Performance of Multi-Touch Table Interaction and Physically Situated Robot Agents

Mark Micire
Univ. of Massachusetts Lowell
One University Avenue
Lowell, MA, 01854
mmicire@cs.uml.edu

Jill L. Drury and
Brenden Keyes
The MITRE Corporation
202 Burlington Road
Bedford, MA, 01730-1420
{jldrury,bkeyes}@mitre.org

Holly A. Yanco and
Amanda Courtemanche
Univ. of Massachusetts Lowell
One University Avenue
Lowell, MA, 01854
{holly,acourtem}@cs.uml.edu

ABSTRACT

Recent developments in multi-touch technologies have exposed fertile ground for research in enriched human-robot interaction. Although the 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. The field test shows that multi-touch interaction does not in any way impair the performance of the user in a navigation and search task. In fact, our results show an increase in learnability over the original design using joystick and keyboard-based control mechanisms. Further, we analyzed users' interaction styles with the multi-touch interface in detail to isolate mismatches between user expectations and interaction functionality.

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User Interfaces*

General Terms

Design, Human Factors, Measurement, Performance

1. INTRODUCTION

The last few years have shown a great deal of interest in multi-touch tabletop and screen display research. Hardware solutions like the Mitsubishi DiamondTouch [1] and Touchtable by TouchTable, Inc., have been in low volume production for some time now. Increases in processor and graphics co-processor speeds have allowed for innovative software solutions that now rival the responsiveness of the exclusively hardware solutions [3]. Even commercial entities such as Phillips and Microsoft have been introducing new prototypes of various hybrid solutions.

Regardless of the design of the multi-touch interface, it is difficult to argue the enhanced interactivity of such a display. By removing the joystick, mouse, or keyboard from the interaction, we increase the degree of direct manipulation, thereby increasing interaction by removing a layer of interface abstraction [9]. In the case of human-robot interaction, this should allow users to more directly interact with the robot and affect its behavior. To the our knowledge, this study represents the first use of a multi-touch table with a physical agent. Many unexpected events occur when a system contains a moving, semi-autonomous physical object that is affecting the world. As such, we must determine if multi-touch interaction decreases the performance of systems in the real, dynamic, and noisy world.

A mature and well-studied joystick interface forms a baseline for comparison [4][11]. The University of Massachusetts Lowell (UML) Urban Search and Rescue (USAR) Interface system encompasses a wide range of user functionality and autonomy capabilities. While leaving the visual presentation the same, this system was ported from a joystick and keyboard interface to a Mitsubishi DiamondTouch [1]. A description of the original joystick design and DiamondTouch features is provided in Section 2. The similarity in design enables us to test whether we are impairing performance with the new interaction method (see Sections 3 and 4).

Beyond establishing a baseline, the new DiamondTouch-based interface will need to evolve as we learn more about how to best take advantage of the multi-touch technology. This study assists in the evolutionary process by providing a detailed analysis of users' varied interaction styles. Our analysis, described in Section 5, sheds light on how users perceive the interface's affordances [6] and highlights mismatches between users' perceptions and the designers' intentions. These mismatches point towards design changes to better align users' expectations and interface realities.

2. DESCRIPTION

The UML USAR interface, which has evolved as a result of usability studies, was originally designed to test the recommended guidelines produced by [8] and [10] to improve situation awareness. These guidelines proposed that all USAR interfaces should include a map of where the robot has been, more spatial information about the robot in the environment, indications of the current camera position, and fuse data to lower the cognitive load on the user [4].

The interface consists of six panels that make up the interface. The most frequently used is the main video panel.

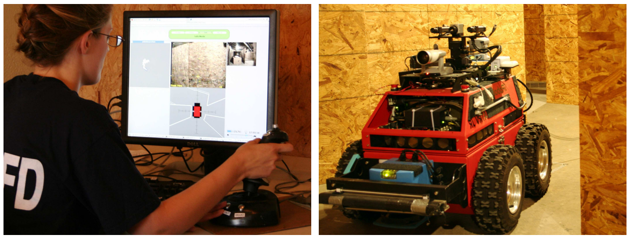


Figure 1: The UML USAR interface (left) is shown with a participant using the joystick configuration. This interface allows the user to operate the iRobot ATRV (right) though the NIST USAR course.

It is in the center of the interface and acts as the center of focus for the user. We observed in many studies that all users rely heavily on the main video screen and very rarely notice other important information presented on the interface [10]. For this reason, all of the important information, such as ranging information, is presented on or around the main video panel. The main video panel has a cross-hair overlay to indicate the current pan and tilt orientation of the video camera.

