

Multi-touch Interaction for Robot Control

Mark Micire¹, Jill L. Drury², Brenden Keyes², and Holly A. Yanco¹

¹University of Massachusetts Lowell
1 University Avenue, Lowell, MA 01854 USA
{mmicire, holly}@cs.uml.edu

²The MITRE Corporation
202 Burlington Road, Bedford, MA 01730 USA
{jldrury, bkeyes}@mitre.org

ABSTRACT

Recent developments in multi-touch technologies have exposed fertile ground for research in enriched human-robot interaction. Although multi-touch technologies have been used for virtual 3D applications, to the authors' knowledge, ours is the first study to explore the use of a multi-touch table with a physical robot agent. This baseline study explores the control of a single agent with a multi-touch table using an adapted, previously studied, joystick-based interface. We performed a detailed analysis of users' interaction styles with two complex functions of the multi-touch interface and isolated mismatches between user expectations and interaction functionality.

Author Keywords

Human-robot interaction, multi-touch interaction, interaction styles.

ACM Classification Keywords

H5.2. Information systems, information interfaces and presentation: User interfaces.

INTRODUCTION

There is much excitement surrounding multi-touch tables and displays (e.g., Dietz, 2001; Han, 2005) because of the potential for enhanced interactivity. Removing the joystick, mouse, or keyboard can remove a layer of interface abstraction and increase the degree of direct manipulation (Shneiderman, 1983). In the case of human-robot interaction, this technology should allow users more direct interaction with the robot to influence its behavior.

A joystick interface limits the user to a relatively small set of interaction possibilities. Digital buttons, analog gimbals, and analog sliders are the three common input modes. The multi-touch surface is quite different, allowing for almost limitless interaction methods on a 2D plane. Where the joystick limits the user through mechanical and physical constraints, the multi-touch surface serves as the blank canvas on which control surfaces are dynamically created.

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.

IUI 2009, February 8–11, 2009, Sanibel Island, FL, USA.
Copyright 2009 ACM 978-1-60558-246-7/08/04...\$5.00

But the flexibility and freedom of the interface also presents problems for designers. Designers must carefully choose control methods that give appropriate feedback and extremely clear affordances (affordances mean that perceived and actual functionality match; see Norman, 1988). Users are accustomed to haptic feedback, such as spring-loaded buttons and gimbals, and auditory feedback, such as clicks, even from a non-force-feedback joystick controller: feedback not provided by multi-touch surfaces.

While multi-touch table technology seems to offer promise for human-robot interaction, we could not find any previous work that characterized human performance when directing physical agents (robots) via multi-touch table interaction. We wanted to understand how to best take advantage of the possibilities offered by the multi-touch modality for controlling robots. Accordingly, we conducted a user test of an interface hosted on a Mitsubishi DiamondTouch table (Dietz, 2001). This interface controlled an urban search-and-rescue (USAR) robot and was based on a well-studied joystick interface (e.g., Keyes, 2007; Yanco et al., 2007; Yanco et al., 2004) that had been adapted to take advantage of multi-touch features.

Similar to the evolutionary approach we have taken in joystick-based interfaces, we expect the DiamondTouch-based interface to evolve. This study aids the evolutionary process by providing a detailed analysis of users' varied interaction styles. Our analysis sheds light on how users perceive the interface's affordances and highlights mismatches between users' perceptions and the designer's intentions. These mismatches point towards design changes to better align users' expectations and interface realities.

INTERFACE DESCRIPTION

The interface contains six panels or groupings of interface components, with the main video panel occupying a central area. The main video panel has a cross-hair overlay to indicate the video camera's current pan and tilt orientation.

The main video panel also functions as the camera control mechanism. As annotated in Figure 1, the user presses and holds a finger on the region of the screen corresponding to the direction of movement. The user presses the left center quadrant to pan left, upper center quadrant to tilt up, etc. To zoom in, the user places their two index fingers in the center of the image and then rapidly moves them diagonally towards the corners of the screen. This movement increases the zoom factor by two times for each motion. Tapping

