25th IEEE International Symposium on
Robot and Human Interactive Communication (RO-MAN)
August 26-31, 2016. Columbia University, NY, USA

Realizing Mixed-Reality Environments with Tablets for Intuitive Human-Robot Collaboration for Object Manipulation Tasks

Jared A. Frank[1], Matthew Moorhead[1], and Vikram Kapila[1]

Abstract— **Although gesture-based input and augmented reality (AR) facilitate intuitive human-robot interactions (HRI), prior implementations have relied on research-grade hardware and software. This paper explores using tablets to render mixed-reality visual environments that support human-robot collaboration for object manipulation. A mobile interface is created on a tablet by integrating real-time vision, 3D graphics, touchscreen interaction, and wireless communication. This mobile interface augments a live video of physical objects in a robot's workspace with corresponding virtual objects that can be manipulated by a user to intuitively command the robot to manipulate the physical objects. By generating the mixed-reality environment on an exocentric view provided by the tablet camera, the interface establishes a common frame of reference for the user and the robot to effectively communicate spatial information for object manipulation. After addressing challenges due to limitations in mobile sensing and computation, the interface is evaluated with participants to examine the performance and user experience with the suggested approach.**

I. Introduction

Even as robotics technologies and applications experience accelerating advances, pervasive adoption and diffusion of robots in society requires development of interfaces that permit non-technical users to effortlessly and intuitively interact with robots. For example, research has shown that use of gestures captured through vision [16] and touchscreens [9] promotes highly interactive experiences in operating robotic platforms. Moreover, AR, the projection of virtual elements onto a real worldview, plays an important role in providing visualizations that can overcome a user's perceptual limitations when collaborating with robots [6]. Specifically, [10] has examined the use of gesture-based interactions with virtual objects to communicate spatial information to a robot about tasks to perform with physical objects. Although the interactive AR techniques are aligned with guidelines for efficient HRI [5], current implementations rely on specialty hardware that can be costly, limited in mobility, and unfamiliar to the general public. Recent advances in mobile technologies allow image processing, multi-touch gesture detection, and 3D virtual graphics rendering all to be integrated in real time. Thus, mobile devices offer capabilities to provide portable interfaces for enhanced HRI. Moreover, with their familiarity and ease of use, mobile devices can support intuitive HRI applications with comparable performance and usability *vis-a-vis* conventional interfaces, for a fraction of the cost and training [15].

This work is supported in part by the National Science Foundation awards RET Site EEC-1132482, GK-12 Fellows DGE: 0741714, and DRK-12 DRL: 1417769, and NY Space Grant Consortium grant 76156-10488.

[1]Mechatronics and Controls Lab, Mechanical and Aerospace Engineering, NYU Tandon School of Engineering, Brooklyn, NY 11201 [jared.alan, mrm678, vkapila]@nyu.edu

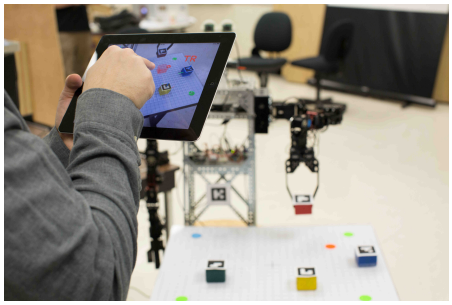

Fig. 1: Proposed environment for human-robot collaboration.

Robots have limited abilities in sensing (e.g., field of view) and cognition (e.g., object recognition). In a controlled setting, such limitations of a robot can be compensated by purposeful design of the robot's environment. For example, a camera fixed in the environment can be used to autonomously drive the robot using visual servoing [13] or to provide users visual feedback for teleoperation [7]. Recently, mobile interfaces have been developed to enable HRI with shared or adjustable autonomy to interact with service robots in real-world scenarios [3], [11]. However, use of robot-mounted cameras in [3], [11] renders an egocentric perspective requiring users to move and change the robot's gaze direction to discover objects in its blind spots, even when being collocated with the robot. Thus, when a robot is moved from a structured environment into a real-world scenario, its perceptual limitations may jeopardize task performance.

