5-1-2022

THE PROBLEM OF INEFFICIENT MACHINERY IN INDUSTRIES AFFECTS OVERALL PRODUCTION AND CAUSES SUBSTANTIAL ERRORS

Katia Lopez

THE PROBLEM OF INEFFICIENT MACHINERY IN INDUSTRIES AFFECTS OVERALL PRODUCTION AND CAUSES SUBSTANTIAL ERRORS

by

KATIA LOPEZ

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

HONORS BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2022

ACKNOWLEDGMENTS

Foremost, I would like to thank God for guiding me and for strengthening me to be able to complete this project with success.

I immensely thank my beloved family, especially my mom, who has supported me unconditionally. Thank you for your patience and understanding, and above all, for your love.

I would like to express my special gratitude and thanks to my professor, Dr. McMurrough, for his knowledge and guidance during the project. I offer my sincere appreciation for the learning opportunities provided by the project, which was funded by the Computer Science and Engineering Department.

Lastly, but not least, my completion of this project could not have been accomplished without the support of my team members.

May 3, 2022

ABSTRACT

THE PROBLEM OF INEFFICIENT MACHINERY IN INDUSTRIES AFFECTS OVERALL PRODUCTION AND CAUSES SUBSTANTIAL ERRORS

Katia Lopez, B.S Software Engineering

The University of Texas at Arlington, 2022

Faculty Mentor: Christopher McMurrough

The problem of inefficient machinery in industries affects overall production and can cause significant errors. The Human Assistance for Robot Arm (HAFRA) project consists of a Universal Robot, specifically the UR5, an Intel RealSense depth camera, and software programs. The robot receives a position from the computer vision program, which uses the camera to detect an AruCo tag, and then suctions the envelopes from a conveyor belt and places them into a bin. The project also includes human assistance, which allows remote picking by an operator. A human is able to view an image and use the mouse cursor to select a pixel coordinate. This coordinate is converted to a real-world position and sent to the robot so it can continue picking. Human assistance increased the robot's performance

and achieved a success rate of nearly 100%. Industries with product lines can use HAFRA to optimize their tasks with precision.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Background

Currently, robots and machines are being implemented into industries to perform regular tasks that humans do. Industrial robots can perform various tasks, including picking from a bin, picking from a conveyor belt, and organizing boxes on a pallet. Replacing a human worker with a robot would increase performance, however accomplishing a faultless robot that operates autonomously proves difficult. With current research and development, machines can reach a high performance but still experience mistakes. The purpose of this project is to implement human assistance for the UR5 robot, to eliminate the small percentage of mistakes. In case the robot reaches a scenario, it does not know the solution to, the robot will communicate with a human remotely. A human will be able to help the robot by providing it instructions on how to solve the problem it has encountered. Once implemented, the robot could perform with a success rate of 100%.

1.2 Related Work

There is peer-reviewed scholarship relevant to the research in universal robots and bin picking robots. There are many universities and companies who are working to develop fully functioning random bin-picking robots. Through all this research and development in computer vision and motion control, software has enabled robots to make basic distinctions between objects. However, robots still struggle with precision and accurately detecting shapes and objects. According to the article by Fujita and other authors, robot competitions have encouraged the development of bin-picking robot technologies [3]. The bin-picking systems in the competitions used a grasping design which used the combination of a suction and two-finger suction [3]. In the paper published by IEEE, a universal software framework with a focus on virtual bin picking was presented, which enabled the integration of various algorithms for recognition, motion planning, several types of robots, grippers, and vision systems [4]. In addition, the International Conference on Intelligent Robots and Systems held in 2010, approached the bin-picking issue by applying the latest state-of-the-art hardware components, incorporating an environment model, and allowing for the physical human-robot interaction during the entire process [2]. In another robotics conference held in Karlsruhe, Germany, an applicable solution was presented for the bin-picking problem which was based on a standard 3D-sensor and was able to handle arbitrary objects [1]. Bin picking is so complex, specialists have attempted to apply deep learning techniques, but the results have been disappointing, at least for industrial use [5]. As a result, more research is being done in computer vision and robotics projects.

1.3 System Overview

The project has three major components, the UR5 robot, the computer vision program which also implements human assistance, and the movement program that is responsible for moving the robot to specified positions. The UR5 robot is shown below in Figure 1.1. The computer vision program uses the RealSense camera to detect an AruCo tag and calculate the center pixel coordinate. The vision program coverts the pixel coordinate to a real-world coordinate, based on the robot’s perspective, and sends the position to the movement program. The movement program moves the robot to perform its task, which is to pick up envelopes off a conveyor belt and place them into a bin. The robot picks up the envelopes by using a suction gripper which is connected to a vacuum. Human assistance is implemented inside the vision program so that if the robot detects something unique or a problem it does not have a solution to, it will communicate with a human. For example, if there is not an AruCo tag on the envelope or the tag is torn, it will notify the human and wait for instructions. The human will then be able to access the robot remotely, will assist the robot, and ultimately fix the problem.