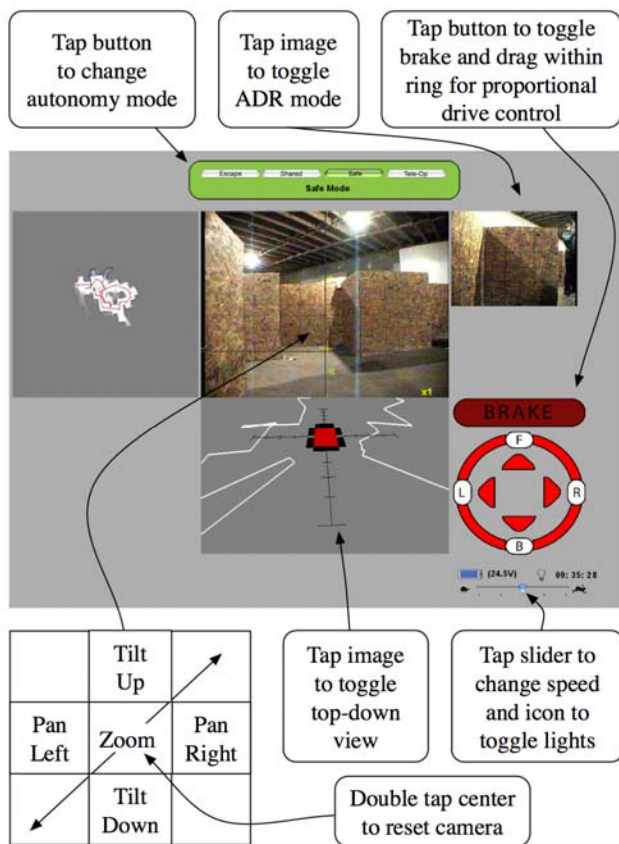


Figure 1. Interface overview with description of interaction gestures.

twice in the center of the image re-centers the pan and tilt and resets to a one times zoom factor.

The small rear view panel, which displays the video feed from the robot's rear camera, is located to the upper right of the main video panel. The rear camera's video stream is mirrored to imitate the view one receives from a rear view mirror. If the user wants to see a larger view of the rear camera's video they can switch to Automatic Direction Reversal (ADR) mode. This mode causes the rear camera's video to be displayed in the larger main video panel while switching the forward-facing camera's video to the rear view mirror panel. This act also reverses the driving commands as well as the ranging information displayed on the distance panel directly under the main video panel. To select the ADR mode, the user taps the small video panel, causing it to swap its contents with the main video display.

The distance panel depicts the robot's chassis with white lines displayed around it, identifying distance readings from the sonar sensors and the laser range finder. These lines help give users a frame of reference regarding how close objects may be to the robot. This panel can be toggled from a perspective view to a top down view by tapping the panel.

To the left of the video panel is a map that is dynamically generated as the robot moves around an area. The robot's location is a green triangle and its trail is a red line.

The mode panel is displayed above the main video panel. This panel contains four buttons that each represent one of the autonomy modes of the system. The user taps the corresponding button to engage the requested autonomy mode.

The drive control panel in the lower right-hand corner of the interface is a visual analog to a joystick, but in a top-down view. The relative position of the user's fingertip within the panel is translated into rotation and translation vectors. The panel changes color from red to green when the user engages the control. Directly above this panel is a button that engages and disengages the robot's brake.

Directly below the drive control panel is a group of status icons. The user taps the light bulb icon to turn the robot's lights on or off. A slider below these icons provides a speed limiting scalar. The user taps or slides a finger along the scale to set the desired top speed for the robot.

METHOD FOR CHARACTERIZING INTERACTION

During usability tests, we videotaped over users' shoulders, captured users' screens, logged users' commands, and took observation notes. We also obtained performance data (not reported on here) by videotaping the robot, hand-drawing maps of the robot's paths, noting driving problems (bumps, scrapes, etc.) and degree of success in task completion. This paper reports on the analysis of the use of the camera and driving controls. The other controls, such as those to turn the lights on and off, required much simpler motions.

Data analysis began by looking for large-scale patterns of movement. Once we identified these patterns, we reviewed the video to broadly characterize each participant's interaction with the interface.

We further refined our description of the patterns of interaction by isolating components of actions using open and axial coding from grounded theory (Glaser and Strauss, 1967). Open coding involves analysts noting data that can be grouped or categorized while axial coding is a process of refining the groupings. Once types of interactions emerged we matched (coded) instances of users' gestures to them. To ensure reproducibility of the data analysis, we computed Cohen's Kappa statistic for two coders and found very good agreement: .84 after chance was excluded (agreement was .88 if chance was not factored out).

Finally, to illustrate these user-developed interaction styles, we developed narratives of each participant's interactions.

RESULTS OF OPEN AND AXIAL CODING

We identified three components of interaction: gesture magnitude, gesture alignment, and gesture sequence.