The 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. This panel is placed there to mimic the location of a car's rear view mirror. Similarly, the rear camera's video stream is mirrored to imitate the view one receives from a rear view mirror. This panel is smaller than the main video screen, so the operator does not get confused as to which one is the main camera. However, 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 relegating the front camera's video to the smaller rear view mirror panel. This act also reverses the driving commands as well as the ranging information displayed on the distance panel. This reversal makes driving while looking out the rear camera the same as driving while looking out the front video camera.

The distance panel is located directly under the main video panel. This panel consists of an image of the robot's chassis, with white lines displayed around it representing distance readings from the sonar sensors and the laser range finder. There are also black tick marks on each side of the robot, each representing 0.25 meters. This helps to give the user a frame of reference regarding how close objects may be to the robot. When a user pans the video camera, this panel rotates in the opposite direction of the pan to line up what the operator is seeing in the video with the ranging information being displayed. This panel is rendered in a perspective view by default; the operator can toggle it to a top down view.

To the left side of the video panel is a map. This map, which uses a simultaneous localization and mapping (SLAM)-based algorithm, is dynamically generated as the robot is maneuvered around an area. The map shows open space as white space, obstacles are depicted as black lines, and grey represents unexplored space. The map shows the robot's location as a green triangle and its trail is a red line.

The mode panel is displayed on top of the main video panel. This panel consists of four buttons that each represent one of the four autonomy modes of the system. When a mode is selected, the button corresponding to is highlighted, and the background color is changed.

The status panel is located on the bottom right of the interface. It is the only panel that does not border the video panel; it contains information that is not as critical as the video displays and distance information. This panel contains the current battery level, whether or not the lights are on, and the robot's maximum speed indicator.

2.1 Joystick Interface Design

The interface uses a ten button joystick that consists of a trigger, five standard buttons, and a hat sensor located on the top of the joystick. To move the robot, the user must press and hold the trigger, so that if the joystick is accidentally pushed, the robot will not move. The robot is controlled by pressing the trigger and pushing the directional gimbal on the joystick in the direction of the desired motion. Full mixing of translation and rotation is provided. If ADR mode is active, forward and backward is reversed, as explained in the previous section.

Camera controls occupy all but two of the buttons on the joystick. We decided that camera controls should all be on the joystick because maneuvering the camera is the task that takes the most time after navigation. The pan and tilt actions are controlled by the hat sensor on the top of the joystick. Directly beneath the hat senor on the joystick is a button that will "home" the camera by bringing it back to its centered position.

On the base of the joystick are four buttons. The left-most button toggles the ADR mode. The bottom left and bottom right buttons control the zoom feature of the camera. The button to the right of the handle toggles the robot's brake. There is also a scroll wheel on the left side of the handle that controls the robot's maximum speed.

As the joystick does not have enough buttons to fulfill all the functionality of the interface, six actions have been relegated to the keyboard. Changing autonomy modes is set to buttons F1-F4. The lighting system on the robot is toggled on and off by pressing the caps lock key. Changing the distance panel from its perspective view to the top down view is accomplished by pressing the F8 key.

2.2 DiamondTouch Interface Design

We spent a considerable amount of time ensuring that the multi-touch interface was as visually identical to the above mentioned joystick interface design as possible. The goal was to duplicate all of the functionality without creating any confounding issues in presentation or arrangement of display elements. Each of the discrete interaction elements is shown in Figure 2 and described below.

Autonomy Mode Panel: At the top of the interface is a rectangular panel populated with the four autonomy modes as described above. The participant simply needs to tap the corresponding button to engage the requested autonomy mode. Visual feedback is provided by changing the background color of the panel depending on the mode.

ADR Mode Panel: The upper right of the interface shows the view from the rear camera mounted on the robot. This mirrored image is also the panel that selects the ADR mode. The user taps the small video panel, and it switches with the main video display. While in ADR mode, the drive control panel functions are inverted.