In this paper, we present the development of an application providing a mixed-reality graphical environment for users to intuitively interact with a robot for object manipulation (Fig. 1). The environment is immersive for users since it blends their visual space with the working space of the robot, deeply engaging users while allowing them to intuitively communicate spatial commands to the robot. A tablet's back-facing camera captures video of the robot's workspace which is overlaid with virtual objects that are linked to corresponding physical objects in the workspace. To command the robot to manipulate the physical objects, the user manipulates the virtual objects with multi-touch gestures on the tablet screen. The proposed interface approach supports human-robot collaboration for object manipulation tasks by providing a shared space in which the user and robot exchange relevant spatial and task information. Several existing techniques are adapted and integrated on a mobile platform to conduct HRI research. Such an approach necessitated consideration of a set of challenges imposed by the limitations of the

mobile platform's hardware and software, e.g., how to map commands from a 2D interface to a 3D workspace and how to obtain accurate visual measurements in this workspace while maintaining real-time responsiveness of the interface with limited computational resources. After addressing these challenges, a user study was conducted in which participants are tasked to command the robot to pick, place, and stack blocks. The results of the study demonstrate the capabilities of the robot, the user, and the proposed interface approach, all of which can be leveraged to enhance teaching the robot tasks that it may not be able to accomplish alone due to its perceptual limitations. Insights gained are shared as a set of guidelines for developers interested in designing efficient interfaces on mobile platforms for conducting HRI research.

II. System Description

The system used in this study includes a humanoid robotic platform with two 6 degree-of-freedom (DOF) arms, a table with blocks of different shapes and colors, and a tablet device held by the user (Fig. 1). The motors in the robot's arms are driven by a microcontroller. The microcontroller hosts a Wi-Fi module to communicate with the tablet for wireless exchange of information (e.g., user commands, vision-based measurements) that is used to plan and execute object manipulation tasks. The tablet, which runs a mobile application, is held by the user and pointed at the robot and its workspace from an arbitrary perspective. The tablet's back-facing camera captures video of the scene that is used to render an immersive environment on the tablet screen to interact with the robot. Markers affixed to the robot, its workspace, and objects of interest are detected by an image processing routine running on the mobile application. Use and detection of markers permit vision-based spatial measurements that enable the performance of object manipulation tasks by the robot and overlaying of a mixed-reality environment on the video feed. The mixed-reality environment consists of virtual objects linked to corresponding physical objects in the real world. The user may directly manipulate these virtual objects using touch gestures on the tablet screen to issue commands to the robot to manipulate physical objects in the real world.

A. User Interface

This study uses an Apple iPad 2, which has a 9.7 inch (250 mm) screen with a 1024×768 pixel multi-touch display, a 1 GHz dual-core processor, and a 0.7-megapixel back-facing camera. The user interface (Figure 2) has a minimalist design, which employs the entire screen of the tablet to display the exocentric view of the robot and its environment captured by the camera. In the background, the application is split into three processes to: (1) capture and process camera video frames to detect markers, estimate their real-world poses, and establish associated coordinate frames; (2) augment video frames with virtual elements to enhance the user's situational awareness; and (3) capture and map user's multi-touch interactions with virtual elements to generate commands for the robot to manipulate physical objects. Open source libraries are used to perform image processing (OpenCV), rendering of AR content (OpenGL ES), and TCP/IP communication with the robot's controller (CocoaAsyncSocket). The software architecture of the interface is adapted from [4], which focused on enhancing students' interaction with engineering laboratory test-beds.

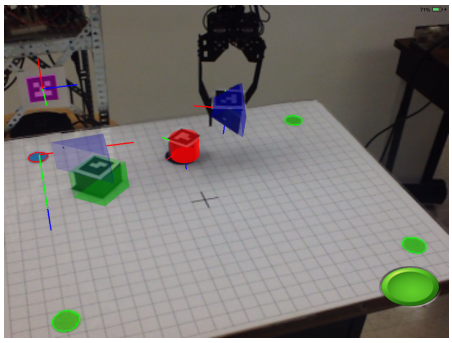

Fig. 2: Screenshot from the user interface of the application.