Some users acted as though they thought the magnitude of their gestures was significant, while others did not. We developed two categories of movement that described two different approaches to the magnitude of users' gestures. One category, called proportional velocity, assumes that movement and speed are controlled together with gesture magnitude perceived to be important. The user perceives that the location of their finger with respect to the center of

the drive control panel determines both direction and speed, with speed increasing as the distance from the center of the panel grows. The other category, discrete velocity, assumes that the user expects to control direction of movement independent of speed, and also assumes that gesture magnitude is unimportant.

The second component, gesture alignment, describes how users' gestures were aligned with respect to the x- and y-axes created by the table borders. We developed two categories for this component: when users confine their gestures to a cross-like area of up-down and side-to-side motions (on-axis movement) and when they make gestures outside of the x- and y-axes (off axis movement).

The third component, gesture sequence, refers to the pattern of primitive interactions we call touch, drag and hold. We identified three categories of gesture sequences. "Touch" starts with no contact and then a brief lowering of a finger onto the table. "Touch, drag, and hold" refers to a tap followed by a sweep and prolonged contact in the final position without breaking contact with the table. "Repeated touch and drag" refers to several (or many) similar short sequences that each end with lifting finger(s) off the table.

Note that the virtual joystick in the drive control panel was intended by the designers to be used with proportional velocity movements, off-axis gesture alignment, and touch gestures. Camera control was designed to be used with discrete velocity movements, on-axis gesture alignment, and touch gestures. We analyzed videotapes from each user, coding their actions into these categories, then compared their gestures to the designers' intended gestures.

NARRATIVES OF USERS' INTERACTIONS

The following are excerpts of the narrative description of users' employment of the gesture categories described above. Many "incorrect" actions indicate failure of the design to communicate how to interact with the interface, even though users did receive a training session. Users were emergency responders.

User 1. This user had a continuous, flowing movement on the drive control panel with a frequent "touch, drag, and hold" approach. Exclusively using his middle finger on his right hand, he made use of proportional control 100% of the time. 55% of his movements were in off-axis areas of the drive control panel. The user also employed continuous, proportional control gestures on the camera control panel even though an experimenter explained to him that the camera was controlled through touching discrete grid locations on the panel, as shown in Figure 1. This example reinforces the often-referenced design principle of consistency (e.g., see Nielsen, 1993): in this case, that the control panels should all adopt the same movement model.

User 2. The second user employed the widest variety of interaction methods. She began her movements in the center of the control panel and quickly moved to the top or bottom of the control surface indicating an initial understanding of proportional velocity control. She never

made movements off of the vertical axis until she was at the top or bottom of the control surface. She would then trace the outer ring of the control surface and repeat this action approximately every two seconds. The user was attempting to "turn" the outside ring of the control surface like a steering wheel in a car. After an experimenter explained that the outer ring was not a steering wheel the user placed her finger on the edge of the control panel to rotate the robot. Rather than her previous approach of smooth motions in constant contact with the surface, she began tapping rapidly in the desired direction. These "touch" movements sometimes exceeded four taps per second, and accounted for 89% of her drive control movements. She used on-axis gesture alignment for 55% of her movements.

User 3. The third user employed a "repeated touch and drag" strategy for robot movement in a mix between the proportional velocity and the discrete velocity categories. After approximately five minutes, the hybrid proportional and discrete finger movement began to resemble a trackpad-like movement that one might make with laptop mouse control surfaces. The user seemed to be pushing or dragging the robot in the desired direction in small steps instead of continuous button presses or proportional control. In contrast, the user pressed the sides of the video control display using discrete velocity control.

User 4. This user employed proportional velocity gestures with multiple fingers, much like a piano player, to shorten the amount of area that his fingertips were required to traverse. For example, if the middle fingertip was at the top of the control, indicating 100% forward translation and 0% rotation, and he wanted to switch to a 100% left rotation, he would lower his index finger. Upon making contact with his index finger, he would slowly raise his middle finger and allow drive control to transfer to the index finger. This method had the effect of providing very smooth transitions in what would otherwise have been a "touch" style gesture sequence. This user continued to use proportional velocity gestures on the camera control panel even though discrete control was the only method described by experimenters and demonstrated by the robot's pan-tilt unit.

User 5. This user began with proportional velocity gestures, but after two minutes he began pressing the inner directional triangles exclusively. His interaction with these buttons was a mix of proportional and discrete velocity control. Regardless of the control method, he never moved outside of the circular boundary created by the outside of the triangular button images. This artificial boundary meant that the robot never accelerated to full translation or rotation. Like User 4, he used multiple fingers to activate the drive control panel but maintained very discrete finger contact with the triangular buttons. His discrete velocity control and "touch" gestures allowed him to interact with the camera control panel as the designers intended.

