DEVELOPMENT OF AN UNMANNED GROUND VEHICLE PHOTOGRAMMETRY DEVICE

Muammar Ibrahim
Faculty of Mechanical Engineering
Universiti Teknologi Malaysia
(UTM) