2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

An Augmented Reality Interface for Human-Robot Interaction in Unconstrained Environments

Sonia Mary Chacko[1] and Vikram Kapila[1]

Abstract— **As robots start to become ubiquitous in the personal workspace, it is necessary to have simple and intuitive interfaces to interact with them. In this paper, we propose an augmented reality (AR) interface for human-robot interaction (HRI) in a shared working environment. By fusing marker-based and markerless AR technologies, a mobile AR interface is created that enables a smartphone to detect planar surfaces and localize a manipulator robot in its working environment while obviating the need for a controlled or constrained environment. The AR interface and robot manipulator are integrated to render a system that enables users to perform pick-and-place task effortlessly. Specifically, a smartphone-based AR application is developed that allows a user to select any location within the robot's workspace by merely touching on the smartphone screen. Virtual objects, rendered at user-selected locations, are used to determine the pick and place locations of objects in the real world. The virtual object's start and end points, originally specified in the smartphone camera coordinate frame, are transformed into the robot coordinate frame for the robot manipulator to autonomously perform the assigned task. A user study is conducted with participants to evaluate the system performance and user experience. The results show that the proposed AR interface is user-friendly and intuitive to operate the robot, and it allows users to communicate their intentions through the virtual object easily.**

I. INTRODUCTION

Augmented reality (AR) technology is gaining broad acceptance and is being widely used for varied domains and applications, including education, gaming, entertainment, tourism, medicine, military, manufacturing, navigation, and robotics, among others [1], [2]. As the AR technology continues to advance rapidly, it is drawing increasing attention for its potential applications in the field of robotics. Specifically, recent advancements in the AR technology and built-in AR capabilities of smartphones have opened new opportunities in the design and development of natural and intuitive mechanisms for human-robot interaction (HRI). In prior research, many researchers have explored the applicability of AR to multiple domains of robotics. For example, in a pioneering application of AR to robotics, [3] has used overlaid virtual stereographic to control a teleoperated robot manipulator. KUKA, a robot manufacturer, has explored the use of AR for user training, programming and operation, as well as service and maintenance of industrial robots [4]. To represent the state of a robot through different gestures and emotions, [5] featured an augmented virtual character that is rendered on the real robot. To interact with and control a multi-robot system, [6] developed a mobile mixed-reality approach by leveraging the visual and inertial sensing of mobile devices. A hand-held AR device to teach sequential tasks to a household robot through photographs is described in [7]. Using the exTouch interface of [8], users can change the position and orientation of an omnidirectional mobile robot through multi-touch gestures or by physically moving the mobile touch screen in relation to the robot. A drag and drop touch screen interface serves as the AR interface of [9] for instructing a mobile robot to deliver an object. With the TouchMe interface of [10], users can manipulate each part of a teleoperated robot by touching and dragging the computer graphic models overlaid on the robot. These prior research studies illustrate that HRI can be enhanced with AR-based interfaces.

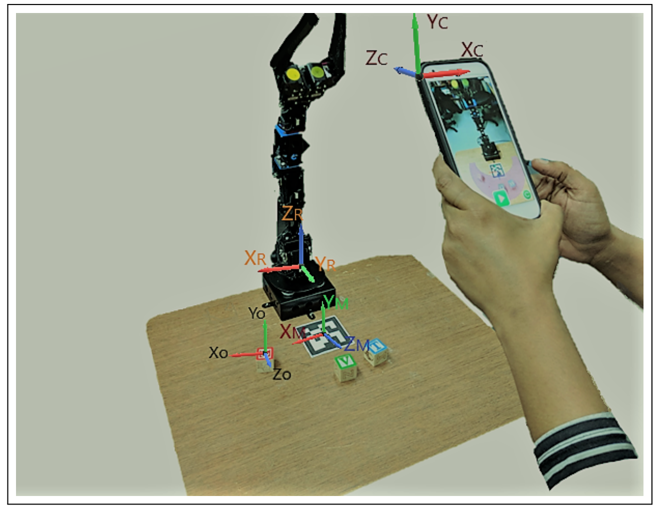


Fig. 1: The HRI system: a robot manipulator and a smartphone hosting the AR application

Building on the aforementioned AR-based techniques for HRI, this paper proposes and demonstrates a novel method of applying the AR technology to create a simple and intuitive interface for HRI in an unstructured working environment. A smartphone-based AR interface is developed to intuitively instruct a robot to pick and place objects from any desired location in its workspace. By leveraging the capabilities of the AR and smartphone technologies, we propose an innovative and user-friendly method to communicate the user intent to the robot. In the proposed method, the smartphone replaces both the traditional image capturing camera sensor and the user interface hardware. For example, the mobile AR

Work supported in part by the National Science Foundation under DRK-12 grant DRL-1417769, ITEST grant DRL-1614085, and RET Site grant EEC-1542286, and NY Space Grant Consortium grant 48240-7887.

[1]Mechatronics, Controls, and Robotics Laboratory (MCRL), Mechanical and Aerospace Engineering, NYU Tandon School of Engineering, Brooklyn, NY 11201, USA [sonia.mary, vkapila]@nyu.edu

interface provides a real-time view of the robot's workspace through the smartphone camera, thus eliminating the need for installing fixed cameras in the environment or on the robot.

