controlling space.

RELATED WORKS

Various types of sketching interfaces have been proposed in the field of human-computer interaction. Forsberg et al. applied stroke gestures to a music composition system [2]. Perlin et al. developed a rapid text-writing tool with a single stroke [10]. In computer graphics, Igarashi et al. proposed a 3D modeling system [6] that enables users to easily create 3D models by drawing contour lines of the desired shape.

On the other hand, in the field of human-robot interaction, some research has been done on the teleoperation of various robots, for example, army robots [4], rescue robots [7], and humanoid robots [3, 12]. The interface of the teleoperation systems for these robots were proposed and evaluated in these studies. However, these robots are not intended for general users, and it would be hard to use these interfaces in a home environment because they require high computational power.

We consider that the user interface is the most important element for home robots. At home, the main users are who are not necessary technology oriented people such as children, elderly people and housewives. The usability of the interface must be high. Recently, some remote-control devices for home electronics products have been developed and marketed with a touch panel display. We anticipate that

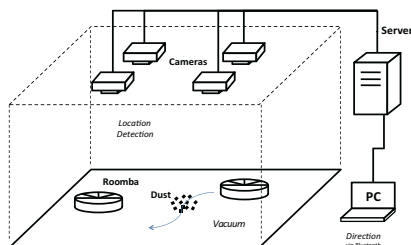


Figure 2: System Overview

because in the near future, computers and robots will be more pervasive and ubiquitous in the environment and in the home, the interfaces for controlling them will be a touch-panel-type small computer. Currently, these controllers are expensive. However, this type of controller is getting cheaper. If the remote controller for home electronics goods had more power of expression and higher functionality, we could add an interface to it to control home robots. We believe that this is a natural evolution of remote controllers for home electronics.

We believe that an intuitive user interface for robot control would extend the potential of home robots and the knowledge of graphical user interfaces (GUI) is naturally applicable to control robots. There are a couple of attempts to design intuitive interfaces for robots in the real world were conducted. Kemp et al proposed an interaction technique to designate objects with a green laser pointer [8]. However, we consider that the GUI technique is more familiar to general users. It is easy to give feedback to users.

Among existing GUI techniques, we are especially interested in pen-stroke gestures. To analyze input strokes, there is a well-known algorithm [11] that estimates specific paths based on statistics and learning. Arvo and Novins proposed a smoothing algorithm for ambiguous hand-drawn shapes [1]. In a recent study, Wobbrock et al. introduced a simple robust recognizer for single strokes [14]. We adopt this algorithm for the detection of halt and resume gestures.

As mentioned above, stroke-based interfaces are relatively common, and their potential has been demonstrated in a variety of applications. Skubic et al. developed a sketch-based interface for controlling team of robots [13]. However, their work focused on the reasoning of handwritten sketches. We focus on the design of a user interface using a ceiling-mounted live camera view. We consider that this approach is also applicable to robot controlling to enhance controllability.

SYSTEM IMPLEMENTATION

The system receives gesture commands from the user and controls the robot in the real world. Users draw freeform strokes on the computer screen to control the robot. Then the system conducts a detection and recognition process of the sketch. If a detected sketch is a command, this system executes the task by driving the robots in the real world.

In the prototype system, we focused on a floor-cleaning task in a room with a robot system because this cleaning task is a realistic one for current home robots. We describe the details of this interface below and show the system overview in Figure 2.

Home Robot

The Roomba by iRobot¹ is a commercially available product (Figure 3). It is a well-known home robot used for

¹ <http://www.irobot.com/>

vacuuming that can function autonomously. We can control the robot by using an accessory remote control. The Roomba is 92 mm high and 340 mm in diameter. It has two degrees of freedom in the base for driving. The robot can drive 500 mm per second at maximum. A user program can be developed through an open serial control called Roomba Open Interface (ROI). In our system, we control the Roomba from a computer that runs the sketching interface via the Bluetooth Serial Port Profile (SPP).



Figure 3: iRobot Roomba 570

Top-down View

We believe that a top-down view, which is an actual view of a room from a ceiling camera, can give users a more realistic feeling of controlling the robot than a robot-centered view because the user can observe the robot’s state in the actual room from a top-down view. For this reason, we propose a behavior direction interface by using a top-down view of ceiling cameras. This makes it possible to give users the feeling of sketching on the actual room.

