A Clickable World: Behavior Selection Through Pointing and Context for Mobile Manipulation

Hai Nguyen, Advait Jain, Cressel Anderson, Charles C. Kemp

***Abstract*— We present a new behavior selection system for human-robot interaction that maps virtual buttons overlaid on the physical environment to the robot's behaviors, thereby creating a clickable world. The user clicks on a virtual button and activates the associated behavior by briefly illuminating a corresponding 3D location with an off-the-shelf green laser pointer. As we have described in previous work, the robot can detect this click and estimate its 3D location using an omnidirectional camera and a pan/tilt stereo camera. In this paper, we show that the robot can select the appropriate behavior to execute using the 3D location of the click, the context around this 3D location, and its own state. For this work, the robot performs this selection process using a cascade of classifiers.**

We demonstrate the efficacy of this approach with an assistive object-fetching application. Through empirical evaluation, we show that the 3D location of the click, the state of the robot, and the surrounding context is sufficient for the robot to choose the correct behavior from a set of behaviors and perform the following tasks: pick-up a designated object from a floor or table, deliver an object to a designated person, place an object on a designated table, go to a designated location, and touch a designated location with its end effector.

I. Introduction

For assistive robots, being able to correctly decipher user commands would be advantageous for performing useful services. Many methods have been proposed for human-robot interaction but none thus far have been adopted extensively. Interfaces based on the traditional WIMP (windows, icons, menus, pointers) model are often criticized as being an unnatural mode for interaction, while natural interfaces based on speech or gestures are themselves plagued by performance problems in realistic environments. To cope with these difficulties, we present a new human-robot interaction system for which the physical world is viewed as having overlaid virtual buttons that trigger robotic behaviors when clicked by the user.

In general, these virtual buttons can be clicked by providing a 3D location to the robot. For this work, the user clicks these virtual buttons using an uninstrumented laser pointer. As we have previously described in [8], our robot El-E has a laser-pointer interface that detects when a user illuminates a location in the environment and estimates its 3D location. We previously validated this approach in the context of object grasping and a preliminary object-fetching application [10]. Within this paper we generalize this approach to form a clickable world interface and demonstrate its efficacy in the context of a full assistive object-fetching application designed for motor-impaired individuals.

Charles C. Kemp is with the Faculty of Biomedical Engineering at Georgia Tech charlie.kemp@bme.gatech.edu

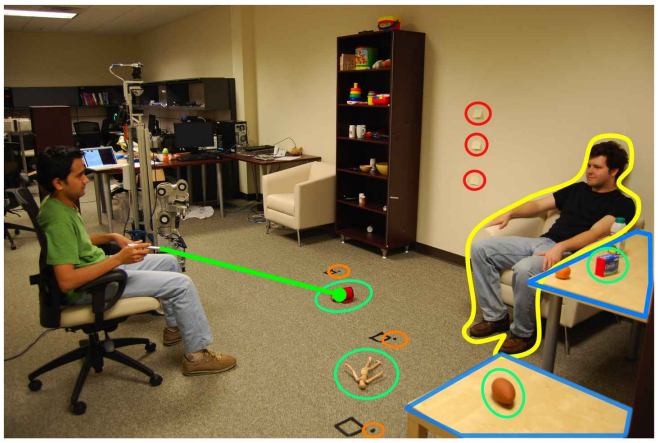


Fig. 1. A clickable world interface enables a user to trigger appropriate robotic behaviors by clicking on virtual buttons using a laser-pointer.

We first discuss the relationship to previous works in Section II. Then, in Sections III and IV, we describe our robot along with details of the clickable world interface as it applies to assistive robots. To evaluate the effectiveness of the system at selecting appropriate behaviors, we present experiments and associated results in Sections V andVI. Finally, we close with concluding remarks.