Many prior research works on AR in robotics are based on traditional fiducial markers to track the robot, its workspace or objects in the environment [6], [8], [10], [11]. For example, [8] utilizes an image based AR recognition technology (Vuforia) to detect a mobile robot affixed with an image marker and to superimpose virtual graphics on the robot image. Moreover, multiple fiducial markers are used in [10] to track the robot and render the computer graphics model over it. Similarly, in [11], multiple markers are used to define the robot workspace and to transform device screen coordinates to workspace coordinate frame by using projective transformation. Furthermore, objects to be manipulated are required to be known *a priori* or affixed with markers to detect their positions in the workspace [6], [11], [12]. That is, a key limitation of such prior approaches is their reliance on affixing markers on the robot, the workspace, and the objects of interest, thus restricting practical applications in a real-world, unconstrained environment. In the current work, we have combined the marker-based and markerless AR technologies to obviate the reliance on multiple markers to define the robot's working environment. A single fiducial marker is used to localize the pose of the robot and define its workspace. Markerless AR technology is used to detect a planar surface that contains the robot workspace. Moreover, the markerless AR technology enables placement of virtual anchors in user-selected locations on the detected plane and determination of coordinates of physical objects at the corresponding locations (by a raycast to user-selected points in the detected plane). Thus, this approach is suitable for manipulating unknown and randomly placed objects in the robot workspace. The minimum prerequisite (a single marker at a known location in the robot coordinate frame) for setting up the robot work environment and the capability of markerless AR technology to detect almost all plane surfaces, make this interface easily adaptable to other robots and environments, thereby providing a generic solution for HRI.

From a usability perspective, the AR interface proposed herein reduces cognitive load on and effort of the user as there is no interaction with the virtual objects or a virtual robot, such as dragging, moving, or manipulating to a desired pose, unlike in some prior works [8], [9], [10]. Specifically, the methods described in [8] and [10] are based on interacting with a virtual robot that is rendered over a real robot to accomplish tasks. In a similar vein, many prior researchers have followed human-virtual robot interactive methods wherein a virtual object resembling a robot is rendered as a medium to interact with the real robot for performing manipulation tasks or training tool motions [13], [14]. The user interfaces proposed in the aforementioned prior works are specific to a particular robot. In contrast, the AR interface of current work is independent of the robotic platform, and it is based on creating virtual objects on the fly to mediate user intention, i.e., communicate task-specific information to the robot. Specifically, when a user selects the target object's start and goal locations, augmented virtual objects are rendered on the corresponding locations. Moreover, such location information is communicated to the robot and rendered as visual feedback for the user about the intended locations for the pick-and-place task.

Although several researchers have used advanced and expensive head-mounted displays for AR applications (see, [12], [15]), a smartphone constitutes a more convenient tool to operate a robot considering its prevalence in modern society and workplace. Moreover, the popularity of smartphone applications and familiarity with touch-screen interaction make the smartphone an ideal platform for ordinary people to interact with robots by using a mobile AR application. We created our mobile AR interface by using Google's latest ARCore technology [16]. This AR technology enables a smartphone to sense its environment and continuously track a *single* fiducial marker, affixed in the vicinity of the robot, and the position of the smartphone. The fiducial marker's pose relative to the robot base is known and it is used to calibrate the robot's pose. A user can choose any target location in the workspace by a simple touch on the smartphone screen. The screen coordinate selected by the user is sensed and coordinate transformations are used to determine the corresponding workspace position relative to the robot. The rest of the paper is organized as follows. Section 2 describes the methodology and approach behind the design and development of the proposed AR system and gives details about the integration of AR interface with the robot system. Section 3 reports on the experiments conducted to evaluate the system performance and its results. Section 4 concludes the paper by summarizing the work and future plans.

II. AR SYSTEM DESCRIPTION

The system used in this work is comprised of a smartphone device, a four degree of freedom (DOF) robot manipulator, and a desktop computer for processing the data transmitted by the smartphone. As seen in Figure 1, the robot manipulator is mounted on a table and a single marker is affixed in the vicinity of the robot with known (X, Y) coordinates in the robot coordinate frame. Using the hand-held smartphone device, the user can see the robot's workspace, the robot, and the surrounding environment as shown in Figure 2. The user communicates pick-and-place instructions to the robot through touchscreen interaction on the smartphone device. All relevant information from the smartphone is communicated using Bluetooth (BT) communication to the desktop computer, where it is processed and sent to the robot for performing commanded tasks.

A. AR framework

Recent advances in hardware and software systems for smartphone platforms have unleashed exciting innovations in mobile AR. Multiple development platforms and frameworks are readily available to support the smooth and rapid development of mobile AR applications. For this paper, Unity3D,