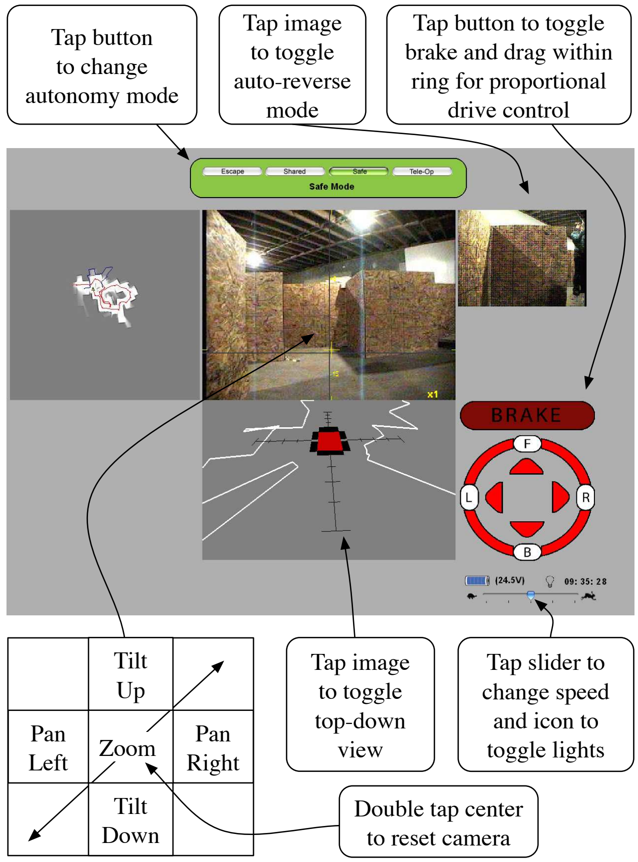


Figure 2: Screenshot and guide to gestures that activate interface features and autonomy modes.

Drive Control Panel: The drive control panel is the only visual element that was not included in the original joystick design. This panel addresses the need to duplicate the proportional velocity control of the translation and rotation of the robot. The interface panel is a visual analog to the joystick, but in a top-down view. The user places their finger inside of the ring, and the relative position of their fingertip within the panel, horizontal and vertical, is translated into rotation and translation vectors respectively. The panel changes color from red to green when the participant engages the control. This panel is positioned in the lower right had corner of the interface to position it near the participant and maintain the right-handed configuration of the original joystick interface. Directly above this panel is a simple button that engages and disengages the brake mechanism on the robot. Visual feedback for the brake is provided through the button brightness, "lighting" and "dimming" as the brake is engaged or disengaged respectively.

Status Panel: Directly below the drive control panel is a group of icons representing the battery state of the robot, the external light state, and the time since the start of the run. The user taps the icon for the light to engage or disengage the lights on the robot. The icon turns yellow or gray relative to the state of the lights. A slider below these icons provided a speed limiting scalar. The user taps or "slides" the control to the desired top speed for the robot.

Distance Panel: Directly below the main image, the distance panel gives a combined display of the sonar and laser range readings relative to the robot. The user can choose between a perspective display (shown in Figure 2) or an overhead display by tapping in the panel.

Camera Control Panel: The camera control panel allows the participant to affect all of the functions of the pan and tilt zoom cameras on the robot. As shown in Figure 2, the user presses and holds their finger on the region of the screen corresponding to the direction of movement. The left center quadrant is pressed for pan left, upper center quadrant for tilt up, etc. To zoom in, the user begins with their fingers in the center of the image and then rapidly expands them diagonally to the corners of the screen. This movement increases the zoom factor by two times for each motion. The use then can tap twice in the center of the image to recenter the pan and tilt and reset to a one times zoom factor.

It should be noted that, outside of the drive control panel, we made no visible changes to the interface. Despite this, the DiamondTouch was immediately able to provide more functionality that the joystick could alone. For example, the autonomy mode selection was offloaded to the keyboard in the joystick interface due to a limited number of buttons. In the case of the DiamondTouch, the buttons that were already displayed were used for this purpose. This "free functionality" was also true for the distance panel and light control.

3. METHODOLOGY

The goal of the experiment was to compare participants' performance and interaction with two different versions of the same interface: one based on a "traditional" PC and joystick, and the other based on a multi-touch table. Accordingly, we designed a within-subjects experiment so that each participant would use both interfaces.

We conducted the experiment in the Reference Test Arenas for Autonomous Mobile Robots at the National Institute of Standards and Technology (NIST) [7]. We used a portion of the arena that was configured as a maze of wooden panels. The panels formed corridors that required tight turns and close attention to the relationship of the robots to the wooden walls. NIST personnel placed mannequins and baby dolls within the maze to represent victims of a catastrophe.

3.1 Procedure

After signing a consent form, participants filled out a pre-experiment questionnaire requesting demographic information and probing their relevant experience with computers, robots, remote control vehicles, video games, and joysticks. Participants consisted of six people (4 men, 2 women), ranging in age from their 20's to 60+, who are members of the search and rescue community. All have used PCs for at least five years. Four consider their computer expertise to be moderate and two assess their expertise to be at the expert level. Five participants have never previously used robots, and the remaining participant had taken part in an experiment of ours several years ago during which time he used a much different control interface. Three participants have previously used remote control cars occasionally or many years ago. Four participants never play video games and two participants play video games 4 and 8 hours a week, respectively. Five participants have previously used a joystick, with one of them assessing his joystick expertise as good, three as average, and one as poor.

