Projection-based Augmented Reality Interface for Robot Grasping Tasks

L.L. Gong
Mechanical Engineering Department
National University of Singapore
9 Engineering Drive 1
Singapore 117576, Singapore
gongll@u.nus.edu

S.K. Ong
Mechanical Engineering Department
National University of Singapore
9 Engineering Drive 1
Singapore 117576, Singapore
mpeongsk@nus.edu.sg

A.Y.C. Nee
Mechanical Engineering Department
National University of Singapore
9 Engineering Drive 1
Singapore 117576 Singapore
mpeneeyc@nus.edu.sg

ABSTRACT
This paper presents an augmented reality (AR) interface for robot programming of pick-and-place tasks as well as assembly operations. The aim of the AR interface is to increase the intuitiveness and ease of robot programming. Marker tracking is used to automate sensor registration for easier workspace setup. Object recognition is employed to transform robot programming from an absolute coordinate system to an object-based system. This transformation provides flexibility to robot programming as the AR interface can be applied to workspaces that are configured differently to accomplish the same task. A spatially immersive display method is adopted to provide a projection-based direct overlay virtual workspace, so as to give users a better frame of reference in relation to the real workspace.

CCS Concepts
• **Human-centered computing→ Human computer interaction (HCI) →Interactive systems and tools**

Keywords
Robot programming; human-robot interaction; augmented reality.

1. INTRODUCTION
Industrial robots have been playing a crucial role in factory automation in recent decades. Conventional robot programming methods are critical for robotic manipulators to act with precision, speed and consistency. As such, robot programming requires the same degree of specificity and rigor to ensure the required level of performance. Most industrial robot programming applications use the teach pendant method of online programming. These applications mainly entail a robot repeating the same task, often seen in assembly line manufacturing. However, such applications have been decreasing and frequent reprogramming is the new trend, such as in much cited cases of robotic welding.

ICRCA 2019, July 26–28, 2019, Guangzhou, China

ACM ISBN 978-1-4503-7183-4/19/07…$15.00
https://doi.org/10.1145/3351180.3351204

Augmented reality (AR) is defined as a real-time integration of the real-world environment with computer generated virtual elements. The ability of AR to provide a composite view of virtual elements and the real world environment provides a user with a holistic approach to process information from Computer-Aided-Design (CAD) models and virtual simulations. By taking advantage of this feature, AR interfaces for robotics programming are used to solve the problems of conventional programming, such as intuitiveness, efficiency and safety. Much effort has been made with regards to robot programming using AR and human-robot interactions for applications such like task planning and task visualization.

In this research, a projection-based AR interface supported by a robot simulation and control software module has been implemented as an interactive robot workspace for robotic gripper task definition. The main contribution of this approach is to make use of the advantages of AR to present guidance information during the task definition process. The AR interface will guide the users through the full task definition process, simplifying the process into several steps with accompanying instructions. A laboratory-based prototype system for pick-and-place task definition for simple assembly tasks has been achieved.

The rest of the paper is organized as follows. Section 2 presents the related studies on the simplification of robot programming process. Section 3 gives an overview of the proposed AR-based robot programming system. Section 4 presents the evaluation of the system as well as user study. Section 5 gives the conclusions and future work.

2. RELATED WORK
There are many works that have contributed to the simplification of robot programming, making the process simpler and more intuitive. When considering this issue, it is important to consider the human-robot interface as it is the major determining factor of the user experience. Stadler et al. conducted a study to determine the effectiveness of AR and its impact on programming workload and overall task productivity [13]. The study showed that the effectiveness of the interface is strongly related to the target user demographic, i.e., expert users would require a minimalistic interface to increase productivity while novice users would prefer to have more information. Taking this into consideration, the AR interface to be developed in this research will take a middle ground approach by only displaying vital information and minimal prompts, such as task definition progress, object information and robot status.