II. Related Work

Several other examples of intelligent pointing devices exist, such as Patel and Abowd's iCam augmented reality system [12]. In this work, users could virtually annotate an environment using a handheld computer containing a laser pointer, camera, and sensors that determined the computer's position relative to a localization system installed in the environment. The XWand[15] and WorldCursor[14], developed at Microsoft Research allow people to select locations in the environment. The XWand is a wand-like device that enables the user to point at an object in the environment and control it using gestures and voice commands. For example, lights can be turned on and off by pointing at the switch and saying "turn on" or "turn off", a media player can be controlled by pointing at it and giving spoken commands such as "volume up", "play", etc. This work is similar in spirit to ours, the object to be acted upon is selected using the XWand and a simple command specifies what task is to be performed. For our work, having a robot perform tasks avoids the need for specialized, networked, computer-

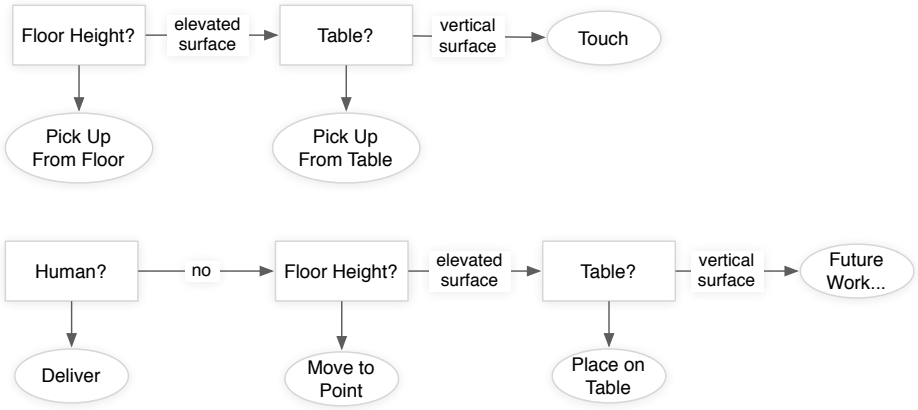


Fig. 2. **Top**: El-E's decision process for mapping from sensory inputs to behaviors to execute. **Bottom**: Corresponding sensory input to behavior mapping in the state where the robot is holding an object.

operated, intelligent devices. A robot also has the potential to interact with any physical interface or object in addition to electronic interfaces. Moreover, unlike these systems our clickable world interface is fully portable with the robot and does not require a model of the environment or any modifications to the environment.

Torralba [13], describes a method of object recognition based on contextual information. Little [9], discusses learning spatial configurations of objects for example cups and plates are typically next to each other on a table. He discusses how this knowledge combined with the shape and appearance of objects can be used for object recognition. He motivates the importance of connecting the spatial and semantic information. Such work in computer vision is relevant to our system because the identity of an object limits the set of robot behaviors that are applicable for that particular object. The motivation behind the clickable world is that the click allows the user to specify where in the world the robot should perform a task. The robot can then use context to infer the task that the user wants the robot to perform.

In robotics, work by Dune [5] [6] describes a visual servoing mechanism for a system that enables users to click on the image of an object in a wide-angle camera and have a camera mounted on a robot arm point at the object. Classic work modeling the mechanisms and structure used for cognition such as ACT-R [2], are related to our work in that these systems can also be used for determining the correct behavior to execute given some input. However, our approach does not attempt to model human level cognition or reasoning.

It is conceivable that a clickable world interface could make use of eye gaze, pointing with the hand, and other natural gestures. There has been extensive research in these areas[16], [4]. Some of these systems are designed with similar objectives to the clickable world interface, but in contrast to the laser-pointer interface, these methods currently do not have the ability to provide a suitably accurate 3D location.