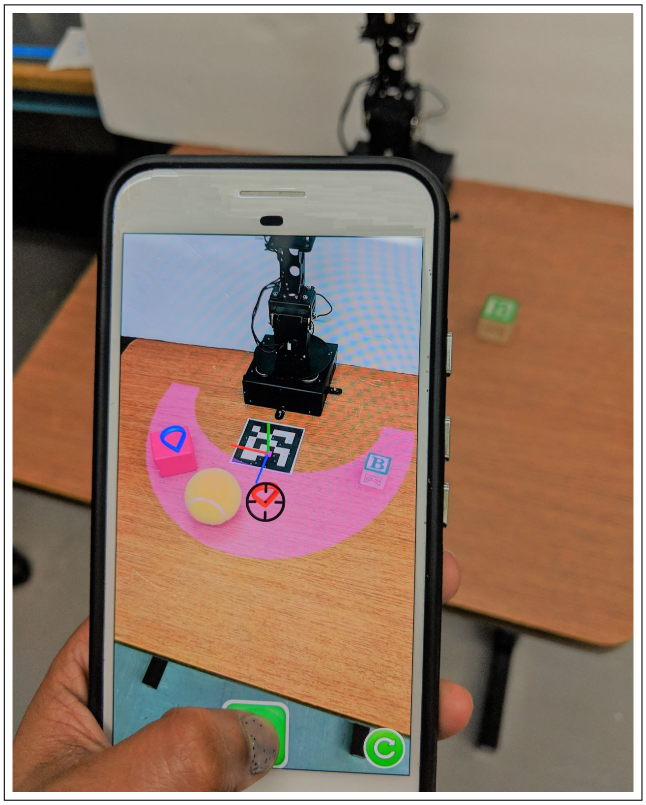


Fig. 2: User selects an object and its goal location

a cross-platform game development engine, is used to build the proposed AR application by incorporating the ARCore library.

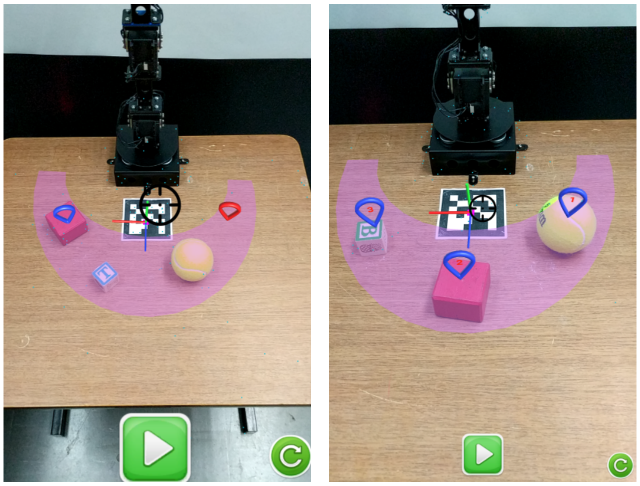
Google's ARCore [16] is a popular augmented reality framework that supports the development of both marker-based and markerless AR applications. By using a set of 2D reference images, ARCore can detect, track, and estimate their physical locations in the 3D world space. The key capabilities of its markerless AR technology, such as motion tracking and environmental understanding, enable ARCore to integrate virtual content with the real world as seen through the phone's camera. Specifically, ARCore's motion tracking feature allows the smartphone to understand and track its position relative to the world by combining visually distinct feature points and measurements from the inertial measurement unit (IMU) on-board the smartphone by using a technique called concurrent odometry and mapping. Moreover, ARCore's environmental understanding feature allows the smartphone to detect the size and location of planar surfaces by looking for clusters of feature points that are common to flat surfaces. The combination of the aforementioned capabilities of ARCore enables a smartphone to sense the environment and build its understanding of the world around it.

AR interface implementation: The augmented reality application is developed and deployed on a smartphone of following specifications: Google Pixel XL with 5.5 inch screen, 1440×2560 pixel resolution, 12.3 Mega Pixel rear camera, and Quad-core CPU (2×2.15 GHz Kryo and 2×1.6 GHz Kryo). To enable a simple and intuitive AR-based HRI interface, as shown in Figure 2, an application with just two buttons and a cross-hair is designed. The main button at the bottom center is for choosing the target location in the workspace. A second button at the bottom right corner is for resetting the selected target location and simultaneously clearing the virtual object created there. The cross-hair is integrated in the application to help the user easily aim at a target location.

Using the rear-facing camera, the smartphone captures images that are processed to identify feature points in the environment. Once the AR application is initiated, the starting location of the device becomes the origin in the world space where the camera coordinate frame $\{\mathcal{F}^{\mathrm{C}}\}$ is attached. The AR application concurrently searches for the known fiducial marker and flat surfaces in the environment. Once the marker is detected, the application begins to track it continuously by placing an anchor on the corresponding position. The marker coordinate frame $\{\mathcal{F}^{\mathrm{M}}\}$ is attached to the anchor. Once tracked, the application provides estimates for the position, orientation, and physical size of the marker. The marker's dimension and pose allow localization of the robot in the world space.

After detecting the fiducial marker and a flat table-top surface, the AR application waits for commands from the user. Now, the user can select any object lying in the robot's workspace by merely moving the smartphone to align the cross-hair on the center of the screen with the desired object and by touching the main button. Since the AR application continuously tracks the location of the smartphone, the user is free to move around in the shared space while giving instructions to the robot. When the user taps the main button, a ray is cast from the center of the cross-hair to the detected flat surface, and an anchor is created at the hit pose on the flat surface. Finally, a virtual object is instantiated at the hit pose as shown in Figure 2. This AR mechanism provides visual feedback to the user to verify that the anchors are instantiated at the intended positions. The AR application estimates and tracks the virtual target's location relative to the smartphone. It then transmits this data to the desktop computer, which in turn commands the robot to manipulate the object. The combination of the aforementioned capabilities of ARCore enables a smartphone to sense the environment and build its understanding of the world around it. Besides, the AR interface displays the reachable 2D workspace of the robot by rendering a pink cross-sectional plot overlaying the workspace, which informs the user about the reachability of the robot in its planar workspace (see Figures 2 and 3). Even as the mobile AR interface is used to communicate user intent to the robot, the interface communicates to the user as well through visual cues such as workspace reachability (shown by the augmented workspace), tool initial and final locations (by virtual objects), etc.

