End-User Framework for Robot Control

Kaustubh Rajpathak
University of Texas at Arlington
Arlington, Texas, USA
kaustubh.rajpathak@mavs.uta.edu

Krishna Chaitanya Kodur
University of Texas at Arlington
Arlington, Texas, USA

Maria Kyrarini
University of Texas at Arlington
Arlington, Texas, USA

Fillia Makedon
University of Texas at Arlington
Arlington, Texas, USA

ABSTRACT

This paper showcases a developed end-user framework for a human-robot collaborative system for common tasks, such as pick and place. The system is designed for semi-automated pick and place tasks as well as manual operation making it flexible for multiple use-case scenarios. The goal of the system is to make the robot multi-functional, easy to use with a graphical user interface and to perform common tasks with the help of a human teammate. Integration with object recognition neural network (YoloV3) and an RGB-Depth camera helps automate pick and place tasks with a wide variety of objects.

CCS CONCEPTS

• **Human-centered computing** → **Graphical user interfaces**; **User interface programming**; • **Computer systems organization** → Real-time system architecture.

ACM Reference Format:
Kaustubh Rajpathak, Krishna Chaitanya Kodur, Maria Kyrarini, and Fillia Makedon. 2021. End-User Framework for Robot Control. In *The 14th PErvasive Technologies Related to Assistive Environments Conference (PETRA 2021), June 29-July 2, 2021, Corfu, Greece.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3453892.3464894

1 INTRODUCTION

Robot automation is quickly becoming the norm in many industries, such as manufacturing [1, 2], logistics[3], healthcare [4], or even personal use robots [5]. There is a growing demand for robots and humanoids that stem from both intrigue and functional use of the machines. Along with the excitement in the adoption of these technologies, there is also a growing fear among labor unions across the globe fearing the loss of their jobs [6]. The proposed framework in this paper not only helps existing workers adapt to new and better work environments but also helps the companies with a smooth transition to automation. As we have shifted to automated systems, there always has been a fear of robots malfunctioning [1]. Having a human supervise the actions of the robot is often the chosen direction to ensure the operational security of the robots.

PETRA 2021, June 29-July 2, 2021, Corfu, Greece

ACM ISBN 978-1-4503-8792-7/21/06.
https://doi.org/10.1145/3453892.3464894

Currently available Graphical User Interfaces (GUIs) for the most common industrial robots require the end-user to have robotic expertise [7]. Although existing software enables the end-user to program the robot in limitless ways, there often is a steep learning curve associated with using it and is prone to human error [8]. This makes the software complicated for novice users to use, discouraging widespread adoption of robotic systems. Thus there is a need for a GUI which makes it is easy to use for operating robotic systems in common world scenarios.

In this paper, a framework is proposed that enables novice end-users to control the robot in an easy-to-use approach through a GUI. Adding useful automation systems, such as object detection and position estimation of objects, enable the user to efficiently perform common manipulation tasks. A demo video is provided to demonstrate the working principle of the proposed framework.

The paper is organized as follows: In section 2, we describe the various components used and their operations in brief. In section 3, the operational working of the system is explained by demonstrating a simple cup stacking task. We conclude the work and provide future directions in section 4.

2 SYSTEM ARCHITECTURE

The goal of the proposed system is to enable the end-user to easily move a robotic arm in 3-Dimensional space, and pick and place target objects by manually choosing the orientation of the end-effector. The user is also able to choose an object from different objects in the scene and to place the object in a predefined location or manually move the robot to whichever location inside the robot's operating range. To implement such a system, an RGB-Depth camera (Intel Realsense D435i) is used, which not only is an input to the neural network but also useful in finding the real-world coordinate of the target object. The purpose of the object detection neural network is to identify different objects in the 2D image sent from the camera and surround them with a bounding box to identify their pixel position in the image frame. Coordinate transformation is performed using ROS Transform. To control the Franka Emika Panda robotic arm, we utilize a popular robot motion planning interface MoveIt [9] which works with a wide range of robot controllers. All these components run on Robot Operating System (ROS) [10] that enables the communication between these modules. The inter-relations between these modules are shown in Figure 1.

Object Detection (OD): This module consists of a neural network and the subscribers for the ROS transform (TF) and bounding boxes. The neural network is the ROS version of the YOLOv3 OD

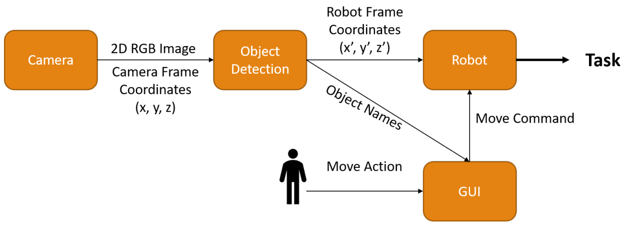

Figure 1: System Overview

system [11]. YOLO publishes the bounding box information which contains the name of the object and its pixel bounds. The OD module calculates the centroid of the object using the pixel bounds, and with the help of pyrealsense2 SDK, the module converts a pixel position in the image to a real-world coordinate in the camera's coordinate frame. Using the ROS TF, the object position is then computed with respect to the robot coordinate frame.