III. Clickable World Framework

In the behavior-based robotics framework [3], robot behaviors can be viewed abstractly as mappings from stimulus, S to to motor responses, R:

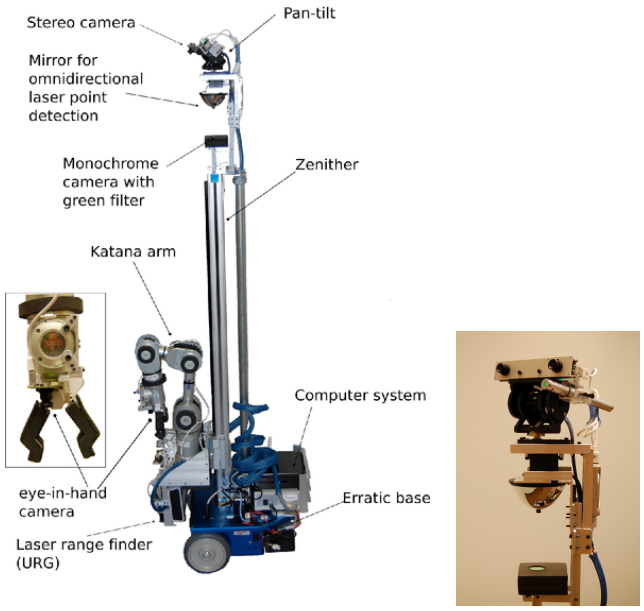
$$\beta : S \mapsto R \tag{1}$$

When there are multiple behaviors or sets of behaviors from which to choose, creating a mapping from stimuli to behaviors can become a challenge. With our clickable world interface, we posit that a location in the world can be a powerful cue for user-directed behavior selection. In our clickable world interface, each 3D location, p, provided by the laser-pointer interface is mapped to a behavior, β_i, executable by the robot:

$$f : p \mapsto \beta_i \tag{2}$$

For the examples we describe in this paper, this 3D location serves two roles. First, the robot uses the 3D location and contextual information around the 3D location to select and execute the appropriate behavior, thereby implementing the mapping of equation 2. Consequently, by giving a 3D location to the robot the user commands the robot to execute a desired behavior. Second, the selected behavior uses the 3D location as a parameter. This 3D location is often critical to the behavior, such as when it tells the robot where to move or where to manipulate. Within our implementation, these two distinct roles of selecting behaviors and providing parameters to behaviors are intertwined. For example, the robot often moves towards a location in order to better assess the surrounding context and thereby distinguish which of several behaviors to execute.

As we demonstrate in this paper, the power of this approach as a user interface derives from the intuitive relationship between a location and a mobile manipulation behavior. When acquiring an object, the location of an object is sufficient to tell the robot which object to pick up. Likewise, when delivering an object, the location for delivery is sufficient to tell the robot where the object should



(a) El-E (b) Camera system

Fig. 3. (a) An image of the entire mobile manipulator with the integrated interface system. (b) The laser pointer interface is integrated into the robot's head. It consists of an omnidirectional camera (bottom half) and a pan/tilt stereo camera (top half).

be delivered and the manner in which it should be delivered. Furthermore, if a user wishes to have the robot manipulate a fixed part of the environment, such as a door handle or a light switch, the location of the manipulable device is sufficient to command the robot to reach out and make contact with it.

Since mobile manipulation activities typically involve task-relevant locations with which the robot makes contact either directly or indirectly, we expect that this type of interface will extend to a wide variety of activities. For example, when acquiring objects the location specifies the object with which the end effector should make contact, and when delivering objects the location specifies where the object held by the robot should make contact.

Within our system, the user selected 3D point is given to a behavior selection mechanism that uses a manually constructed cascade of classifiers to decide which behavior to execute (Figure 2). Each module in this cascade results in a difficult to reverse change in the robot and world state as the robot attempts to collect more information and thereby disambiguate the command.

IV. Implementation

The robot, El-E, is a mobile manipulator with a 5-DoF Neuronics Katana 6M arm mounted on a linear actuator which sits on top of a Videre Erratic mobile base (Figure 3(a)). In addition to its head, which is specially designed for detecting laser pointers (Figure 3(b), and more extensively described in [8]), El-E has a color camera on its end effector and a URG laser range finder on the linear actuator carriage that also contains the manipulator.