Two variants of the AR interface have been developed as shown in Figure 3. With the first variant, a user can give instructions to pick and place one object at a time by selecting its start and goal locations. In contrast, with the second



(a) (b)

Fig. 3: Two variants of the AR interface. (a) Stores pick and place locations of a single object, (b) Stores multiple objects' locations at once. See video at: http://engineering.nyu.edu/mechatronics/videos/AR-HRI.html

variant, the user can select and store start and goal locations of multiple targets and send them to the robot at once. In this second variant, the user can choose objects in any desired order, and a virtual object with a number is rendered over the image of the real object, indicating the order of its selection (Figure 3b). Once the robot receives coordinates of selected objects, it follows the same order to pick the objects and place them at a predefined location, mimicking object placement on a conveyor belt or on an object delivery system. Such a technique can be useful in assembly-related tasks to select and place individual subassembly components in a prescribed order from arbitrary locations.

B. Coordinate transformation

The AR application estimates the coordinates of the reference marker and virtual objects in a left-handed coordinate system (LHS), since the Unity3D uses LHS. As seen in Figure 1, in the LHS, the X_{C} axis points right, Y_{C} axis points up, and Z_{C} axis points into the smartphone screen. Each virtual object's target coordinates are first transformed from the camera coordinate frame $\{\mathcal{F}^{\mathrm{C}}\}$ to the marker coordinate frame $\{\mathcal{F}^{\mathrm{M}}\}$, which is also expressed in the LHS. Next, the resulting coordinates are transformed to a right-handed coordinate system (RHS), since the robot coordinate system is in RHS. Finally, these coordinates are transformed to the robot's coordinate frame $\{\mathcal{F}^{\mathrm{R}}\}$. The detailed steps are described below.

The pose of the reference marker and object location, measured relative to the origin of the camera coordinate frame ($P_{\mathrm{M/C}}$ and $P_{\mathrm{O/C}}$, respectively) are expressed in the camera coordinate frame (as ${}^{\mathrm{C}}P_{\mathrm{M/C}}$ and ${}^{\mathrm{C}}P_{\mathrm{O/C}}$, respectively) and used to estimate the location of the object relative to the robot (i.e., $P_{\mathrm{O/R}}$) expressed in the robot coordinate frame $\{\mathcal{F}^{\mathrm{R}}\}$ (i.e., ${}^{\mathrm{R}}P_{\mathrm{O/R}}$). To do so, the marker and object locations, measured with respect to the camera coordinates in the camera coordinate frame, are converted into the marker coordinate frame. The marker position (${}^{\mathrm{C}}P_{\mathrm{M/C}}$), it's Euler angles (α, β, γ) and the object position (${}^{\mathrm{C}}P_{\mathrm{O/C}}$) in the camera coordinate frame are transmitted by the AR application to the desktop computer. This data is used to calculate the object position relative to the marker (i.e., $P_{\mathrm{O/M}}$) in the reference marker's coordinate frame (i.e., ${}^{\mathrm{M}}P_{\mathrm{O/M}}$). Now, the object's coordinates, which are expressed in a LHS, need to be converted into a RHS ($({}^{\mathrm{M}}P_{\mathrm{O/M}})_{\mathrm{RHS}}$) by rotating it about the X axis by a $90°$ angle and then swapping the resulting Y and Z coordinates. The reference marker is placed at a fixed distance from the robot with known coordinates in the robot frame $(0, Y_{\mathrm{dist}}, 0)$. The location of object in the robot's coordinate frame is computed by translating this new object position $({}^{\mathrm{M}}P_{\mathrm{O/M}})_{\mathrm{RHS}}$ to the robot's coordinate frame. See below (1)–(6) for details of various transformations

$$
\begin{aligned}
{}^{\mathrm{C}}P_{\mathrm{O/M}} &= {}^{\mathrm{C}}P_{\mathrm{O/C}} - {}^{\mathrm{C}}P_{\mathrm{M/C}}, && (1)\\
{}^{\mathrm{M}}P_{\mathrm{O/M}} &= R_{\mathrm{C}}^{\mathrm{M}}\left({}^{\mathrm{C}}P_{\mathrm{O/C}} - {}^{\mathrm{C}}P_{\mathrm{M/C}}\right), && (2)\\
{}^{\mathrm{M}}P_{\mathrm{O/M}} &= (R_{\mathrm{M}}^{\mathrm{C}})^{\mathrm{T}}\left({}^{\mathrm{C}}P_{\mathrm{O/C}} - {}^{\mathrm{C}}P_{\mathrm{M/C}}\right), && (3)\\
Q &= (R_{\mathrm{X}}(90°))({}^{\mathrm{M}}P_{\mathrm{O/M}}), && (4)\\
({}^{\mathrm{M}}P_{\mathrm{O/M}})_{\mathrm{RHS}} &= EQ, && (5)\\
{}^{\mathrm{R}}P_{\mathrm{O/R}} &= ({}^{\mathrm{M}}P_{\mathrm{O/M}})_{\mathrm{RHS}} + [0, Y_{\mathrm{dist}}, 0]^{\mathrm{T}}, && (6)
\end{aligned}
$$