In order to increase the ease and accuracy of task definition, AR has been used to provide instantaneous feedback to the users. Using laser projection onto the workspace, users can view their inputs and correct them concurrently [9]. When considering a display method to adopt, spatially immersive displays have been noted to be much more beneficial over the conventional head mounted displays in industrial applications. Spatially immersive displays adopt the use of video projection, which is able to provide a larger field of view to the users, allowing for multiple users for collaborative work, as well as adapt to applications where users can interact with the environment through virtual tools [11].

In order to reduce the complexity of the robot programming task, system features, such as environment and object recognition, have been extensively researched and used to curtail the need for detailed models and measurements during robotic task definition [10][8][3][6]. Through object recognition, the programming process for robotic grasping and manipulation is streamlined and becomes more adaptable as tasks can be defined based on objects as compared to an absolute coordinate system. The same robot program or task can be applied to workspaces configured differently as long as the objects can be detected. Although most applications of this methodology revolve around autonomous robots, the adoption of this level of autonomy can be effective in reducing programming workload in industrial applications.

A similar study on human-robot interaction in an industrial application uses a marker cube as an interaction device to allow a user to plan robot trajectories by specifying waypoints and via-points while the cube is tracked using a stereo camera [4][5]. However, such methods pose a few shortcomings, such as the need for calibration of the marker, robot-arm and camera location, in order to obtain accurate results. Additionally, due to the size of the cube-shaped marker as an interaction device, the precision of the user's input is reduced, and this is undesirable in the context of pick-and-place or assembly operations. Other innovative input methods include the use of spoken dialogue, gesture-tracking and active gaze tracking [1, 2, 9, 12]. Spoken dialogue as an input method was briefly considered in this research as it could provide significant convenience to the users, allowing multitasking or teleoperation. However, upon further consideration of the industrial environment, loud noise and the lack of precision in verbal commands will present more of a hindrance than a convenience. Moreover, these methods are tailored for single applications and are not versatile sufficiently by themselves for a holistic approach towards an interface.

3. SYSTEM OVERVIEW

3.1 System Architecture

The setup of the system is shown in Figure 1. It includes an ABB robot, Microsoft Kinect camera, OptiTrack system and a laser projector. Microsoft Kinect is used for its built-in depth sensor to obtain the depth map of the environment for object tracking, and the RGB camera of the Kinect is used for fiducial marker tracking. For the user interaction method, an infrared motion tracking system, OptiTrack, is adopted for its high accuracy that meets the system requirements. A miniature laser projector, Microvision SHOWWX, is used to render the projection-based AR display.

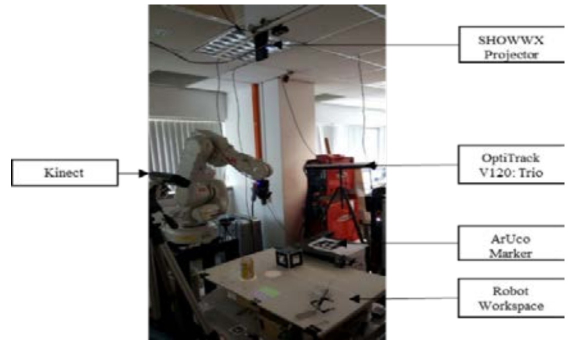

Figure 1. Workspace Setup.

In this prototype system, the features presented in the earlier section have been implemented as shown in the system architecture in Figure 2. This system is supported by three main software pillars that run together as a full system. Firstly, the main software used for communication with the robotic gripper is the Robot Operating System (ROS). Secondly, the system leverages on the object recognition capabilities of Object Recognition Kitchen (ORK), which is a plugin package built based on ROS. The last pillar of the system is the AR display and object data management algorithms, AR Stream and Object Manager respectively. AR Stream is developed as the AR display control and interface logic of the system while Object Manager is used to manipulate the outputs of ROS and ORK, so as to integrate the two to be used for task definition.