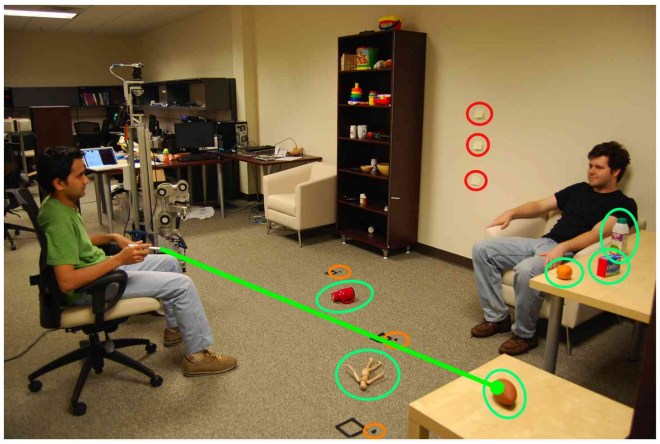


Fig. 4. In the starting configuration, users can select object buttons on the ground and table to be picked up by the robot. Drive to commands can be given by clicking on ground buttons in areas where there are no objects present. In addition, the robot can be instructed to touch a location when the user selects points on vertical surfaces representing buttons on walls.

The specially designed hardware and software system forming the laser-pointer interface enables El-E to detect when a user illuminates a location with a green laser pointer and estimate the 3D location selected by this point and click. Given this 3D location and its immediate context obtained through the robot's sensors, our system activates the correct behavior to carry out desired user commands.

We now describe El-E's decision process after the user clicks a button in the world. The set of behaviors that El-E has to choose from are first determined by the current state of the robot, either the robot is free to grasp an object (*Free_To_Grasp*) or has an object in its gripper (*Object_In_Gripper*).

For each behavior, if the 3D location given is initially further than a threshold distance away, El-E drives towards it but stops before the given point is in manipulable range to request another laser detection. This two step process was created to reduce the error in the estimated 3D location of the designated point since, for stereo ranging, the error in triangulation increases with increasing distance away from the camera.

A. *Robot State* – Free_To_Grasp

In this state the robot does not have an object in its gripper. During this mode users are able to click on the following buttons: objects on the floor, empty locations on the floor, objects on the table and wall locations. When activated, these buttons (illustrated in Figure 4 and 5) trigger the following associated behaviors: *Grasp_On_Floor*, *Follow_Laserpoint*, *Grasp_On_Table*, *Reachout_And_Touch* .

To determine which of the buttons were activated and thus which behaviors to execute, our robot uses the decision process summarized in Figure 2. In more detail, first El-E determines whether the click was on the floor button by checking the height of the laser point. El-E assumes that the floor is flat and that its base sits on the floor. If the height of the laser point is less than a threshold height (currently

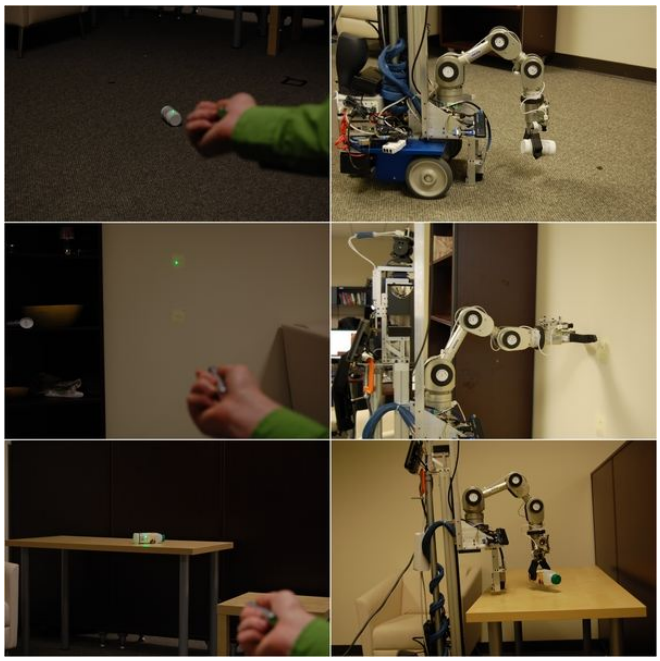


Fig. 5. Each row shows the user clicking a button and the robot selecting the appropriate behavior when the robot is in the *Free_To_Grasp* state. Top to bottom: *Grasp_On_Floor*, *Reachout_And_Touch*, *Grasp_On_Table*