where the rotation matrix from the marker coordinate frame to the camera coordinate frame $R_{\mathrm{M}}^{\mathrm{C}}$ is obtained by using the marker coordinate frame's Euler angles relative to the camera coordinate frame as below [17]

$$
R_{\mathrm{M}}^{\mathrm{C}} = \begin{bmatrix} \mathrm{c}_\alpha \mathrm{c}_\beta & -\mathrm{s}_\alpha \mathrm{c}_\gamma + \mathrm{c}_\alpha \mathrm{s}_\beta \mathrm{s}_\gamma & \mathrm{s}_\alpha \mathrm{s}_\gamma + \mathrm{c}_\alpha \mathrm{s}_\beta \mathrm{c}_\gamma \\ \mathrm{s}_\alpha \mathrm{c}_\beta & \mathrm{c}_\alpha \mathrm{c}_\gamma + \mathrm{s}_\alpha \mathrm{s}_\beta \mathrm{s}_\gamma & -\mathrm{c}_\alpha \mathrm{s}_\gamma + \mathrm{s}_\alpha \mathrm{s}_\beta \mathrm{c}_\gamma \\ -\mathrm{s}_\beta & \mathrm{c}_\beta \mathrm{s}_\gamma & \mathrm{c}_\beta \mathrm{c}_\gamma \end{bmatrix},
$$

with $\mathrm{s}_\psi \triangleq \sin(\psi)$ and $\mathrm{c}_\psi \triangleq \cos(\psi)$. Moreover, $R_{\mathrm{X}}(90°)$ is a rotation operator that performs the rotation about the X axis by $90°$ [17]. Finally, the elementary row operator E to swap rows two and three of position vector Q is given by

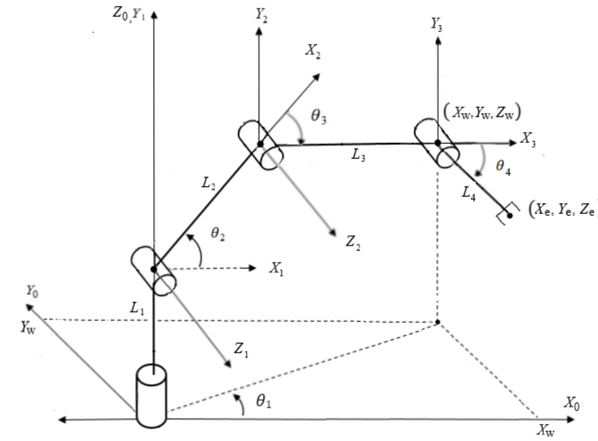
$$
E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.
$$

C. Robot System

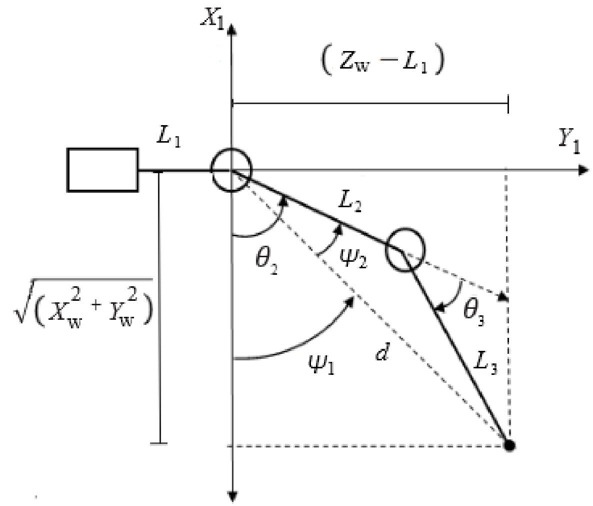
The robot manipulator of this work is a four DOF robot manipulator with a one DOF gripper. Each joint of the robot manipulator is actuated by a Dynamixel servomotor. The rotation of the base is restricted to 0°–180°. The gripper is always held in a downward orientation for picking and placing objects from the workspace. The smartphone transmits objects' target locations (pick and place locations) to the desktop computer through BT communication. The Dynamixel SDK running on the desktop computer receives this data and computes the manipulator's joint angles by using inverse kinematic computations as described below. A USB2AX interface is utilized to enable the desktop computer to directly control the Dynamixel servos.

Inverse kinematics computation: A kinematic model is constructed for the four DOF manipulator with link lengths given by base $L_1 = 115$ mm, shoulder $L_2 = 155$mm, elbow $L_3 = 145$ mm, and wrist $L_4 = 190$ mm. The inverse kinematics problem solves for joint angles $q = [\theta_1, \theta_2, \theta_3, \theta_4]^{\mathrm{T}}$ (see Figure 4) using the geometric method described in [18]. We begin by considering that the gripper is always held in a downward orientation with a fixed wrist angle ϕ. The base angle (θ_1) is solved by projecting the robot manipulator onto the $X_0 - Y_0$ plane, provided that both the X and Y coordinates of the end-effector are not zero. To find the shoulder (θ_2) and elbow (θ_3) angles, given the base rotation angle (θ_1), we consider the plane formed by the shoulder-elbow link as shown in Figure 4. The computation of the elbow angle (θ_3) yields both positive and negative solutions. The negative solution is retained considering the downward orientation of the gripper. The shoulder joint angle (θ_2) is calculated using the elbow angle (θ_3) and the wrist position $(X_{\mathrm{w}}, Y_{\mathrm{w}}, Z_{\mathrm{w}})$. The wrist angle ($\theta_4$) is computed from θ_1, θ_2, and θ_3. Once the gripper grasps the object, it moves slightly upward to avoid collisions in the workspace before moving along a straight path to the target location.

The robot manipulator used in our experiments has a two-fingered gripper with a maximum spread of 24 cm and a depth of 12 cm. The objects for the experiments were selected within the range permissible by the robot's gripper. A fixed closing gap with tolerance is considered to accommodate several objects of 3–6.5 cm width (e.g., a small cube, a big cube, a tennis ball, etc.) for performing the pick-and-place task. The picking height is fixed to 3 cm for all the objects assuming that the objects are placed on the same plane. We developed two programs for the robot system to test both variants of the AR applications. In the first program, the robot expects to receive pick and place locations of a single object sent by the smartphone, while in the second program, the robot receives picking locations of multiple objects and a predefined location is used to place the objects.



(a)



(b)

Fig. 4: (a) A four DOF manipulator model and (b) projection onto the X_1-Y_1 plane

III. System Evaluation and discussion

The proposed AR interface for robot manipulator is tested for its usability and performance with 34 participants with little to no knowledge of such interfaces. The participants included 14 females and 20 males and they ranged in age from under 18 to 64 years (below 18: 6, 19–24: 9, 25–34: 14, 35–44: 2, 45–54: 2, and 55-64: 1). Each user is provided with a short demonstration on using the AR interface and two AR applications are introduced to them. The first application entails picking and placing a single object at a time while the second application requires pick-and-place interaction with multiple objects. The two applications involve a total of three tasks to test usability and performance of the system.

In the first task, the user is asked to physically place a block at a random location in the workspace of the robot. Having selected the initial placement of the block, the user employs the AR interface to communicate with the robot the block's pickup location and its desired drop-off location, both within the robot's workspace. The purpose of this task is to validate whether the user is able to pick and place objects within the robot's reachable workspace with the aid of the augmented rendering of the workspace. In performing the first task, 31 participants were successful in their first attempt while the remaining three participants completed it successfully in their second attempt. This first task addresses questions 4 and 5 in a usability questionnaire detailed below. Next, in the second task, the user is asked to pick an object placed at a random location in the robot's workspace and place it at a *marked* fixed location ((-8, 21) cm from the base of the robot). This task tests the accuracy with which the user is able to communicate their pick-and-place intentions to the robot. Moreover, the second task verifies robot's ability to accurately execute the user's commands. In performing the second task, all 34 participants experienced success in the first trial itself. This second task addresses questions 6–8 in the usability questionnaire below. In the third task, the user selects the location of three objects in any desired order as shown in Figure 3b. The app stores these locations in the same order and then transmits them to the robot for picking the selected objects at the user-specified locations one by one and placing them at a fixed location. Task 3 tests the AR interface's capability to give multiple commands simultaneously for picking multiple objects from different locations. Similar to the second task, all 34 participants experienced success in the first trial itself in doing the third task.

A. System performance

The performance of the AR interface is evaluated by conducting one-factor analysis of variance (ANOVA) tests on the X and Y coordinate values obtained from the second task mentioned above (with $(x_i = -8, y_i = 21)$ cm). The mean commanded position of the object is $(\mu_x = -8.025, \mu_y = 21.037)$ cm with standard deviation $(\sigma_x = 0.178, \sigma_y = 0.347)$. The mean measured position of object is $(\mu_x = -8.038, \mu_y = 21.088)$ cm with standard deviation $(\sigma_x = 0.183, \sigma_y = 0.312)$. The results of ANOVA $(\alpha = 0.05)$ tests indicates that there are no significant differences between the ideal, commanded, and measured values of X coordinates $(F(2, 99) = 0.596, p = 0.55 > 0.05)$ as well as Y coordinates $(F(2, 99) = 0.908, p = 0.41 > 0.05)$, indicating that there are no significant differences between the positions being compared. This illustrates that the AR interface yields an acceptable performance.

B. User experience

The user experience is assessed with the aid of users' responses to a usability questionnaire, the NASA Task Load Index (TLX) [19], and open-ended feedback. First, to evaluate the user experience, inspired by [20], the following post-study system usability questionnaire was developed and administered. The usability questionnaire contains 10 questions, of which seven are positive and three are negative, with a 5-point Likert response scale (1-strongly disagree and 5-strongly agree).