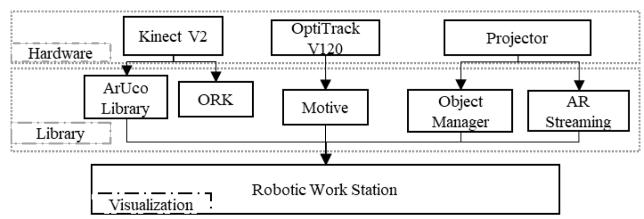

Figure 2. System Architecture.

3.2 System Implementation

ROS (Robot Operating System) is an open-source framework for writing robot software and is widely implemented in the field of research and on a variety of robot systems [7]. ROS provides an interface for inter-communication with the robot. ROS operates based on nodes and a publisher/subscriber mechanism which allows the integration of a variety of sensors and communication using a standard message format. As such, ROS serves as the main platform in which the different components of the system are integrated into. RVIZ (ROS Visualization) is one of the key tools that provides a general purpose three-dimensional visualization of sensor data and robot pose that the system uses for calibration of the sensors during development. RQT is applied as a Qt-framework for developing the AR interface that is projected onto the workspace on ROS. With RQT, the plugins of the system can be seamlessly integrated into RVIZ as one interface and controlled from RVIZ. Figure 3 presents the RVIZ interface with calibrated point cloud and camera image.

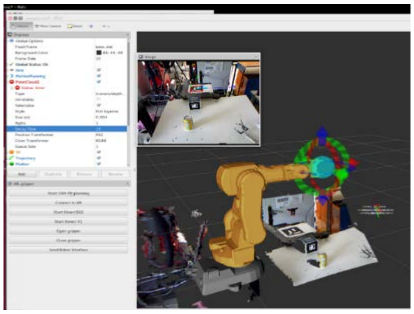

Figure 3.RVIZ interface with calibrated point cloud and camera image.

The AR Streaming provides the interface logic that guides the task definition process. Figure shows the projection-based AR display with AR elements. Figure 4 demonstrates the projection-based AR display with AR elements.

The program renders the display to be projected onto the workspace. Informative text prompts will be displayed showing instructions or error messages that will guide the user through task definition. Besides, during the place location selection phase, the shadow of the selected object will be projected constantly at the pointer location so as to allow the user to view the end location of the object as well as the two-dimensional space taken up by the object. Furthermore, interactive buttons are displayed on the interface and the users can use them to select their desired configuration

Figure 4. Projection-based AR display showing AR elements.

3.2.1 Task Planning

Upon completion of the task definition process, task planning can be performed. The coordinate location of the selected object will be set as the first way-point from its current position. The place location will be determined from the X and Y coordinates of the pointer when the place location is selected. As for the Z coordinate, the system assumes a flat plane throughout the entire workspace unless an object is recognized and detected. As such, the Z coordinate of the selected object will be stored and reused in defining the height of the place location. This method is more reliable and reduces the possibility of dropping or crushing an object should the place height be defined incorrectly by the user. If the option of placing an object on another object is selected, the place location height is calculated by adjusting the increase in height due to the height of the object to be placed on.

Intermediate points are added above the pick-and-place locations so as to avoid collision with other objects during these pick-and-place operations. Using six waypoints, a task is planned using the ROS Cartesian path planning library. The completed trajectories to be executed are passed onto the robotic gripper controller.

3.2.2 Assembly Planning

Task definition for each assembly step is handled in the same manner as a basic pick-and-place task using the interface. Once all tasks have been defined, the programmed tasks are released to the robot controller in the correct sequence. Similarly, path planning, pick and place locations are calculated separately for each assembly step. In every step, each object is defined to be placed on top of each other when in their original position before any execution. Hence, there is a need to account for the changes in object locations after the execution of the previous step. The relative position between the object to be placed on and the pointer during selection is stored for task planning. After each successful assembly step execution, the location and pose of the objects within the workspace are updated. Using the updated data ensures that the correct object location data is used for the planning of subsequent steps. The sequence definition steps for an assembly task can be summarized in Table 1.