III. Computer Vision

As the user points the tablet at the robot and its workspace from an arbitrary perspective, the video captured by the back-facing camera is used to extract vision-based estimates of the pose of the robot, the table, and objects of interest. These estimates serve as the foundation of a mixed-reality environment; both to establish a common frame of reference for exchanging spatial information between the user and the robot and to render interactive graphics. The virtual elements in the environment (1) act as stimulating visual aids to enhance the monitoring of tasks performed by the robot and (2) are manipulated by the user to intuitively command the robot to manipulate physical objects. Thus, real-time computer vision techniques play a critical role in allowing users to naturally interact with the robot from the interface. Although markerless techniques exist to detect and estimate the poses of objects, they are currently computationally too expensive to implement in real time on a mobile platform [1]. Thus, in this work, components of the system are affixed with visual markers that can be efficiently detected, recognized, and localized so that the poses of the components may be estimated in real time.

A. Marker Detection

Two types of markers are utilized in this study. The first type are solid colored circular stickers 1 in. (2.54 cm.) in diameter. One blue and three green markers are attached on the surface of the table to form the four corners of a rectangle and to establish a coordinate frame for the robot's workspace. To detect these markers, a color segmentation approach offers the benefits of computational efficiency and simple implementation, as the neon colors are easy to distinguish from the background. For each color of interest, this approach compares each element of a $3 \times w \times h$ matrix to a specified range, where w and h are the width and height of a video frame, respectively. Thus, a technique is designed that employs the minimum number of marker colors to reduce computation time. One blue marker is used at a known corner of the rectangular pattern so that it is uniquely identified in each frame. Then, three green markers are placed at the remaining corners. Since these markers may not be detected

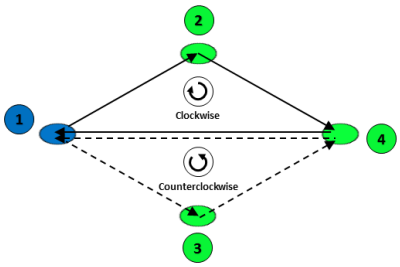

Fig. 3: Diagram of the markers used to establish the workspace coordinate frame, with marker 4 being identified.

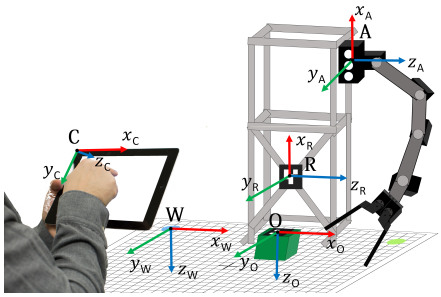

Fig. 4: Coordinate frames used by the interface environment.

in the same order from one frame to another, an algorithm is developed to solve the correspondence problem by performing a test on each marker, which involves separating the marker pattern into two closed triangular loops (as in Fig. 3 to test marker 4). The points that make up the loops are ordered as: the location of the marker being tested, the location of the blue marker, and the location of one of the remaining two green markers. After forming the two loops, their sense is classified as either clockwise (CW) or counterclockwise (CCW) using the signs of cross products computed between the vectors that describe the edges of the triangles in Fig. 3. Then, the senses of the two loops associated with each marker uniquely identify the marker, since the same markers will always have two CCW loops, two CW loops, and one CW and one CCW loop.

Once the colored markers have been located and uniquely identified, the application detects a second type of marker, 3×3 resolution 2D barcodes attached to the robot and to each of the objects of interest. These markers provide the mobile interface with accurate position and orientation information in a compact size, and allow for robust detection from noisy images in near constant time. Moreover, when using barcode-based markers, there is little chance of mistaking one marker for another [2]. To detect these markers in each video frame, a procedure outlined in [1] is performed that consists of producing a binary image from the frame using an adaptive threshold, detecting contours in the image, extracting potential markers from the detected contours, removing the perspective projection of the marker candidates, and reading and decoding the marker codes to identify each marker.

B. 3D Pose Estimation

Once all markers have been detected in an image, their 3D poses are estimated so that the locations and orientations of the robot, table, and objects on the table are known relative to the coordinate frame of the device camera. The estimates are computed using knowledge of the intrinsic parameters of the camera and four 2D-3D point correspondences [1]. Figure 4 shows a diagram of the human-robot system with the coordinate frames used by the interface to express the poses of objects in the scene. To represent the pose of a coordinate frame M with respect to another coordinate frame N, the homogeneous transformation matrix $T_{\mathrm{M}}^{\mathrm{N}}$ is used [14]. After the pose of the workspace coordinate frame is estimated with respect to the camera frame $T_{\mathrm{W}}^{\mathrm{C}}$, the relative poses of the objects, $T_{\mathrm{O}}^{\mathrm{C}}$, and robot, $T_{\mathrm{R}}^{\mathrm{C}}$, are also estimated.