1) The user interface was simple, intuitive, and fun to use.
2) It took a long time to get familiar with this user interface.
3) I needed assistance of a technical person to interact with the robot manipulator.
4) I could easily select the object's pick and place locations using this interface.
5) The virtual graphics helped me understand the workspace of the robot manipulator.
6) The robot picked the objects from my intended locations without fail.
7) The robot placed the objects at my intended locations without fail.
8) It was difficult to pick and place objects from anywhere in the robot workspace using this system.
9) Overall, I felt that I was able to use this user interface to clearly communicate my intentions to the robot manipulator.
10) Overall, I recommend this user interface to people of different age groups to interact with robots.

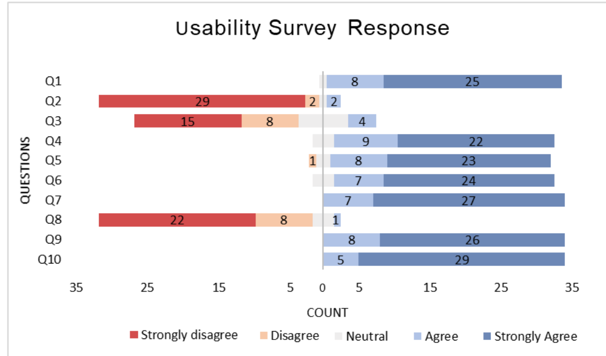


Fig. 5: Response of participants to usability questionnaire

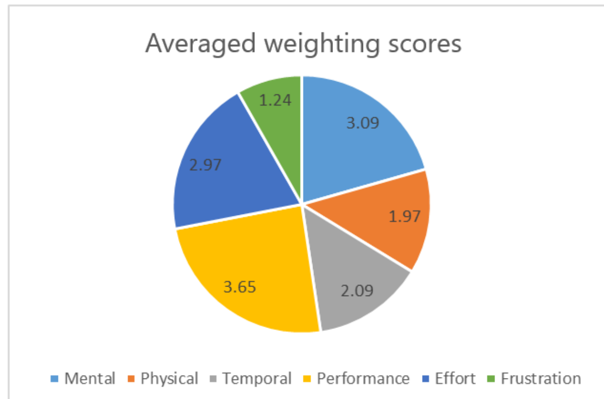


Fig. 6: Averaged weighting score of participants' response to TLX sub-scales

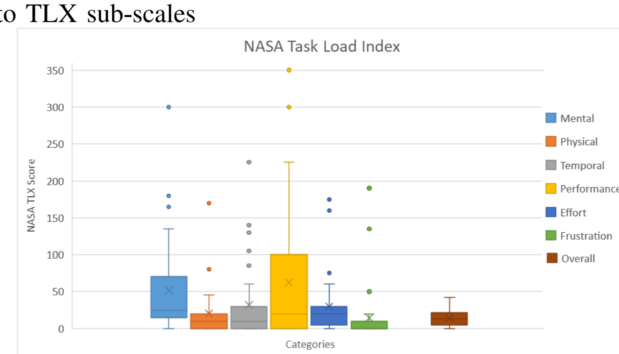


Fig. 7: Mean value of participants' response to TLX sub-scales and overall TLX)

Figure 5 shows the response of participants to the usability questionnaire. The result indicates that the participants were satisfied with the AR interface and highly recommended it for HRI. Specifically, on each of the seven positively directed questions, more than 91% of the respondents reported favorable opinions (agreed or strongly agreed). Similarly, on each of the three negatively directed questions, more than 67% respondents reported favorable opinions (disagreed or strongly disagreed). Most of the participants commented about the simplicity of the AR interface, which allowed them to use it without any assistance.

To further analyze the user experience, we asked the participants to complete the NASA TLX to evaluate the subjective workload for using the interface. The overall workload score is calculated based on the ratings on six sub-scales: mental, physical, and temporal demands and performance, effort, and frustration. In the first part of NASA TLX, the participants marked each of the sub-scale using a 100-point scale. Next, they compared the sub-scales pairwise based on their perceived importance. In Figure 6, the pie chart shows the average weighting scores for each sub-scale, which reports that performance is the most selected subscale while frustration is the least. This indicates that the majority of the participants were satisfied with the performance of the system, and experienced minimum frustration. Figure 7 shows the average task load score of each sub-scale category and overall TLX reported by the participants. The box plot indicates that each of the sub-scales contributes a relatively low score, thereby producing a low overall workload. Among the sub-scales, mental demand and performance contributed significantly higher scores, while the frustration and physical demand contributed relatively lower scores. The result reveals that the participants perceived more mental workload and comparatively low physical demand and frustration during the task. We also observed some outliers (shown as dots in Figure 7), which may have occurred due to the age diversity of users who participated in the experiments.

IV. Conclusion and Future Work

In this paper, a novel AR user interface to operate a robot manipulator is presented. The user response to NASA TLX instrument shows that our proposed system requires a minimum overall workload (on average 14 on a scale of 100). The usability questionnaire indicates that even novice users are able to interact with the robot using this interface easily. The proposed AR interface was integrated and demonstrated with a four DOF manipulator, and it can be easily integrated with various types of robot manipulators. The AR interface approach of this paper requires minimum cost and effort for its adaptation into existing robotic environments and reduces many additional requirements such as fixed overhead camera, specialized hardware, multiple markers, etc. Moreover, it makes it feasible to operate robots in homes, offices, and even outdoor environments. In an industrial setting, using the AR interface of this paper, an operator can command the robot to pick and place objects from any random locations in its workspace, thus allowing flexible changes in the pick-and-place pattern without having to reprogram the robot.