Table 1. Steps of the assembly task definition.

Step 1	Define the location of the box B to be the place location for the can C2;
Step 2	Define the location of the can C2 to be the place location for the can C1;
Step 3	Execution: pick up the can C2 and place it onto the box B;
Step 4	Execution: pick up the can C1 and place it onto the top of the can C2;

The second assembly task is defined for the drink can C1 and place location on top of the other drink can C2. Although the current defined place location is at C2, only the relative position between the object and the pointer is recorded. Figure 5 depicts the planning and execution procedure for each planning step.

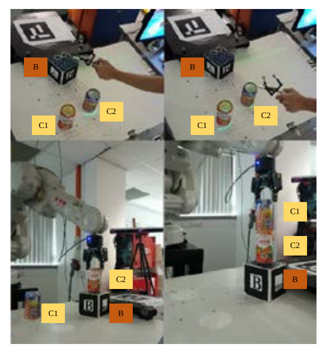

Figure 5. Assembly task definition sequence.

4. System Evaluation and User Study

After implementation and testing, the interface was assessed based on the requirements mentioned in the earlier sections, namely accuracy and precision. A user study was carried out to determine the overall effectiveness of the AR interface with regards to task definition in a pick-and-place context. Sources of error, their impacts and methods of correction will also be discussed.

4.1 System Evaluation

4.1.1 Objection Detection

Despite the cost effective performance of the Kinect sensor, there are some shortcomings that have to be addressed. Firstly, the Kinect has a minimum range of 0.5m whereby objects within this distance cannot be detected. This limitation together with the sensor's field of view, restrict the sensor's position relative to the workspace. As the sensor has to be sufficiently far from the workspace, the quality of its depth map will therefore be lower. Multiple trials of object detection were conducted and the results obtained from detection were compared to the object's actual location. From the trial result analysis, the detection error observed was identified to be a system error due to its low standard deviation (3.07 mm in X-axis and 3.54 mm in Y-axis) and was accounted for and corrected within the system. One source of this error would be the internal calibration of the Kinect sensor where its RGB camera location and depth camera location are not accurately calibrated with one another. As the translation between both sensors is not accounted for during the object coordinate transformation process, error will arise within the detected object location.

4.1.2 AR Projection

Accuracy in the projection display is critical as the users would rely on it greatly in task definition. Object shadows displayed have to be projected in the correct locations as the users will use these elements as a form of reference for their input to the real workspace. In the prototype setup, the laser projector was positioned directly above the robot workspace. This display has to be programmed using pixel count and a correlation had to be found between pixels and robot coordinates in order to render the virtual elements in the correct location.

Firstly, a pixel grid of 20px × 20px was projected over the entire workspace and an anchor location was defined. If the projector is out of place, aligning it to the anchor will reproduce the same configuration as the height of the projector has been fixed. Next, OptiTrack markers were placed at regular intervals along the grid lines. Motive was then used to record the location of each marker in the robot coordinates, hence forming a new grid expressed in the robot frame. With the pixel coordinate and real robot coordinate known for each specific grid point, this forms the basis of conversion between the real coordinates and the pixel location within the display. A simple method of interpolation was used to determine the coordinate to pixel conversion. However, this results in a severe error between the projected elements and the real location of the object due to the distortion of the projected image. Ideally, a one-to-one mapping for each pixel would give the highest accuracy; however, this is impractical to implement. Hence, more data samples at shorter intervals were taken to form smaller sample grid squares to increase the accuracy of the interpolation. Each projected element was then interpolated with different values based on its current location within the sampled grid defined.