A top-down view is also useful for detecting the location of robots and objects because it is not necessary to apply coordinate transformation. We can reduce the error by using a vertical image. A single ceiling camera can capture a 2.25 x 3 [m] area. However, this is too small to cover the working area. Therefore, we combine images from four cameras into a single global image. The combined image size is 4.5 x 6 [m].

Object detection

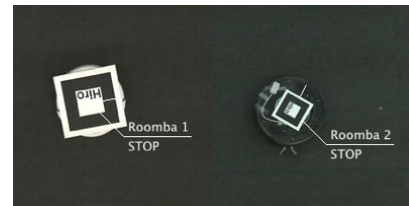
This system uses ARToolkit² to detect the locations of robots and objects (Figure 4). ARToolkit detects objects by using a specified marker in a given image. The marker is a squared image like a 2D barcode. Note that ARToolkit is only used as an initial test bed. It lacks speed and accuracy necessary for more advanced control.

Prepared gesture commands

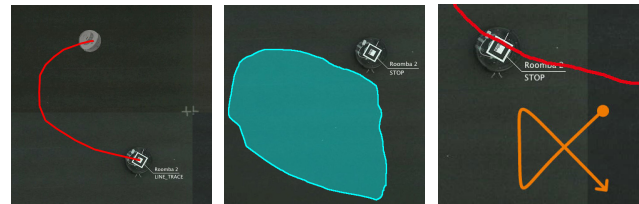
To control a robot using a stroke gesture, we prepared three gestures. We explain how commands are executed by the stroke gestures below.

Move command

This interface recognizes a line starting from the robot as a path the robot should follow. This line can be a curve or a zigzag stroke. It must be drawn in a single stroke (Figure 5, left)



• **Figure 4: Robot detection by ARToolkit**



• **Figure 5: Stroke gestures (left: move; center: vacuum; right: stop)**

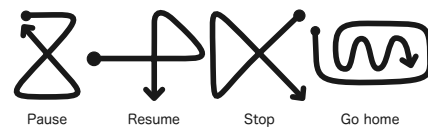
Vacuuming command

A lasso stroke is recognized as a vacuuming command (Figure 5, center). This lasso stroke must not encircle any robots. In the stroke gesture, the user cannot draw a perfect loop. Thus, we recognize a lasso if the opening is less than 20% of the entire loop.

Auxiliary commands

With users able to command home robots to perform various tasks, it will also be necessary for the robots to have “stop” and “pause” functions. We have developed four auxiliary commands for controlling robots. These commands will appear as menu items on the screen when a lasso stroke is used to select, or encircle, the robot. Furthermore, we adopt stroke recognition algorithm [14] for the detection of auxiliary commands gestures (Figure 6).

- Pause – The robot will pause; it will resume the task when given a “resume” command.
- Resume – The robot will resume its movements.
- Stop – This command will stop/cancel the robot’s movements.
- Go home – The robot will return to its base and charge its battery.



• **Figure 6: Available stroke gesture commands**

Figure 7 shows an overview of the implemented interface.

RESULTS

We conducted a pilot test of our interface at our laboratory. Five university students participated in the test. They were able to control the robot freely with the interface, even when given only brief instructions on its use. We simply told them what the functions were.

² <http://www.hitl.washington.edu/artoolkit/>

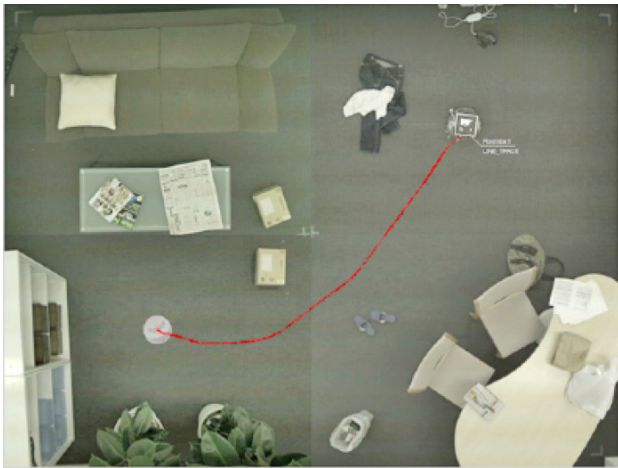


Figure 7: Implemented Interface: Sketch and Run

Once we draw a path on the interface, no other actions are required to get the robot to perform the task. This means that when the robot starts following the given path, the user can work on something else. This asynchronism is an important function of this interface. Most robots that are already on the market have asynchronism. Users want to reduce the amount of time spent on housework and increase their personal time. Our interface makes it possible for the user to save their own time.