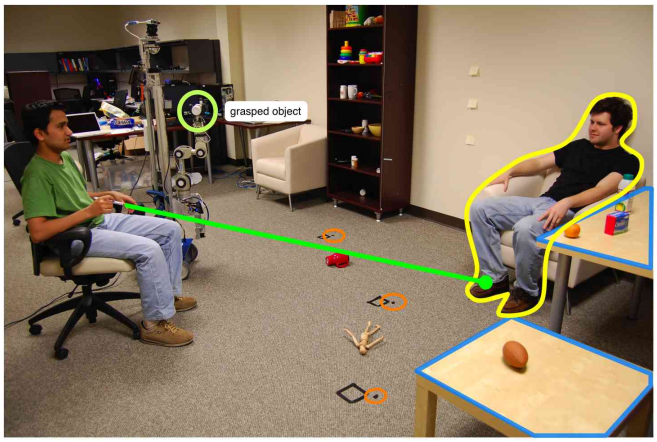


Fig. 6. When El-E has an object in its gripper (green circle), users can select from buttons representing a person (yellow shape), an elevated horizontal surface (blue shape) or location on the floor (orange circles). Selecting a person cause the robot to hand over the object to that person. Selecting elevated surfaces cause the robot to drop off the object. Selecting locations on the floor moves the robot to the laser pointer's position.

30 cm) above the assumed floor height, El-E perceives it as a floor button. It then drives towards the laser point. If there is an object in close vicinity of the click, the robot picks it up. If the human has pointed at an empty location on the floor then the robot moves to the location selected by the user without grasping anything. El-E uses the 3D location of the button and the local context (presence or absence of an object) to select between the *Grasp_On_Floor* and *Follow_Laserpoint* behavior. For the *Follow_Laserpoint* behavior, no further information is necessary so the robot simply drives towards the user indicated location.

If the height of the button is greater than the threshold height (again, 30cm), the robot goes closer to the selected point and then calls a classifier to determine whether the selected button is a horizontal surface (table) or a vertical surface (wall). This classifier works in a manner similar to the table detector described in [10].

The classifier first takes a rectangle of range readings around the user-selected 3D location. It then calculates the differences between horizontal scan lines in this rectangle, finds the maximum difference, and classifies the input as table if this maximum difference is above a threshold. More intuitively, if there is a large difference between two range readings from adjacent heights then our classifier considers that location to be a table as only horizontal surfaces parallel to the scanning plane of the laser range finder would be likely to cause such a sudden change in the amount of free space perceived. The current implementation of the detector classifies a range rectangle as a vertical surface (wall) if it is not classified as a horizontal surface (table).

If the selected button is classified as a table, El-E selects the *Grasp_On_Table* behavior. This behavior uses the laser range finder mounted on a linear actuator to determine the exact height of the table edge and then grasps the selected object. More details on both the *Grasp_On_Floor* and *Grasp_On_Table* behavior can be found in [10].

Finally, if the classifier reports that the clicked button is a wall, the robot executes the *Reachout_And_Touch* behavior. In this mode, the robot drives towards the point, orients itself so that it is perpendicular to the wall and facing the given 3D location. With its manipulator, El-E then reaches out to touch the selected point on the wall. To orient the robot, we perform a least squares line fitting operation to find the line representing the plane of the wall in the ranging information returned by El-E's laser range finders. This behavior serves a plausible precursor for future behaviors such as operating a light switch or opening a door, which would require that the robot reach out and make contact.