Figure 1.1: The UR5 Robot

CHAPTER 2

METHODOLOGY

2.1 Solution

The overall project was to develop an envelope picking application using the UR5 robot. For the purposes of the Honors project, a component of the project was providing a software solution for remote picking by a human operator. The purpose of human assistance was to be able to oversee and fix any mistakes the robot made. This would increase overall productivity and ultimately reduce any errors or mistakes.

2.2 Method

The project used agile development, which is a software product development process. For the first semester of this project, the project plan had to be established, and the requirements specification and initial system design had to be completed. For the second semester, a detailed system design, a system test plan, the product implementation, and a final demonstration of the project had to be delivered. Acceptance testing also had to be performed towards the end of the project.

First, the team had to configure the computer in the lab, so it used Ubuntu 18.04 as its operating system. The router and the modem also had to be configured so the computer and the UR5 robot would be in the same network and be able to communicate with each other. In addition, the Robot Operating System (ROS), which is a set of software libraries and tools that are used to program robots, had to be installed. The version of ROS that was installed was Melodic which used the Python 2 programming language instead of Python 3. The computer was able to communicate with the robot by using ROS, calibration and driver commands, and a simple Python script that moved the robot to a specific position based on the six joint angles.

A Python program was then able to turn the vacuum on and off by using general-purpose input/output (GPIO) pin inputs. This enabled the robot to pick up the envelopes by using the suction gripper which was connected to the vacuum. Next, librealsense, an open-source SDK and software library that is used for vision programs, had to be installed. The Python version of it, called PyRealSense, was used to write the program to detect an ArUco code which would be on the envelopes.

Finally, it was necessary to implement human assistance into the computer vision program as a main portion of the Honors project component. If the robot encountered something unexpected and could not find a solution, it had to save an image of what it was viewing and send it to the user's computer. The user had to be able to use their mouse cursor on the image to select a coordinate where the robot should aim to pick the envelope. The user had to send this information to the robot and the robot would continue picking as normal.

CHAPTER 3

IMPLEMENTATION

3.1 ROS Visualization

For testing purposes, ROS Visualization (Rviz), a 3D visualization tool for ROS, was used. This tool was used to create virtual objects that would represent boundaries in which the robot would not be able to trespass. Four objects were created which were a ceiling, backwall, floor, and tool object.

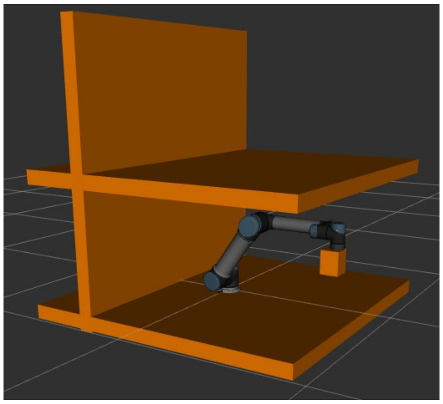

Figure 3.1: Rviz Environment

These objects were based on the location of the robot in the lab. The robot was placed on a table, so a floor object had to be created so the robot would evade hitting it. The table was also near a wall of the lab, so a backwall object had to be added for the same safety reason. A tool object was also created to represent the suction gripper so when the robot went down to pick up an envelope it would not go too far. Finally, a ceiling object was created to limit the possible paths of movement the robot was able to move to.

3.2 Functionality

The vision program detected a fiducial which was an ArUCo Tag on an envelope. It obtained the center pixel coordinate and converted it to a real-world (x,y) position. If it could not detect a tag, it would send an image to the human assistance program. If it did, it would send the coordinates to the movement program which moved the robot to pick up an envelope and drop it off at the bin.

3.3 Human Assistance

The human assistance program opened an image of what the robot was viewing if it could not find a tag. The user could use their mouse cursor on the image to view the x and y values of the pixel coordinate. For example, in Figure 3.2, the x value is 551 and the y value is 320. The user could select a pixel coordinate that was close to the center of the envelope and input these values into the command terminal that was used to run the programs. The program would then convert this pixel coordinate into a real-world coordinate from the robot's perspective.

At the beginning of the project, six joint angles were used to create a position for the robot to move to. Later, the robot could move to a position based on an x,y,z coordinate and orientation values for each component. This would facilitate sending a coordinate to the robot because only an x and y value would have to be sent instead of six joint values which were angles.