IV. Mixed-Reality Interaction

The estimated poses of coordinate frames attached to the robot, the workspace, and various objects of interest are used both to render virtual elements on the screen for augmenting the live video of the scene as well as to map user gestures on the touchscreen to appropriate spatial commands for the robot. This link between the extrasensory visualizations afforded by AR and the fluid interactivity provided by the touchscreen allows for interactive experiences with the mixed-reality interface. On the tablet screen, live video from the back-facing camera is shown at 30 frames per second. Projected onto this view are virtual coordinate axes registered in the scene on top of the markers attached to the table, robot, and objects. In addition, virtual objects are displayed in the mixed-reality environment, whose locations, orientations, and colors are linked to the objects detected by the interface.

Although interactions on the touchscreen are 2D in nature, they can be mapped to locations and orientations in 3D space due to the physical constraints of the application investigated in this study (where objects with the same height can only be placed on a flat surface or on top of another object). Thus, the robot's environment is treated as a series of parallel planes that the interface navigates between depending on the actions of the user. To begin, all objects are assumed to be located on the surface of the table. When a user taps on the touchscreen, the coordinates of the tap are converted from screen coordinates to image coordinates through a simple resolution conversion. Then, the inverse of the transformation $T_{\mathrm{W}}^{\mathrm{C}}$, obtained by pose estimation, is used to map the tapped location in the image to a location in a plane parallel to the table. The interface compares this mapped location to the known locations of the objects to determine whether an object has been selected, and transforms the pose of the object until it has been represented with respect to the coordinate frame attached to the robot arm $T_{\mathrm{O}}^{\mathrm{A}} = T_{\mathrm{R}}^{\mathrm{A}} T_{\mathrm{W}}^{\mathrm{R}} T_{\mathrm{O}}^{\mathrm{W}}$, where $T_{\mathrm{R}}^{\mathrm{A}}$ is the pose of the frame established at a point on the robot with respect to a coordinate frame at the robot's shoulder, $T_{\mathrm{W}}^{\mathrm{R}} = T_{\mathrm{W}}^{\mathrm{C}} (T_{\mathrm{R}}^{\mathrm{C}})^{-1}$ is the pose of the frame established at the workspace with respect to the robot coordinate frame, and $T_{\mathrm{O}}^{\mathrm{W}} = T_{\mathrm{O}}^{\mathrm{C}} (T_{\mathrm{W}}^{\mathrm{C}})^{-1}$ is the pose of the frame established on the object with respect to the workspace coordinate frame. Spatial relations are sent to the robot so that it may position and orient its tool to pick up the object.

After an object is selected, the user drags and rotates her fingers on the screen to choose a new location and orientation

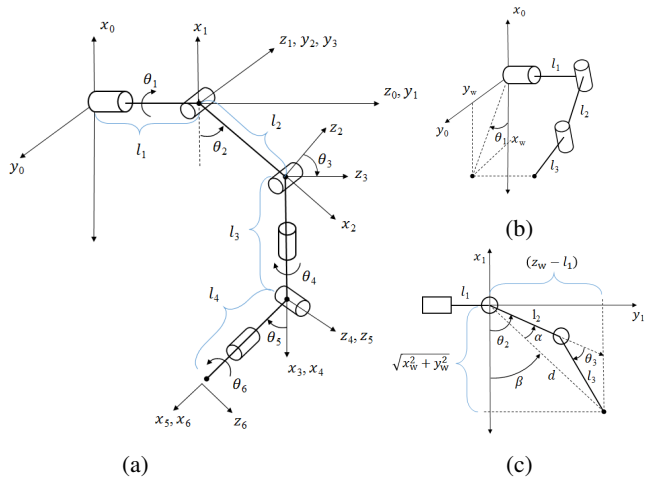

Fig. 5: Model of (a) the left arm of the robotic platform and its projections onto (b) $x_0 - y_0$ and (c) $x_1 - y_1$.