B. *Robot State* – Object_In_Gripper

Figure 2 shows the robot's decision process when it has an object in its gripper. The set of buttons which the user can click and their associated behaviors are: directly in front of a person (*Deliver_To_Person*), an empty location on the floor (*Follow_Laserpoint*), and a table (*Deliver_To_Table*).

El-E first determines whether the click should trigger the *Deliver_To_Person* behavior of a human button by first checking for a face in a 3D gravity oriented cylinder around the laser point. To detect the face we used the Viola-Jones frontal face detector as implemented in OpenCV [11] in a manner similar to Edsinger *et al* [7]. If a face is detected then the robot drives towards the person, extends its hand so that the object can be grasped by the person, and releases the object after a preset amount of time.

If a face is not detected, then the robot chooses the *Follow_Laserpoint* behavior for laser point with height less than a threshold (30 cm). Finally, if the height of the laser

Fig. 7. Each row shows the user clicking a button and the robot selecting the appropriate behavior when the robot is in the *Object_In_Gripper* state. Top to bottom: *Deliver_To_Table*, *Deliver_To_Person*

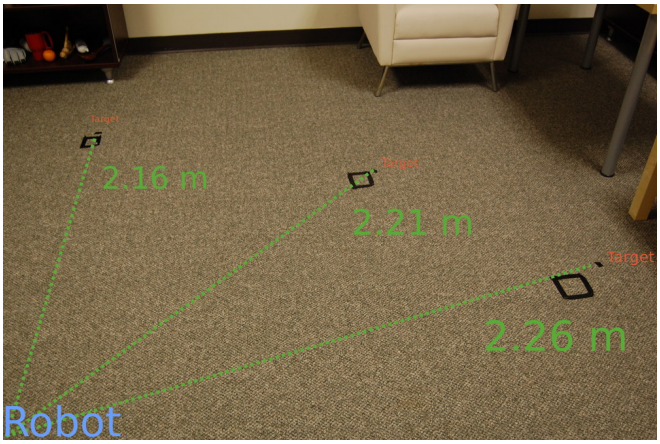


Fig. 8. Floor buttons that the robot is supposed to drive to. Since the robot drives so that the front of it just barely touches the location, we placed the targeting mark outside the square that the robot has to sit on.

point is greater than the given threshold, the robot calls the same classifier as described in the previous sub-section to distinguish between a table and a vertical flat surface. If the user had clicked the table button, the robot selects the *Deliver_To_Table* behavior and places the object on the table.

C. Local context in the decision process

In addition to the 3D coordinate of the click and the state of the robot, the decision processes described in the previous two sub-sections utilize the local context around the click to select the appropriate behavior. In a situation with very few behaviors and buttons such as *Follow_Laserpoint*, only the 3D location of the click may be required. But, as the number of behaviors or the complexity of the button increases, local context information needs to be added to select the correct behavior.

For example, to recognize a click on a human button, El-E looks to see if there is a face in the vicinity of the click. Similarly, a click on an object on the floor is recognized by both the height of the laser point and the presence of an obstacle near the click (detected using a laser range finder which can scan across the surface). Buttons which require additional context information are the vertical surface and the table buttons. The classifier which is used to distinguish between a vertical and horizontal surface requires a three dimensional depth map of a rectangular region around the click. The depth map is obtained by using the laser range finder which is mounted on a linear actuator.

D. Physical Extent of World Buttons

Each button that can be activated by the user in our interface has a physical extent that is determined largely by the classifier that is used to recognize it. The buttons that signify grasping commands (green in Fig. 4), are approximately the size of a circle placed on the floor centered on the object with radius t, where t is a threshold specified by the interface designer. The robot will only pick up an object if the object is within distance t of the 3D location selected by the user. If two object buttons overlap, then the robot will pick up the closest object, which results in a straight edge separating the two buttons much like a Voronoi cell.