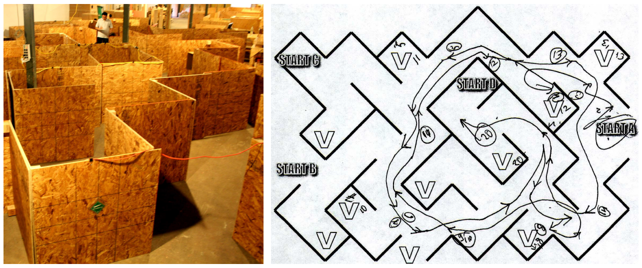


Figure 3: The robot operated in the NIST USAR arena (left) while experimenters uses ground truth maps (right) to record specific incidents.

We showed the participants what the robot looks like and then trained them on how to control the robot using one of the interfaces. We then allowed participants time to practice using the robot in a location outside the test arena and not within their line of sight so they could become comfortable with remotely moving the robot and the cameras. We then moved the robot to the arena and asked them to maneuver through the area to find as many victims as possible during a 25-minute period. We asked participants to "think aloud" [2] during the task so we could determine when participants were having trouble with parts of the interface and/or had a different mental model of how the interface works than was intended by the designers. After task completion, an experimenter asked six Likert scale questions. We then repeated these steps using the other robot interface.

We counterbalanced the experiment in two ways to avoid confounders. Three of the participants started with the joystick interface and the other three started with the multi-touch interface. Additionally, we used two different starting positions in the arena so that knowledge of the arena gained from using the first interface would not transfer to the use of the second interface. The two counterbalancing techniques led to four different combinations of initial arena entrance and initial interface.

3.2 Data collection

We collected four types of data: video, logs, observer notes, and annotated maps. Besides a video of the robot's progress through the arena, we videotaped over the shoulder of each participant to capture his/her interactions, and we mounted a video recorder pointing down at the multi-touch table. We videotaped a direct output of the PC's screen to record the state of that interface at all times. Custom logging software captured each time the participants changed modes, moved the camera, or activated other controls. An experimenter sat with the participant and hand-wrote observations. Finally, an experimenter following the robot manually marked its progress through the maze on a run sheet that also provided space to note when and where each bump, scrape, or "e-stop" (emergency halting of the robot) occurred. Figure 3 contains a reproduction of the map portion from a run sheet that shows numbers coded to specific incidents that are enumerated on the run sheet's second page.

4. EXPERIMENTAL RESULTS

The two interfaces differ in at least two major ways: in ergonomics and in the degree of direct manipulation that each attains.

The differences in ergonomics can best be explained by briefly describing the necessary physical interactions. During one run, participants sat at a table containing a standard 17" PC display, a multi-button joystick, and a keyboard. Using the joystick necessitates pulling a trigger and moving the whole hand in the desired direction, sometimes while simultaneously activating other buttons such as the one on the top of the joystick. Users moved a hand to the keyboard to activate autonomy mode changes. For the other run, participants sat in front of the 36" multi-touch surface, which was canted up at a slight angle, and extended their arms over the interface surface to activate the desired control. Thus the nature of the movements necessary for the two interfaces differ substantially. Also, the visual perception of the interfaces differ because the same number of pixels viewed on a 36" table look much less "crisp" than when viewed on the 17" display.

Differences in the degree of direct manipulation [9] – characterized by highly-visible, rapid, incremental, and reversible actions – have implications for the cognitive load required by each interface. A joystick is a pointing device and thus inserts a layer of indirection between the user and the interface. The user must mentally translate from the desired robot movement to the hand movements necessary to give the commands for those movements. In contrast, the DiamondTouch table affords more direct manipulation: to move the camera, for example, the operator puts a finger on the video display and moves the finger in the direction of the desired motion.

Since the two interfaces make use of the same graphical elements and provide the same functionality, we hypothesized that performance using the two interfaces would be comparable. But because of the differences just described, we could not be sure. Thus we tested this hypothesis by measuring the number of victims found, the amount of new area covered, and the number of destructive incidents incurred by participants when using each interface. Also because of the differences described above, we hypothesized that participants might form different subjective judgments regarding each interface's helpfulness, comfort, efficiency, ease of use, ease of learning, and pleasure/irritation level.