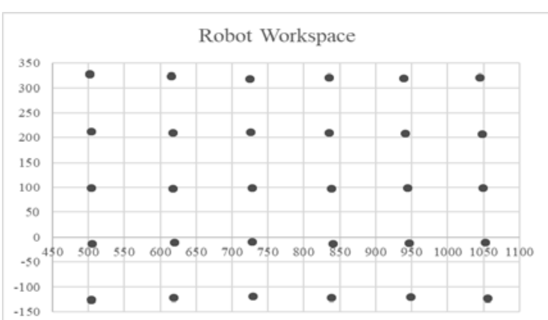

Figure 6. AR map shown in robot coordinates (mm).

From the AR map shown in robot coordinates in Figure 6, distortion effects in the projection can be observed from the sampled points. Although the error presented is relatively minor, it is one main source of error that remains unresolved and would increase in significance for a larger workspace. Another main source of error that remains unresolved within the display arises from the infrared tracking mentioned earlier. OptiTrack was used in the calibration of the interface; its inherent errors that could not be eliminated were brought over to the display interface.

Pick-and-place trials were conducted to measure the error between the projected location and the actual location of the object placed. From the trial results, the average error in both the X-axis and Y-axis are relatively low with low standard deviations as well (4.133 mm in X-axis and 6.36 mm in Y-axis). This degree of error is acceptable for pick-and-place operations and the given projection setup.

4.2 User Study

The user study was conducted with a group of ten subjects, each with various levels of skills and knowledge in robot programming. Each subject was given a demonstration and was tasked to plan and execute a simple pick-and-place operation that involves placing a canned beverage within a square of 8cm × 8cm. Next, they were given a questionnaire to evaluate their opinions of the system. The questionnaire results illustrated in Table 4 have shown that the majority of the subjects felt that the interface was fairly intuitive as the process was simple and easy to understand. Most of the subjects actively used the object shadow as a key reference to plan the place location of the can; some subjects however decided to directly place the pointer's tip on the center of the square as they felt that it gave them a better gauge. This was because the object shadow did not portray the center point of the can and was larger than the can's actual dimensions. Some subjects were seen to correct their inputs several times till they were contented with the results, hence showing the advantages of the immediate feedback and review that the AR display provides. Overall, the user study revealed positive feedback with regards to the AR display and the overall interface.

Table 2. Total score of questionnaire.

Question	Ratings (Total 100)
Ease of usage	84
Helpfulness of the AR display	82
Accuracy of the system	91
Speed	90
Usefulness of projected instructions	76

Additionally, to evaluate the effectiveness of the AR interface, each subject was timed in their attempt to complete the given task and the timings taken to complete the same task are recorded and analyzed. A programmer familiar with task definition using the teach pendant was also timed for completing the same task and this timing was used as a baseline for comparison. From the recorded timing results, all subjects took a shorter amount of time to complete their task (33.8s in average) as compared to the programmer using the teach pendant (72s). Some subjects have been noted to take a relatively longer duration than others as they spent a significant amount of time correcting their inputs to achieve results to their satisfaction.

5. CONCLUSION

In this project, a system design for a projection-based AR interface for robotic gripper applications has been discussed and a prototype system has been implemented to test and evaluate the effectiveness and benefits of the AR interface. A projection-based AR display is adopted to increase the intuitiveness of task definition as well as to provide the users with immediate feedback regarding user input. Object recognition and tracking were used to reduce the need for accurate user input as well as increase the versatility of task definition. Leveraging on the data obtained from object recognition, algorithms are used to determine user input and streamline the task definition process by automating certain aspects, such as calculation for collision and object place height. Methods, such as fiducial marker tracking, have been utilized to automate the calibration and tracking of sensors.

Further improvements can be made to develop the prototype system into a commercial setup that can be readily incorporated in robotic workshops. To better solve the pertinent issue of display calibration, distortion correction techniques can be applied to compensate for the distortion error encountered. Functionality in assembly task definition can be further developed in order to log and store longer and more complex tasks.