With the human buttons (yellow in Fig. 4), the system uses a distance threshold to the closest detected face effectively creating cylindrical buttons with radius t_f. For vertical buttons (red in Fig. 4) and table buttons (blue in Fig. 6), the classifier looks at a square patch from the laser range finder around the selected point. This square patch dictates the effective size of the button.

V. Experiments

To demonstrate that our clickable world interface is able to robustly support a variety of tasks relevant to assistive applications, we tested El-E's ability to select and execute several mobile manipulation behaviors according to user intentions. We conducted our study with 5 members of our lab.

In the first experiment, the robot started out next to the subject with an object in its gripper with both subject and robot facing the same direction. The experimenter would then instruct the subject to command the robot to perform one of its programmed actions: drive to a location, place the object on a table, or hand the object over to a person. The subject was asked to command the robot to perform each action 3 times. Prior to each single command the robot was returned to its starting position.

For the drive to location command, we designated three locations on the floor that the subject could click on. Figure 8 shows the targets. At each location we marked out a square on the floor with tape and affixed another smaller piece of tape nearby. The subject was instructed to click on the smaller tape patch. If the robot base moved toward the small patch and stopped such that it covered the square we considered the command to have been successfully executed.

With the object placement command, the subject was asked to click on either a short or a tall table. The robot was deemed to have successfully carried out this command if it placed the object stably on the selected surface.

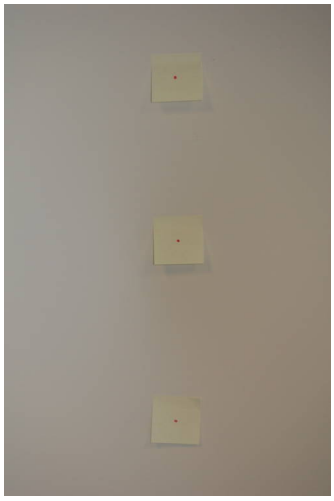


Fig. 9. Wall buttons that the robot is supposed to touch. Success in this task is defined as the robot touching the yellow square (7.7 cm) on which the red targeting dot was placed.

In the human button case, the task was considered successful if El-E handed the object to the selected seated person, such that the person could retrieve the object from El-E without standing up.

The second set of experiments were run similarly to the first experiments. However, El-E started the trials without an object in its gripper. The subject was instructed to select a target Post-it note on the wall opposite to the robot (Figure 9). The robot was considered to have successfully carried out this command if its gripper touched the selected Post-it note.

In the third experiment, the robot performed the object grasping tasks. Each subject was asked to use the robot to fetch two objects, one from the floor and another from a table. After acquiring the object the subject clicked on himself, so that the robot would deliver the object. The trial was considered to be successful if the person successfully obtained the object using the robot.

VI. Results

Task/Subject	A	B	C	D	E	Total Successes	%
Driving	3	2	3	3	3	14	93.3%
Placing	3	3	3	3	0	12	80.0%
Delivery	3	3	3	3	2	14	93.3%
Touching	3	3	3	2	1	12	80.0%

Fig. 10. Results for experiment 1 and 2 testing the world buttons for driving to a desired location, putting objects on elevated surfaces, delivering to a person, and touching. The column for every subject (A-E) reflects the number of successes (out of 3 trials) for each task

Figure 10 summarizes our results for the first and second experiments. For driving to a given location, the only failure resulted from El-E being 5 cm away from the border of the marked target square. When placing an object on table surfaces, the three failures were due, respectively, to the object bouncing off the table surface after being dropped from slightly above the table, a table scan failure, and the robot failing to acquire a second laser pointer detection, probably due to the very shallow angle of incidence with which the user illuminated the table from a long distance.

In the delivery task, there was one failure due to the person's face being just outside the field of view of the stereo camera. Finally, the three failures in the touching task were, in two cases, due to the robot touching a location approximately 1 cm away from the border of the yellow square. In the last case, the robot did not recognize that the point was on a vertical surface due to a misclassification and as a result thought that the user had clicked on a table button.