To do the computation of translating a pixel coordinate to a real-world coordinate, another program was used. This program compared the pixel distances of an image to centimeter distances. Based on this, an equation was developed to convert easily from a pixel value to a centimeter value. This coordinate was then sent to the movement program which would move the robot to the specified position that was sent from the user. The robot would then suction the envelope, place it inside the bin, and continue this cycle again.

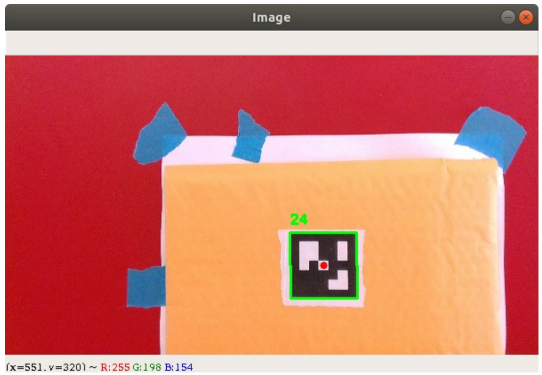

Figure 3.2: Human Assistance Window

CHAPTER 4

RESULTS

By adding human assistance to the project, the success rate of the robot increased to 100%. Various scenarios were checked to test the program. For example, an envelope was placed without an AruCo tag, and the robot took a picture and sent it to the human assistance portion correctly. Another envelope was placed that had a tag that was torn. Again, the robot asked for human assistance and was able to pick up the envelope. In case there was no envelope to pick up, the user was able to enter an exit code, which was the coordinate (0,0) to terminate both programs.

To summarize, if the robot, specifically the vision program, encountered something it was not familiar with, the robot paused and sent an image to the human assistance portion. A human was able to view the image and use their mouse cursor to select a pixel coordinate. The pixel coordinate was converted to a real-world coordinate and the position was sent to the robot to continue picking.

CHAPTER 5

CONCLUSION

A robot that can mimic and perform human tasks efficiently and correctly can be a great benefit to industries. The UR5 robot increased performance and achieved a success rate of 100% through human assistance. This project is a profitable idea by replacing the cost of paying a human worker constantly to only a one-time investment.

There were limitations for this project. To begin, it had to be completed in one semester, specifically in approximately four months. In addition, work on the project could only be done in person since the robot was located in the Engineering and Research Building, in lab 305. Another constraint was that the budget for the project was $800.

The project could later be implemented in industries with product lines. During the Covid-19 pandemic, we have learned the importance of hygiene and disinfecting all surfaces. An industrial robot, such as the UR5, could be used to handle mail such as envelopes, therefore mitigating the risk of spreading viruses.

In the future, artificial intelligence and machine learning could be used to improve the performance of the robot. The algorithm can be updated each time it encounters a situation it does not know the solution to, so that it saves the information and progressively needs less human assistance.

REFERENCES

[1] Dirk Buchholz, Marcus Futterlieb, Simon Winkelbach, and Friedrich M Wahl. Efficient bin-picking and grasp planning based on depth data. In 2013 IEEE International Conference on Robotics and Automation, pages 3245–3250. IEEE, 2013.

[2] Stefan Fuchs, Sami Haddadin, Maik Keller, Sven Parusel, Andreas Kolb, and Michael Suppa. Cooperative bin-picking with time-of-flight camera and impedance controlled dlr lightweight robot iii. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4862–4867. IEEE, 2010.

[3] Masahiro Fujita, Yukiyasu Domae, Akio Noda, GA Garcia Ricardez, Tatsuya Nagatani, Andy Zeng, Shuran Song, Alberto Rodriguez, A Causo, I-Ming Chen, et al. What are the important technologies for bin picking? technology analysis of robots in competitions based on a set of performance metrics. Advanced Robotics, 34(7-8):560–574, 2020.

[4] Adrian Schyja, Alfred Hypki, and Bernd Kuhlenkötter. A modular and extensible framework for real and virtual bin-picking environments. In 2012 IEEE International Conference on Robotics and Automation, pages 5246–5251. IEEE, 2012.

[5] E Truebenbach. Is fully automated bin picking finally here? 2019.

BIOGRAPHICAL INFORMATION

Katia Lopez started her bachelor's degree in computer engineering at the University of Houston in 2018. After her first year, she decided she wanted to major in software engineering. Since the University of Houston did not offer that major, she transferred to the University of Texas at Arlington, which had a successful Computer Science and Engineering department. She became a member of the Honors College and other student organizations. She hopes to become a successful software engineer and later continue her education in graduate school.