4.1 Performance

We assessed the positive, or constructive, aspects of performance based on measuring the number of victims found and the amount of new or unique territory the robot covered while traversing the arena. These measurements are related because it is difficult to find additional victims if the operator is not successful in maneuvering the robot into previously unexplored areas. Table 1 shows that participants explored an average of 376 square feet and found an average of 5 victims when using the joystick-based interface. The DiamondTouch interface shows remarkably similar results: participants directed robots to 373.3 square feet of territory and found 5.7 victims. Thus, there is no difference in the constructive performance of the two interfaces.

We also assessed the negative, or destructive, aspects of performance. Damage to the robot may delay or curtail real-life rescue operations, and damage to the robot's surroundings may result in causing unstable structural members to fall and injure trapped victims. We categorized the destructive incidents as pushes (the robot moves an obstacle away

Table 1: Constructive performance in the USAR arena.

	Joystick Interface		DiamondTouch Interface	
Participant	New Area Discovered	Victims Found	New Area Discovered	Victims Found
1	272	3	304	6
2	288	3	288	2
3	352	3	240	3
4	480	8	480	7
5	384	7	464	6
6	480	6	464	10
Average	376	5	373.3	5.7
Std Dev	90.4	2.3	107.4	2.9

from its normal position), scrapes (some part of the robot brushes up against an obstacle), bumps (the robot impacts an obstacle), and e-stops (experimenters fear severe damage will occur if operations continue and so halt the robot).

Table 2 contains the numbers of destructive incidents for the joystick and DiamondTouch interfaces. Note that the numbers vary widely: the standard deviations in each case are larger than the average values. While there are more scrapes and bumps using the joystick interface and more pushes and e-stops with the DiamondTouch interface, none of the differences are significant. (Paired, two-tailed t-tests with five degrees of freedom result in values of 0.32, 0.64, 0.33, and 0.31 for pushes, scrapes, bumps, and e-stops, respectively.) Thus we confirmed that there was no difference in constructive or destructive performance when using the two interfaces as they are currently designed.

Note that the interface design was originally developed with the joystick in mind and has previously gone through multiple iterations as a result of earlier user testing (see [4]). Now that we know that performance is not degraded by the act of porting the interface to the DiamondTouch table, we can begin optimizing the design for use with multi-touch interaction based on incorporating what we learn from participants' subjective feedback and a detailed understanding of how they interacted with the interface.

4.2 Subjective Assessment

To get a first look at participants' preferences, we asked them six Likert-scale questions. Using a scale of one to five, we asked how they would rate each interface along six dimensions: hindered in performing the task/helped in performing the task, difficult to learn/easy to learn, difficult to use/easy to use, irritating to use/pleasant to use, uncomfortable to use/comfortable to use, inefficient to use/efficient to use.

Prior to the experiment we conjectured that participants would find the DiamondTouch interface easier to learn and use and to be more efficient. The rationale for the ease of learning is that the controls are more dispersed over the table and incorporated into the areas that they relate to, as opposed to being clustered on the joystick where users must remember what motions and buttons are used for what functions. The predictions for ease of use and efficiency spring from the postulation that an interface with a higher degree of direct manipulation will be easier and faster to use.

Table 3 shows that the DiamondTouch interface scored the same or higher on average in all categories, although four of these categories evidenced no statistically significant difference. We found weak significance using a paired, 1-tailed t-test for ease of learning ($p = 0.088$, dof=5) and efficiency ($p = 0.055$, dof=5), and assert that it is likely we would have attained true significance if we had had access to several more participants.

We believe that the scores given the DiamondTouch interface for ease of use and irritating/pleasant to use suffered because of several implementation problems. Sometimes the robot did not receive the "recenter camera" command despite the fact that the participants were using the correct gesture to send that command, requiring the participants to frequently repeat the recentering gesture. At other times, the participants attempted to send that command by tapping on the very edge of the region in which that command could be activated, so sometimes the gesture was effective and at other times it failed, and it was difficult and frustrating for the participants to understand why the failures occurred. Also, it was not always clear to participants how to form the optimal gestures to direct the robot's movement. We discuss what we mean by this and also characterize a number of gesture styles used by the participants in Section 5.

4.3 Learnability Assessment

Because differences in Likert-scale scores for ease of learning were on the edge of significance, we looked for other supporting or disconfirming evidence. We noted that participants asked questions about how to activate functions during the runs, which we interpreted as indication that the participants were still learning the interface controls despite having been given standardized training. Accordingly, we investigated the number of questions they asked about each system during the runs as well as the number of times they showed uncertainty in finding a particular function such as a different autonomy mode. We found that five of the six participants asked a total of eight questions about the joystick interface and one participant asked two questions about the DiamondTouch interface ($p = 0.072$, dof = 5 for paired, 1-tailed t-test). This result, while again being on the edge of significance due to the small sample size, tends to support the contention that the DiamondTouch interface is easier to learn than the joystick interface.