Task	Total Successes	%
Grasp & deliver object from floor	5/5	100%
Grasp & deliver object from table	5/5	100%

Fig. 11. Results for experiment 3. Each of the 5 subjects commanded the robot to grasp one object from the floor and the table and deliver it back to himself.

In the object fetching experiment, the robot successfully grasped and returned one object from the floor and a table in 10 out of 10 trials as shown in Figure 11.

VII. Conclusion

We have presented a new framework for building applications for mobile manipulation where the user commands the robot by clicking virtual buttons in the environment that map to robot behaviors. We have also shown a practical assistive object-fetching application implemented on a robot and experimental evidence demonstrating the robustness and utility of our approach.

We believe that there are a large number of extensions that can be pursued. In future work, we hope to further demonstrate the applicability of our approach to robots that assist motor-impaired people with activities of daily living. While we have only implemented a stub behavior for touching vertical surfaces, we are interested in adding additional behaviors for operating doors, drawers, cabinets, light switches, and automatic door buttons. Towards this goal, we have already implemented an isolated behavior that opens a door when commanded using the laser-pointer interface [1]. Likewise we have begun work that augments this architecture with speech. Although we use a laser-pointer interface in this work, the framework should be equally applicable to any interface that provides a 3D location to the robot in the robot's frame of reference. We expect that clickable world interfaces could support a diverse array of applications with distinct sets of task-relevant virtual buttons.

VIII. Acknowledgments

We thank Dr. Jonathan Glass for valuable discussions.

REFERENCES

[1] C. C. K. Advait Jain. Behaviors for robust door opening and doorway traversal with a force-sensing mobile manipulator. In *RSS Workshop on Robot Manipulation: Intelligence in Human Environments*, 2008.

[2] J. Anderson. A simple theory of complex cognition. In *American Psychologist*, 1996.

[3] R. Arkin. *Behavior Based Robotics*. MIT Press, Cambridge, Ma, 1998.

[4] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical report, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[5] E. M. Claire Dune, Christophe Leroux. Intuitive human interactive with an arm robot for severely handicapped people - a one click approach. In *IEEE Int. Conf. on Rehabilitation Robotics, ICORR*, 2007.

[6] E. M. Claire Dune, Christophe Leroux. One click focus with eye-in-hand/eye-to-hand cooperation. In *IEEE Int. Conf. on Robotics and Automation, ICRA*, 2007.

[7] A. Edsinger and C. C. Kemp. Human-robot interaction for cooperative manipulation: Handing objects to one another. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2007.

[8] C. C. Kemp, C. D. Anderson, H. Nguyen, A. J. Trevor, and Z. Xu. A point-and-click interface for the real world: Laser designation of objects for mobile manipulation. In *International Conference on Human-Robot Interaction*, 2008.

[9] J. J. Little. Maps, Places, and Worlds for Robots. *International Conference on Robotics and Automation Workshop*, 2007.

[10] H. Nguyen, C. D. Anderson, A. J. Trevor, A. Jain, Z. Xu, and C. C. Kemp. El-e: An assistive robot that fetches objects from flat surfaces. In *Robotic Helpers, Int. Conf. on Human-Robot Interaction*, 2008.

[11] Open source computer vision library: Reference manual, 2001.

[12] G. D. A. Shwetak Patel, Jun Rekimoto. icam: Precise at-a-distance interaction in the physical environment. In *International Conference on Pervasive Computing*, 2006.

[13] A. Torralba. Contextual Priming for Object Detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.

[14] A. Wilson and H. Pham. Pointing in intelligent environments with the worldcursor. *Interact*, 2003.

[15] A. Wilson and S. Shafer. Xwand: Ui for intelligent spaces. *Conference on Human Factors in Computing Systems*, 2003.

[16] J. Ziegler, K. Nickel, and R. Stiefelhagen. Tracking of the articulated upper body on multi-view stereo image sequences. 2006.