for the object as the virtual object in the scene moves and rotates under her fingers. This semi-transparent virtual object represents the set point for the system and shows the user where the object would be located and how it would be oriented after the robot executes its commands. A virtual grid projected in the scene on the surface of the table provides a visual reference as the user manipulates virtual objects. Upon detecting the release of the user's finger from the screen, the interface checks to see if the virtual object collides with another object in the scene. If a collision is detected, the interface attempts to stack the object by iteratively shifting up to the next plane and performing the collision check until the lowest allowable height the object can be placed at is determined. When the object has been successfully placed in the interface, the corresponding location in the workspace is sent to the robot for object placement. The resulting interface provides users a shared interactive space with the robot through which they can directly alter the physical world.

V. Robotic Platform

The robotic platform used in this study to manipulate objects on a table is a humanoid with two 6-DOF arms. Commands received from the interface contain desired poses for the tool to pick-up, place, or stack an object. An inverse kinematic model computes the required angles to orient the arm's joints such that its tool is brought to desired poses and a planner generates a sequence of waypoints for driving the tool along a path to complete the manipulation task.

A. Arm Kinematics Model

The inverse kinematic models of the robot's arms are found by constraining the tool to be oriented downward in a vertical plane and decoupling the 6-DOF problem into two 3-DOF problems, i.e., the inverse position kinematics and the inverse orientation kinematics [14]. The solution to the inverse position kinematics problem gives the joint angles to position the wrist center of the arm and the solution to the inverse orientation kinematics problem gives the joint angles to orient the tool. To solve the inverse position kinematics problem we apply a geometric approach as outlined in [14]. By projecting the arm onto the $x_0 - y_0$ plane (see Fig. 5b), we can solve for θ_1 in terms of the fixed coordinate frame attached to the shoulder of the arm. Next, the arm is projected onto the $x_1 - y_1$ plane (see Fig. 5c) so that θ_3 is found. The joint angle θ_3 has both a positive and a negative solution, corresponding to elbow configurations associated with the left arm and right arm of the robot, respectively. Knowledge of θ_3 and the location of the wrist center with respect to the shoulder frame, (x_w, y_w, z_w), can then be used to solve for θ_2. The constraint on the orientation of the tool, R_6^0, is used to calculate its orientation with respect to the wrist frame

$$(R_3^0)^{-1} R_6^0 = \begin{bmatrix} c_1 c_{2-3} & s_{2-3} & s_1 c_{2-3} \\ s_1 & 0 & -c_1 \\ -c_1 s_{2-3} & c_{2-3} & -s_1 s_{2-3} \end{bmatrix}, \quad (1)$$

where $c_i \triangleq \cos(\theta_i)$ and $s_i \triangleq \sin(\theta_i)$. Moreover, the orientation of the tool with respect to the wrist frame is calculated using cascaded rotation matrices

$$R_6^3 = \begin{bmatrix} c_5 & s_5 c_6 & -s_5 s_6 \\ -c_4 s_5 & c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 \\ -s_4 s_5 & s_4 c_5 c_6 + c_4 s_6 & -s_4 c_5 s_6 + c_4 c_6 \end{bmatrix}. \quad (2)$$

Knowing θ_1, θ_2, and θ_3, the elements of the transformations (1) and (2) are compared to yield three equations for three unknowns θ_4, θ_5, and θ_6. By inspection θ_5 has a positive and a negative solution, resulting in two sets of solutions for θ_4 and θ_6 depending on the sign of θ_5. According to the model, the tool needs to approach the object from the top with the tool facing downward, which requires $\theta_5 < 0$. This yields the following solution for θ_4 and θ_6

$$\theta_4 = \tan^{-1}\left(\frac{s_4 s_5}{c_4 s_5}\right) = \tan^{-1}\left(\frac{-c_1 s_{2-3}}{s_1}\right), \quad (3)$$

$$\theta_6 = \tan^{-1}\left(\frac{-s_6 s_5}{c_6 s_5}\right) = \tan^{-1}\left(\frac{s_1 c_{2-3}}{s_{2-3}}\right). \quad (4)$$