The current AR interface does not include object detection feature since the ARCore technology presently does not support the recognition of 3D objects. Moreover, limitations in memory and computational power of hand-held devices restrict the simultaneous execution of object recognition algorithms along with the high computational processing requirement of AR technology. Thus, even though the present method does not explicitly identify the shape or orientation of an object, it is still useful in many cases where objects are of simple shapes or manipulators have special grippers, such as suction or vacuum grippers, which can handle objects of arbitrary shapes. In future work, we will investigate alternative methods and AR technologies to support 3D object recognition. The present work can be additionally extended by considering obstacles and occlusions in the workspace. Another enhancement may consider path planning by defining waypoints between target locations in a 3D workspace.

Acknowledgment

The authors thank summer interns Andy Zou and Marco Nocito for their support in testing the system. The authors also thank Mr. Derek Paolina for helpful feedback and comments.

References

[1] A. Dey, M. Billinghurst, R. W Lindeman, and J Swan, "A Systematic Review of 10 Years of Augmented Reality Usability Studies: 2005 to 2014," *Frontiers in Robotics and AI*, vol. 5, Article 37, 2018.

[2] M. Mekni, A. Lemieux, "Augmented Reality: Applications, Challenges and Future Trends," *Appl. Comput. Sci.*, pp. 205-214, 2014.

[3] P. Milgram, S. Zhai, D. Drascic, J. Grodski, "Applications of Augmented Reality for Human-Robot Communication," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 3, pp. 1467–1472, 1993.

[4] R. Bischoff and A. Kazi, "Perspectives on Augmented Reality based Human-Robot Interaction with Industrial Robots," *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, pp. 3226–31, 2004.

[5] M. Dragone, T. Holz, and G.M.P. O'Hare, "Using Mixed Reality Agents as Social Interfaces for Robots," *Proc. IEEE Int. Symp. Robot and Human Interactive Communication* (*RO-MAN*), pp. 1161–1166, 2007.

[6] J.A. Frank, S.P. Krishnamoorthy, and V. Kapila, "Toward Mobile Mixed-Reality Interaction With Multi-Robot Systems," *IEEE Robotics and Automation Letters*, vol. 2, pp. 1901–1908, 2017.

[7] R. Fung, S. Hashimoto, M. Inami, and T. Igarashi, "An Augmented Reality System for Teaching Sequential Tasks to A Household Robot," *Proc. IEEE Int. Symp. Robot and Human Interactive Communication* (*RO-MAN*), pp. 282–287, 2011.

[8] S. Kasahara, R. Niiyama, V. Heun, and H. Ishii. "exTouch: Spatially-Aware Embodied Manipulation of Actuated Objects Mediated by Augmented Reality," *Proc. ACM Int. Conf. Tangible, Embedded, and Embodied Interaction*, pp. 223–228. 2013.

[9] K. Ishii, Y. Takeoka, M. Inami, and T. Igarashi. "Drag-and-Drop Interface for Registration-Free Object Delivery," *Proc. IEEE Int. Symp. Robot and Human Interactive Communication* (*RO-MAN*), pp. 228–233, 2010.

[10] H. Sunao, I. Akihiko, I. Masahiko, and I. Takeo, "TouchMe: An Augmented, Reality Based Remote Robot Manipulation," *Proc. Int. Conf. Artificial Reality and Telexistence* (*ICAT*), 2011.

[11] J.A. Frank, M. Moorhead, and V. Kapila, "Realizing Mixed-Reality Environments with Tablets for Intuitive Human-Robot Collaboration for Object Manipulation Tasks," *Proc. IEEE Int. Symp. Robot and Human Interactive Communication* (*RO-MAN*), pp. 302–307, 2016.

[12] S. Blankemeyer, R. Wiemann, L. Posniak, C. Pregizer, and A. Raatz, "Intuitive Robot Programming using Augmented Reality," *Proc. CIRP Conf. on Assembly Technologies and Systems* (*CATS*), pp. 1–6, 2018.

[13] H.C Fang, S. K. Ong, and A.Y.C Nee, "Novel AR-based Interface for Human-Robot Interaction and Visualization,"*Advances in Manufacturing*, vol. 2, pp. 275–288, 2014.

[14] K. Guerin, S. Reidel, J. Bohren, G. D. Hager, "Adjutant: A Framework for Flexible Human-Machine Collaborative Systems," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 1392–1399, 2014

[15] M. Rudorfer, J. Guhl, P. Hoffmann, and J. Krüger "Holo Pick'n'Place," *Proc. Int. Conf. Emerging Technologies and Factory Automation* (*ETFA*), pp. 1219–1222, 2018.

[16] Google, ARCore, https://developers.google.com/ar/reference/.

[17] J.J. Craig, *Introduction to Robotics: Mechanics and Control*, Pearson Prentice Hall, 2005.

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

[19] S.G. Hart, "Nasa-Task Load Index (NASA-TLX): 20 Years Later," *Proc. Human Factors and Ergonomics Society Annual Meeting*, pp. 904–908, 2006.

[20] J.R. Lewis, "Psychometric Evaluation of the PSSUQ using Data from Five Years of Usability Studies," *Int. J. Human-Computer Interaction*, vol. 14, pp. 463–488, 2002.