GUI: Figure 2 showcases the proposed GUI to perform the aforementioned tasks. The directional buttons (I) enable the operator to move the robot in fixed steps in that direction. The step size can be selected from the radio buttons (J) with slow taking small steps. The drop-down menu (A) allows the user to select objects that can be loaded by the load objects button (F). The pick object button (G) will move the end-effector near the object depending on the selected orientation (B, C). If the default or rotate gripper is selected, the end-effector will move to the top and front if horizontal is selected. The user can then position the gripper in any way using (I), close the gripper (E, (D) to open the gripper) and either move the arm manually to the place position or press the place button (H) to automatically go to a pre-defined place position. If the object is not identified by YOLO, the user can use the robot in a complete manual mode (teleoperation) by controlling the robot to move to the object using (I) and placing it the same way. The GUI is built using the PyQt5 library.

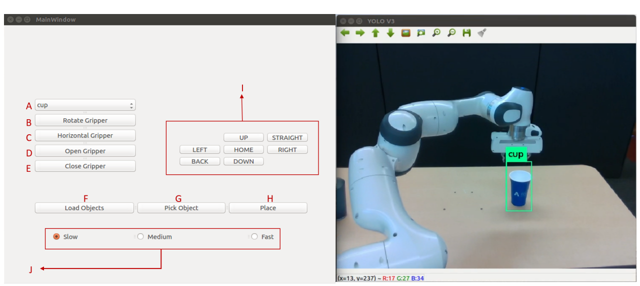

Figure 2: Developed GUI on the left and YOLO output on the right

3 DEMO SCENARIO

In the demo video, there are 2 cups placed on a table and the task is to stack them. To perform this task, the user selects the item cup from the drop-down box and presses the pick item button. This moves the end-effector to the top of the cup. The user then manually adjusts the pickup pose by manually positioning the gripper, closes the gripper, and places the object at a pre-programmed position. After the cup is placed, the user brings back the panda arm to its neutral pose (home). Since the object cup is already selected from the previous selection, the user does not need to select the cup again as the position of the second cup has automatically been updated to the same label. The user presses the pick object button and repeats the earlier procedure making sure that the cup is picked up from a similar pose as before to perfectly stack them when placed.

4 CONCLUSION

In conclusion, the system developed is easy to use and can be used in several pick and place scenarios. The object detection system is capable of detecting a large variety of everyday objects or can be trained with specific objects if needed. The selection of objects is well implemented in the GUI and operational efficiency is demonstrated in scenes with various objects. Position estimation of the camera module was tested on objects with various textures and shapes.

In the future, we aim on adding more functionalities to the system by updating the place position manually, differentiating between similarly identified objects and enabling robot learning by demonstration [12]. We also plan on evaluating the framework with human subjects.

REFERENCES

[1] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," Robotics and Autonomous Systems, vol. 94, pp. 43–52, Aug. 2017.
[2] M. Kyrarini, A. Leu, D. Ristić-Durrant, A. Gräser, A. Jackowski, M. Gebhard, J. Nelles, C. Bröhl, C. Brandl, A. Mertens, C. M. Schlick, "Human-Robot Synergy for cooperative robots, " Facta Universitatis, series: Autonomic Control and Robotics, vol. 15, no. 3, pp. 187-204, 2016.
[3] I. Sokolov, A. Misharin, V. Kupriyanovsky, O. Pokusaev, and Y. Kupriyanovsky, "Robots, autonomous robotic systems, artificial intelligence and the transformation of the market of transport and logistics services in the digitalization of the economy," International Journal of Open Information Technologies, 6(4), pp.92-108, 2018.
[4] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H.R. Nambiappan, K.K. Chaitanya, A.R. Babu, J. Mathew and F. Makedon, "A Survey of Robots in Healthcare," Technologies, vol. 9, no. 1, p.8, 2021.
[5] C. Breazeal, "Social robots: from research to commercialization," In Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction (pp. 1-1), 2017.
[6] T. Coupe, "Automation, job characteristics and job insecurity," International Journal of Manpower, 2019.
[7] T. B. Ionescu, "Leveraging Graphical User Interface Automation for Generic Robot Programming," Robotics, vol. 10, no. 1, 2021.
[8] G. Ajaykumar and C.-M. Huang, "User Needs and Design Opportunities in End-User Robot Programming," in Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, New York, NY, USA, 2020, pp. 93–95.
[9] S. Chitta, "MoveIt!: An Introduction," in Robot Operating System (ROS): The Complete Reference (Volume 1), A. Koubaa, Ed. Cham: Springer International Publishing, 2016, pp. 3–27.
[10] M. Quigley, K. Conley, B., Gerkey, J., Faust, T., Foote, J., Leibs, R., Wheeler, and A.Y. Ng, "ROS: an open-source Robot Operating System". In ICRA workshop on open source software, Vol. 3, No. 3.2, p. 5, 2009.
[11] J. Redmon, S. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016.
[12] M. Kyrarini, M.A. Haseeb, D. Ristić-Durrant, and A. Gräser, " Robot learning of industrial assembly task via human demonstrations," Autonomous Robots, vol. 43, no.1, pp.239-257, 2019.