B. Path Planning

After using the inverse kinematic model to compute the joint angles required to position and orient the tool to pick up and place an object on the table, a path is planned between these locations. To avoid collisions with objects, the arm moves in a horizontal plane at a safe height above the table, descending to pick-up, place, or stack objects. First, a path is decomposed into three segments. Two guarded motions are planned near the object's pick-up and placement locations, where the robot must descend and rise slowly to prevent any sharp collisions with possible obstacles. Then, a free motion is planned that drives the tool from a point above the pick-up location to the point at the same height above the placement location, connecting the two guarded motions [14]. During this free motion, the robot may be driven quickly above the table, since there are no expected obstacles along these path segments. To plan the free motion, the path is a straight line discretized into equally spaced waypoints. Before descending towards the table to pick up and place the object, the robot uses the current orientation of the object from vision measurements on the tablet and the desired orientation of the object from user interaction, respectively, to properly align its

wrist. Although only straight line paths are considered in this study, future work will investigate how the visual information collected by the mobile device may be integrated with data from the robot's sensors to plan trajectories around obstacles.

VI. Evaluation Results

For the proposed HRI environment to be deemed intuitive, users lacking experience with robots and familiarity with the interface must demonstrate an ability to effectively use the interface to interact with the robot, and have an enjoyable experience doing so. To investigate the performance and user experience associated with the interface to perform object manipulation, $N = 40$ participants (aged 19-21, 31 male, 9 female) with no prior experience with the interface are asked to collaborate with the robot to accomplish three challenges: placing a red circular block onto a circular area drawn on the table with known location (22cm, 6cm), placing a green square block onto an "ideal" square area drawn on the table with known location (8cm, 10cm) and orientation (45°), and finally stacking a blue triangle block on top of the square block (Fig. 2). These challenges are designed to build on one another and allow the fundamental interactions enabled by the interface to be studied. Although the robot contains cameras, they are deactivated to illustrate how visual information from the device camera may be leveraged to accomplish tasks. The more general scenario in which the visual information from both the robot's sensors and the device camera are fused is beyond the scope of this paper and will be considered in a future study. While the tablet is held in place using a mount during experiments, to allow participants to interact comfortably with both hands, it can be moved around freely during the interaction. After participants complete the first challenge, which acts as a trial, their block placement during the second challenge and stacking success during the third challenge are each recorded. To observe the intuitiveness of the interface, users are given minimal instruction on how to use it. See an illustrative video at [12].

A. Performance Results

To assess the effectiveness with which the mixed-reality environment provides collaboration with the robot for performing object manipulation tasks, the performance of both the user interaction as well as the robotic manipulation are investigated during the second and third challenges. As each participant places and orients the square block during the second challenge, the commands generated by interacting with the environment are monitored by the interface and the location and orientation of the square block are measured after it has been placed on the table by the robot. This data allows tests to be performed that indicate the ability of the interface to allow participants to accurately communicate intended spatial information to the robot and the ability of the robot to accurately execute intended spatial commands (i.e., pick and place blocks at commanded poses on the table).

The mean *commanded* pose for the square block is ($M_x = 7.985$cm, $SD_x = 0.188$cm), ($M_y = 10.063$cm, $SD_y = 0.289$cm), and ($M_\theta = 45.282°$, $SD_\theta = 1.499°$). Two-tailed t-tests show no statistically significant difference between the *commanded* and the *ideal* poses of the square block: $[t_x(39) = -0.5453,\ p_x = 0.5886,\ CI_x = (7.9258, 8.0427)]$, $[t_y(39) = 1.3921,\ p_y = 0.1718,\ CI_y = (9.9711, 10.1564)]$, and $[t_\theta(39) = 1.2049,\ p_\theta = 0.2355,\ CI_\theta = (44.8059, 45.7661)]$. Thus, participants can use the interface to accurately communicate intended spatial commands to the robot.

The mean difference between the *commanded* and the *measured* poses of the square block is ($M_x = 0.100$cm, $SD_x = 0.685$cm), ($M_y = -0.175$cm, $SD_y = 0.747$cm), and ($M_\theta = -0.23655°$, $SD_\theta = 2.494°$). Two-tailed t-tests show no statistically significant difference between the commanded and the measured poses of the block: $[t_x(39) = 0.9183,\ p_x = 0.3641,\ CI_x = (-0.3187, 0.1197)]$, $[t_y(39) = 1.3967,\ p_y = 0.1704,\ CI_y = (-0.0739, 0.4039)]$, and $[t_\theta(39) = 1.0595,\ p_\theta = 0.2959,\ CI_\theta = (-0.3798, 1.2153)]$. Thus the robot can accurately position and orient a block to the pose commanded from the interface.