User 6. Immediately upon starting her run, the sixth user established a style that used only on-axis gesture alignment. She would begin in the center of the control panel and then

quickly move to the outer ring, establishing 100% translation or rotation, but only one at a time. She used on axis movements 100% of the time, with 76% ending at the outer ring, and 76% of these were proportional velocity commands. When she interacted with the camera control, she only touched the edge of the image even though she had been shown that the pan and tilt control area was much larger. She accidentally triggered nearby controls as a result. This finding reinforces the design criterion derived from Fitts' Law (Fitts, 1954) that the distance between controls should minimally be the width of the users' fingertips to avoid accidental interference.

DISCUSSION AND CONCLUSIONS

In robotics, the term "emergent behavior" is used to describe unintentional or surprising combinations of behaviors or interactions with the environment. These emergent behaviors are unintentional artifacts that may or may not contribute to the desired outcome of the robot's task. During user testing, we found that the novelty of the multi-touch surface created opportunities for emergent interactions that were not anticipated by the system designers. Although each participant was trained on the interface in the same way, they adopted their own interaction styles borrowed from various devices in the world. While the system designers intended the interface to evoke a joystick and button affordance, the participants also demonstrated motions similar to those they would use with mouse track-pads, piano keys, touch-typing, and sliders. This behavior tells us that we need to revise the design to better align perceived affordances and actual functionality.

We did not realize the extent of the differences between the control models used for the camera control versus the driving control until we analyzed users' movements in detail. In hindsight, the inconsistency of movement models seems clear. It now seems obvious that we need to change the interaction to ensure that both types of movement controls use the same model, either proportional or discrete.

Future development will also involve ensuring that the distances between controls is significantly larger than the width of a finger to minimize accidental control triggering.

We are also considering moving the status icons to the lower left-hand corner. We had located them in the right-hand corner because that is where they were located in the earlier joystick-based interface. We wished to keep the interface as similar as possible to the joystick-based interface for this baseline study so we could better compare the two (although we do not report on the comparison in this paper). But now that the baseline study is complete, we can make more significant changes to the interface.

We also plan to undertake longer studies to determine whether there are issues with fatigue due to the longer

distances that users must move their hands and arms, when compared with the joystick-based interface. To mitigate fatigue, we may need to cluster frequently used controls.

The users were enthusiastic, and we are encouraged by the results to continue evolving the interface.

ACKNOWLEDGMENTS

The authors wish to thank the user study participants. We also thank Harold Bufford, Amanda Courtemanche, Munjal Desai, and Kate Tsui for their data gathering assistance. Thanks to Mitsubishi Electric Research Laboratories for the extended use of a DiamondTouch tabletop. And finally, thanks to Adam Jacoff, Elena Messina, Salvatore Schipani and Ann Virts at NIST for the use of their facilities for and support throughout the study. This research was supported in part by NSF IIS-0415224.

REFERENCES

1. Dietz, P. and Leigh, D. DiamondTouch: A multi-user touch technology. In *Proc. of the ACM Symposium on User Interface Software and Technology*, pages 219–226, November 2001.
2. Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 381-391, 1954.
3. Glaser, B. G. and Strauss, A. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine de Gruyter, Hawthorne, NY, 1967.
4. Han, J. Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. of the ACM Symposium on User Interface Software and Technology*, 2005.
5. Keyes, B. MS Thesis: Evolution of a Telepresence Robot Interface. Univ. of Massachusetts Lowell, Lowell, MA, 2007.
6. Nielsen, J. *Usability Engineering*. Academic Press, San Diego, CA, 1993.
7. Norman, D. A. *The Design of Everyday Things*. Basic Books, New York, NY, 1988.
8. Shneiderman, B. Direct manipulation: A step beyond programming languages. *IEEE Computer*, 16:57–69, 1983.
9. Yanco, H. A. and Drury, J. L. "Where Am I?": Acquiring situation awareness using a remote robot platform. In *Proc. of the IEEE Conference on Systems, Man and Cybernetics*, 2004.
10. Yanco, H. A., Keyes, B., Drury, J. L., Nielsen, C. W., Few, D. A., and Bruemmer, D. J. Evolving interface design for robot search tasks. *Journal of Field Robotics*, 2007.