5. INTERACTION CHARACTERIZATION

We concentrated on the camera and driving controls when characterizing the approaches participants used with the DiamondTouch interface. The other controls, such as to turn the lights on and off, required only simple tapping motions that were easily mastered by participants. In contrast, the two movement controls involved more degrees of freedom and, in the case of the moving the robot (versus the cam-

Table 2: Number of destructive incidents in the USAR arena.

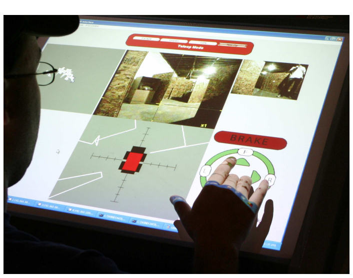
	Joystick Interface				DiamondTouch Interface			
Participant	Pushes	Scrapes	Bumps	E-stops	Pushes	Scrapes	Bumps	E-stops
1	1	1	1	0	0	0	0	0
2	5	5	20	6	2	0	4	3
3	0	0	1	0	11	0	1	6
4	0	0	0	0	1	1	0	0
5	0	0	0	0	6	2	0	6
6	1	0	1	0	1	0	1	1
Average	1.2	1.0	3.8	1.0	3.5	0.5	1.0	2.7
Std Dev	1.9	2.0	7.9	2.4	4.2	0.8	1.5	2.8

Table 3: Participants' subjective assessment.

Likert scale 1 / 5	Joystick	Multi-touch
Hinder / Help	3.7 (1.2)	4.2 (0.4)
Difficult / Easy to learn	4.7 (0.5)	5.0 (0.0)
Difficult / Easy to use	3.5 (1.4)	4.2 (1.2)
Irritating / Pleasant	2.8 (1.3)	3.7 (1.2)
Uncomfortable / Comfortable	3.8 (1.0)	3.83 (1.0)
Inefficient / Efficient	3.3 (1.2)	4.33 (0.8)

era), with variable speeds. As described earlier, the camera movement is controlled by directly touching the main video panel and the robot movement is controlled by touching a movement control panel that looks like a cross between a top-down view of a joystick and a steering wheel. Figure 4 depicts a close-up of this control mechanism.

We then looked for large-scale patterns of movement. We noted that participants tended to work using one of two mental models. One model holds that movement and speed are controlled together: the location of the finger with respect to the origin or center of the movement control panel determines both future direction and speed, with speed increasing with the distance from the center of the panel. We call this the "proportional velocity" model. The other model, which we call the "discrete velocity" model, states that the user expects to control direction of movement independent of speed. There are two major refinements to these models: when participants confine their gestures to a cross-like area consisting of up-down and side-to-side motions (which we term "on-axis" movement) and when they make gestures outside of these x- and y-axes (which we term "off axis" movement). Finally, there were two other classes of movement that occurred sufficiently frequently to warrant their own categories: "trackpad"-type movement and "ring"-type movement. Trackpad movement is reminiscent of how users work with trackpads: with short, repetitive motions. Ring movement occurred along the steering-wheel-like ring that formed the outer circle for the movement control panel.



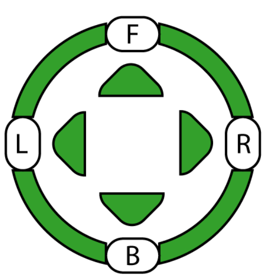


Figure 4: Shoulder (left) and close (right) view of the drive control panel, providing control of translation (vertical) and rotation (horizontal).

Once we identified these patterns, we reviewed the video to broadly characterize each participants' interaction with the DiamondTouch interface (described below). We noted that three participants had largely orthogonal approaches that, when taken together, provided a good cross-section of mental models and responses to the interface. Accordingly, we isolated the data from these participants for further, more detailed analysis. We prepared for the analysis by further refining the models of interaction described in the previous paragraph into categories of actions that were described precisely enough to be able to match instances of participants' gestures against them to code each of their actions.[1] To ensure standardization and 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).

The patterns and coding showed that every participant exposed some base assumption for which we had not accounted in the interface design. Through the detailed post-hoc data analysis, we noticed each user seemed to develop his or her own individual style when interacting with the DiamondTouch interface. In some cases these variations helped the robot move through the course and identify victims. In other cases the style variants did not hinder their performance enough for users to notice a degradation of control or interactivity. Regardless, we noted these "nuggets" of human and robot interactivity and analyzed them qualitatively and quantitatively. To illustrate these user-developed interaction styles, we provide a narrative of each participants' interactions.