The mean *measured* pose of the square block is ($M_x = 8.085$cm, $SD_x = 0.626$cm), ($M_y = 9.888$cm, $SD_y = 0.717$cm), and ($M_\theta = 45.045°$, $SD_\theta = 2.216°$). Two-tailed t-tests show no statistically significant difference between the *measured* and the *ideal* poses of the block: $[t_x(39) = 0.8467,\ p_x = 0.4023,\ CI_x = (7.8837, 8.2838)]$, $[t_y(39) = 0.8853,\ p_y = 0.3814,\ CI_y = (9.6674, 10.1301)]$, and $[t_\theta(39) = 0.3808,\ p_\theta = 0.7054,\ CI_\theta = (44.1685, 45.568)]$. Thus, error associated with user interactions and block placement is sufficiently small to permit effective collaboration with the robot.

As participants complete the third challenge, the success of the block stacking operation is recorded. Out of 40 participants, 33 (82.5%) successfully stacked the triangle block on top of the square block on their first attempt. This challenge was not easy, as small deviations in the visual measurements would have resulted in many failed stacking efforts. The performance results demonstrate the feasibility of exploiting computer vision techniques directly on a mobile platform to support communication of accurate spatial information to a robot. The proposed design allows users, with little to no training, to leverage their mobile devices to produce commands that are sufficiently accurate to enable a sensorless robot to perform precise object manipulations.

B. User Experience Results

To assess aspects of HRI experience provided by the interface, participants are asked to respond to a balanced set of 11 positive and negative statements [8] (see below) on a 5-point scale (1: strong disagreement and 5: strong agreement).

a. It was difficult to interact with the virtual blocks.
b. The virtual graphics were useful visual aids.
c. The interface made interacting with the robot easy & fun.
d. I required assistance using the interface.
e. It took a long time to get comfortable with the interface.
f. It was easy to place blocks using the interface.
g. It was easy to orient blocks using the interface.
h. It was easy to stack blocks using the interface.
i. Overall, I felt that I was able to use the interface to accurately communicate my intentions to the robot.
j. Overall, I would recommend this application to people who work with robots at home or as part of their job.

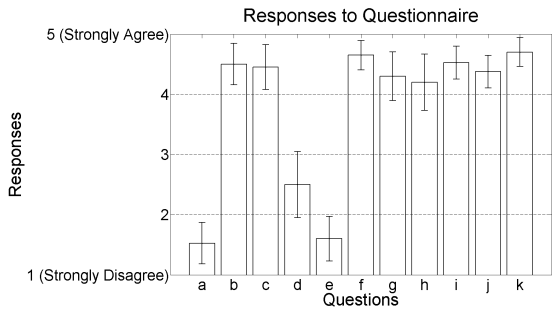

Fig. 6: Responses to a user experience questionnaire.

k. I would like to see more applications like this marketed to people who may have robots at home or at their job.

Participants' responses to the questionnaire are summarized in Fig. 6. They find it easy and comfortable to interact with the interface for collaboratively performing object manipulation tasks with the robot. These results can be explained by the fact that as owners of mobile devices participants are experienced in interacting with smart device touchscreen using a set of previously learned gestures. Thus, developers can leverage gestures from popular apps when designing efficient mobile interfaces for interacting with robots. Although some participants require a small amount of time and assistance from researchers at first, they report that the interface allows them to collaborate well with the robot, and that the onscreen virtual graphics serve as useful visual aids in performing tasks. Thus, for applications in which users are located near the robot's workspace, developers can design more efficient interfaces by providing users a "window" into a shared space with the robot. By projecting in this space interactive virtual objects, which take on the properties of real objects of interest detected by the interface, developers can provide a mechanism for intuitive and direct manipulation of the world, leaving the robot and interface transparent, a desired element of efficient interaction [5]. To lower the cognitive burden on users, the mobile environment is designed without any navigation bars or button layouts. For example, rather than using a button layout to control the height of the plane wherein the user is interacting, the interface discerns the intent of users to stack blocks automatically when users drag the virtual blocks near to each other. Such strategies free users from the need to know the mechanisms behind the interface, thus reducing the amount of mental models required to command the robot effectively. While some interactive elements may be necessary to navigate interfaces for complex HRI applications, developers are encouraged to minimize them to draw attention away from the interface. Finally, from responses and comments left at the end of the questionnaire, participants are excited about using and recommending mobile applications like the one developed in this study to interact with robots in the future.