The speed of the robot was not very fast, at 100 mm per second on average. This is because the ARToolkit that detects the location and orientation of the robot sometime made a mistake. It detected the robot four or five times per second. Furthermore, the top-down view was a combined image from four cameras. When the robot is at image bounds, the ARToolkit was sometimes unable to detect the location of the robot reliably. For these reasons, we reduced the robot's speed to be detected by ARToolkit stably. We plan to use a 3D motion capturing system to allow the robot to move faster and with sufficient stability in the next system.

CONCLUSION

We proposed an interface for controlling robots using stroke gestures. This interface allows users to control robots intuitively. For example, the user can move the robot by drawing a path, initiate vacuuming by drawing a circle, and select the robot to execute specified commands. The test users successfully controlled (or ran) robots using this interface. The stroke gesture interface provides the user with a novel experience to control a robot in home environment.

We used a vacuuming robot in this paper. However, we are not focusing only on vacuuming robots. We believe that this interface can be useful for other home robots. We want to use this interface with robots for delivering objects, picking objects up, and throwing objects into trash bins at home. In these cases, we need to develop other robots that execute the specified tasks. This is one way to familiarize

robots to the public. To do this, however, we need to study various interfaces for robots. We believe that the interface will become a more important factor in robot systems in the home.

ACKNOWLEDGMENTS

This research was partially supported by Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science (2011455).

REFERENCES

1. Arvo, J. and Novins, K. Fluid sketches: continuous recognition and morphing of simple hand-drawn shapes. *In Proc. UIST 2000*, 73-80.
2. Forsberg, A., Dieterich, M., and Zeleznik, R. The music notepad. *In Proc. of UIST 1998*, 203-210.
3. Glas, D. F., Kanda, T., Ishiguro, H., and Hagita, N. 2008. Simultaneous teleoperation of multiple social robots. *In Proc. HRI 2008*, 311-318.
4. Hill, S. G. and Bodt, B. A field experiment of autonomous mobility: operator workload for one and two robots. *In Proc. HRI 2007*, 169-176.
5. Hirai, K., Hirose, M., Haikawa, Y., and Takenaka, T. The development of honda humanoid robot, *In Proc. Int. Conf. on Robotics and Automation 1998*, 1321-1326.
6. Igarashi, T., Matsuoka, S., and Tanaka, H. Teddy: a sketching interface for 3D freeform design. *In Proc. SIGGRAPH 1999*, 409-416.
7. Kadous, M. W., Sheh, R. K., and Sammut, C. Effective user interface design for rescue robotics. *In Proc. HRI 2006*, 250-257.
8. Kemp, C. C., Anderson, C. D., Nguyen, H., Trevor, A. J., and Xu, Z. A point-and-click interface for the real world: laser designation of objects for mobile manipulation. *In Proc. HRI 2008*, 241-248.
9. Okada, K., Ogura, T., Haneda, A., Fujimoto, J., Gravot, F., and Inaba, M. Humanoid Motion Generation System on HRP2-JSK for Daily Life Environment. *In Proc. ICMA 2005*, 1772-1777.
10. Perlin, K. Quikwriting: continuous stylus-based text entry. *In Proc. UIST 1998*, 215-216.
11. Rubine, D. Specifying gestures by example. *In Proc. SIGGRAPH 1991*, 329-337.
12. Shiomi, M., Sakamoto, D., Takayuki, K., Ishi, C. T., Ishiguro, H., and Hagita, N. A semi-autonomous communication robot: a field trial at a train station. *In Proc. HRI 2008*, 303-310.
13. Skubic, M., Anderson, D., Blisard, S., Perzanowski, D. and Schultz, A., Using a hand-drawn sketch to control a team of robots, *In Auton. Robots*, 22, 4 (2007), 399-410.
14. Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. *In Proc. UIST 2007*, 159-168.