Participant 1: This participant had a continuous, flowing movement on the drive control panel. Exclusively using his middle finger on his right hand, he only made use of proportional control. The finger movements can be best described as a continuous line that began at the origin or middle of the panel and then curved smoothly throughout the control panel. The participant appeared to grasp the concepts of proportional and mixed-axis control due to his ability to not only adjust the speed of the robot's movements, but also to mix translation and rotation constructively. Mixed "analog" movements likely indicate an under-

[1]The detailed rules for coding each gesture can be downloaded from the following URL. http://www.cs.uml.edu/∼mmicire/DTCodingRules.pdf

standing that the drive control panel was a direct analogy to the joystick. Besides using proportional movement 100% of the time, 55% of his movements were in off-axis areas of the drive control panel. Interestingly, the participant insisted on using this continuous motion on the camera control panel even though an experimenter explained to him at the beginning of the run that the camera was controlled through discrete grid "buttons" on the panel, as shown in Figure 2. Although the buttons caused discrete camera movement, the user held his fingers to the surface and moved them continuously in the direction of desired motion as he would in a proportional mode control panel. The participant continued this action throughout the run even though the camera provided absolutely no proportional control actions as feedback to the user. Fortunately, this continuous movement did not negatively affect the camera control buttons, so the user did not appear to notice any unanticipated reactions from the robot. This example reinforces the often-referenced design principle of consistency (e.g., see [5]): in this case, that the control panels should all adopt the same movement analogy.

Participant 2: The second participant chose several interaction methods throughout her run and provided the widest variety of unexpected 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 acceleration. Interestingly, 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 with her finger and repeat this action approximately every two seconds. It was only when she rotated her wrist slightly that we realized the incorrect assumption she was making. The participant was attempting to "turn" the outside ring of the control surface like a steering wheel in a automobile. After approximately five minutes, an experimenter explained that the outer ring was not a steering wheel and restated that the robot could be rotated by moving to the left and right components of the control panel. The user acknowledged that she understood and placed her finger correctly on the edge of the control panel to rotate the robot. Rather than holding her finger constantly on the control surface and moving in a constant motion as before, she began tapping the control panel rapidly in the desired direction. This became her preferred mode of interaction for the rest of the run, accounting for 89% of her drive control velocity movements. These "button" movements were sometimes very rapid, exceeding four taps per second in some cases. Strangely, this tapping caused the robot to visibly "bounce" the video image since it was receiving commands to translate, rotate, and brake in rapid succession. The user did not appear to notice this effect. While there was an 89% bias to discrete movements, she showed a relatively small bias to on-axis movement as it accounted for 55% of her movements.

Participant 3: Participant number three began his run with concise and deliberate motions on the drive control panel. He would begin his motions in the center of the movement control panel and then move in a straight line to the desired translation and rotation position. In this respect, he seemed to grasp the proportional control aspects of the interface. Unlike the previous two users, he would lift his finger immediately and restart the motion for every subsequent robot movement. This created an unexpected mix between the proportional velocity control seen in Participant 1 and the discrete control bias seen in Participant 2. After approximately five minutes, the hybrid proportional and discrete finger movement began to resemble a "trackpad" movement that one might make with modern laptop mouse control surfaces. This finger action with a mouse would also be equivalent to directly translating a map in an application such as Google Maps™, or the page surface in Adobe® Acrobat®. In this way, the user seemed to want to "push" or "drag" the robot in the desired direction, but in small steps instead of continuous button presses or proportional control. Similar to Participant 2, this approach created a noticeable bounce in the video display for the end of every finger movement, but the participant did not seem to have a negative reaction to this phenomenon. Even more interestingly, this persistent "trackpad" movement did not manifest itself in any interaction with the camera pan and tilt control. The participant pressed very deliberately at the sides of the video control display and interacted with the video control panel in a way intended by the system designers.

Participant 4: This user appeared to have a very appropriate approach to the drive control panel. The participant would begin in the center of the panel and then proportionally accelerate to the desired speed. It was only through the post-hoc analysis that we noticed a very subtle technique being used. The participant used multiple fingers, much like a piano player, to shorten the amount of area that his fingertips were required to transverse. 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 just lower his index finger. Upon making contact with his index finger on the left side of the drive control panel, he would slowly raise his middle finger and allow drive control to transfer to the index finger. This had the unintended effect of providing very smooth transitions in what would otherwise have been a "button" style non-proportional acceleration. This technique was mixed with standard proportional mode control, although his fingers were switched at seemingly random times. Like Participant 1, this participant insisted on using proportional control of the camera control panel even though discrete control was the only method described by experimenters and demonstrated by the robot's pan-tilt unit.