6. ACKNOWLEDGEMENT

The authors acknowledge contributions from Mr Koh Dong Koon. This research is supported by the Singapore A*STAR Agency for Science, Technology and Research Public Sector Research Funding Programme, Project No. 1521200081.

7. REFERENCES

[1] Atienza, R. and Zelinsky, A. 2002. Active Gaze Tracking for Human-Robot Interaction. In Proceedings. Fourth IEEE International Conference on Multimodal Interfaces. (2002), 261-266.

[2] Bassily, D., Georgoulas, C., Güttler, J., Linner, T., Bock, T. and München, T. U. 2014. Intuitive and adaptive robotic arm manipulation using the leap motion controller. ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of. (2014), 78–84. DOI:https://doi.org/10.1021/acs.energyfuels.7b01991.

[3] Ciocarlie, M., Hsiao, K., Jones, E. G., Chitta, S., Rusu, R. B. and Șucan, I.A. 2014. Towards Reliable Grasping and Manipulation in Household Environments. Experimental Robotics: The 12th International Symposium on Experimental Robotics. O. Khatib et al., eds. Springer Berlin Heidelberg. 241–252.

[4] Fang, H. C., Ong, S. K. and Nee, A. Y. C. 2012. Interactive robot trajectory planning and simulation using augmented reality. Robotics and Computer-Integrated Manufacturing. 28, 2 (2012), 227–237. DOI:https://doi.org/10.1016/j.rcim.2011.09.003.

[5] Fang, H. C., Ong, S. K. and Nee, A. Y. C. 2013. Orientation planning of robot end-effector using augmented reality. International Journal of Advanced Manufacturing Technology. 67, 9–12 (2013), 2033–2049. DOI:https://doi.org/10.1007/s00170-012-4629-7.

[6] Hinterstoisser, S., Lepetit, V. and Ilic, S. 2012. Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. Computer Vision–ECCV (2012), 593–596. DOI:https://doi.org/10.1007/978-3-642-37331-2_42.

[7] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A. Y. 2009. ROS: an open-source Robot Operating System. In ICRA workshop on open source software. 3 (2009), 5-10. DOI:https://doi.org/10.1109/IECON.2015.7392843.

[8] Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M. and Beetz, M. 2008. Towards 3D Point cloud based object maps for household environments. Robotics and Autonomous Systems. 56, 11 (2008), 927–941. DOI:https://doi.org/10.1016/j.robot.2008.08.005.

[9] Sato, S. and Sakane, S. 2017. An Interactive Hand Pointer that Projects a Mark in the Real Work Space. IEEJ Transactions on Electronics, Information and Systems. 121, 9 (2017), 1464–1470. DOI:https://doi.org/10.1541/ieejeiss1987.121.9_1464.

[10] Sbnchez, A. J. and Martinez, J. M. 2000. Robot-arm pick and place behavior programming system using visual perception. In Proceedings 15th International Conference on Pattern Recognition. 4 (2000), 507–510. DOI:https://doi.org/10.1109/icpr.2000.902968.

[11] Sherman, W. R. and Craig, A. B. 2007. The Future of Virtual Reality. Understanding Virtual Reality. January 1996 (2007), 437–457. DOI:https://doi.org/10.1016/b978-155860353-0/50010-0.

[12] Spiliotopoulos, D., Androutsopoulos, I. and Spyropoulos, C. D. 2001. Human-robot interaction based on spoken natural language dialogue. Proceedings of the European Workshop on Service and Humanoid Robots. August 2016 (2001), 25–27.

[13] Stadler, S., Kain, K., Giuliani, M., Mirnig, N., Stollnberger, G. and Tscheligi, M. 2016. Augmented reality for industrial robot programmers: Workload analysis for task-based, augmented reality-supported robot control. 25th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2016. (2016), 179–184. DOI:https://doi.org/10.1109/ROMAN.2016.7745108.