VII. Conclusions and Future Work

This paper presented a tablet-based mobile application that uses live video from the tablet's back-facing camera to render a mixed-reality environment that users can interact with to intuitively collaborate with a humanoid robot to perform object manipulations. Using touchscreen interaction and AR feedback, the interface allows users to directly manipulate virtual objects to command the robot to manipulate physical objects. Results of a study with participants reveal that the proposed environment allows users to effectively collaborate with the robot in pick, place, and stack tasks at locations that are removed or obstructed from the robot's sensing. Responses to a questionnaire show that participants feel their experiences were easy, intuitive, and beneficial. These results confirm the ability of mobile interfaces to provide sufficient precision while maintaining a required degree of real-time responsiveness for positive user experiences. While designing such interactions on a mobile platform, a number of unique challenges are imposed by the limitations of the device hardware and software. In addition to sharing these challenges, strategies to address them are provided. Future work will expand the interactions offered by this novel class of interfaces and employ techniques such as markerless object detection to extend studies to real-world scenarios.

References

[1] D. Baggio *et al*, *Mastering OpenCV with Practical Computer Vision Projects*. Packt Publishing Ltd, 2012.

[2] L. Belussi and N. Hirata, "Fast QR code detection in arbitrarily acquired images," in *Proc. SIBGRAPI Conf. Graphics, Patterns, and Images*, 2011, pp. 281–288.

[3] P. Birkenkampf, D. Leidner, and C. Borst, "A knowledge-driven shared autonomy human-robot interface for tablet computers," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 152–159.

[4] J. Frank and V. Kapila, "Interactive mobile interface with augmented reality for learning digital control concepts," in *Proc. Indian Control Conf.*, 2016, pp. 85–92.

[5] M. Goodrich and D. Olsen Jr, "Seven principles of efficient human robot interaction," in *IEEE Int. Conf. Systems, Man, and Cybernetics*, vol. 4, 2003, pp. 3942–3948.

[6] S. Green *et al*, "Human-robot collaboration: A literature review and augmented reality approach in design," *Int. Journal of Advanced Robotic Systems*, vol. 5, pp. 1–18, 2008.

[7] S. Hashimoto *et al*, "Touchme: An augmented reality based remote robot manipulation," in *Proc. Int. Conf. Artificial Reality and Telexistence*, 2011.

[8] N. Malhotra, "Questionnaire design and scale development," in *The Handbook of Marketing Research: Uses, Misuses, and Future Advances*. Sage Publications, 2006.

[9] M. Micire *et al*, "Analysis of natural gestures for controlling robot teams on multi-touch tabletop surfaces," in *Proc. ACM Int. Conf. Interactive Tabletops and Surfaces*, 2009, pp. 41–48.

[10] P. Milgram *et al*, "Applications of augmented reality for human-robot communication," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 3, 1993, pp. 1467–1472.

[11] S. Muszynski, J. Stückler, and S. Behnke, "Adjustable autonomy for mobile teleoperation of personal service robots," in *Proc. IEEE Int. Symp. Robot and Human Interactive Communication*, 2012, pp. 933–940.

[12] NYU. (2016) Realizing mixed-reality environment for human-robot collaboration for object manipulation. [Online]. Available: http://engineering.nyu.edu/mechatronics/videos/mrmanipulation.html.

[13] A. Saxena, J. Driemeyer, and A. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[14] M. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. John Wiley and Sons, 2006.

[15] Y.-H. Su, C.-C. Hsiao, and K.-Y. Young, "Manipulation system design for industrial robot manipulators based on tablet PC," in *Intelligent Robotics and Applications*, 2015, pp. 27–36.

[16] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, no. 2, pp. 151–173, 2000.