Participant 5: The participant began the run with proportional acceleration, but after two minutes of the run he began pressing the inner directional triangles exclusively. His interaction with these buttons was a mix of proportional and discrete velocity control, but one interesting effect emerged. 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 at any time during his run. Like Participant 4, he used multiple fingers to activate the drive control panel but maintained very discrete finger contact with the triangular buttons. He did not perform any of the subtle "mixing" of multiple fingers detailed in Participant 4. Although Participant 5 did not take advantage of the proportional control of the drive control, his discrete button presses allowed him to interact with the camera control panel without issue.

Participant 6: Immediately upon starting her run, the sixth participant established a clear style that used only the vertical and horizontal axis. She would begin in the center of the control panel and then quickly and deliberately move to the outer ring, establishing 100% translation or rotation, but only one at a time. The post-hoc analysis confirmed this, as she showed 100% of her movements on axis, 76% ended at the outer ring, and 76% of these were proportional velocity commands. She would regularly switch fingers, although no pattern could be detected. Her hand posture was muscularly tight and she held her non-used fingers high like a pianist or touch typist. Another interesting aspect was her interaction with the camera control. She would only touch the edge of the image even though she had been shown that the pan and tilt control buttons were much larger. In fact, there were many accidental triggers of surrounding panels like ADR mode and the distance panel view. This finding indirectly reinforces the design criterion that the borders between panels should minimally be the width of the participant's fingertips to avoid accidental interference.

6. CONCLUSIONS AND FUTURE WORK

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 modes of input. 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. However, the flexibility and freedom of the interface also presents a problem for the designer. Namely, the designer must carefully choose control methods that give extremely clear affordances and appropriate feedback to the user. 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.

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 a catalyst for many "emergent interactions" that were not planned or 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 tells us that we need to revise the design to better align perceived affordances and actual functionality.

Nevertheless, the results show promise for the interface since little optimization was actually performed during the porting process. In fact, we know that several of the interaction methods that survived the porting process are suboptimal and yet performance was not degraded. This research thus provides a good "stepping off point." We are confident that more can be done to enrich the user experience because we no longer are limited to the constraints of the number of degrees of freedom of a joystick. Because this is a software system, it is easier to iteratively tailor the interaction approach than when using a joystick. This feature strikes a beneficial middle ground between a software and hardware solution for interaction functionality.

Our future work will center around the lessons learned from this experiment as drawn from the interaction characterizations. All of the interesting emergent interactions have helped to clarify mis-assumptions in the design of the interface. Additional functionality not explored in this study such as direct map manipulation and "point to send the robot here" commands should provide for ease of navigation. Once this type of navigation technique is fully explored, extending the interface to multi-agent command and control is a natural progression.

7. ACKNOWLEDGMENTS

The authors wish to thank the experiment participants. We would also like to thank Harold Bufford, 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, and Ann Virts at NIST for the use of their facilities for and support throughout the testing. This research was supported in part by NSF IIS-0415224.

8. REFERENCES

[1] P. Dietz and D. Leigh. Diamondtouch: A multi-user touch technology. In *ACM Symposium on User Interface Software and Technology*, pages 219–226, November 2001.

[2] K. A. Ericsson and H. A. Simon. Verbal reports as data. *Psychological Review*, 87:215–251, 1980.

[3] J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. of the ACM Symposium on User Interface Software and Technology*, 2005.

[4] B. Keyes. *MS Thesis: Evolution of a Telepresence Robot Interface*. Univ. of Massachusetts Lowell, Lowell, MA, 2007.

[5] J. Nielsen. *Usability Engineering*. Academic Press, San Diego, CA, 1993.

[6] D. Norman. *The Design of Everyday Things*. Basic Books, New York, N.Y., 1988.

[7] S. Schipani and E. Messina. Maze hypothesis development in assessing robot performance during teleoperation. *National Institute of Standards and Technology NISTIR 7443*, September 2007.

[8] J. Scholtz, J. Young, J. L. Drury, and H. A. Yanco. Evaluation of human-robot interaction awareness in search and rescue. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA), New Orleans*, April 2004.

[9] B. Shneiderman. Direct manipulation: A step beyond programming languages. *IEEE Computer*, 16:57–69, 1983.

[10] H. A. Yanco and J. L. Drury. "Where Am I?" acquiring situation awareness using a remote robot platform. In *Proc. of the IEEE Conference on Systems, Man and Cybernetics*, October 2004.

[11] H. A. Yanco, B. Keyes, J. L. Drury, C. W. Nielsen, D. A. Few, and D. J. Bruemmer. Evolving interface design for robot search tasks. *To appear in the Journal of Field Robotics*, 2007.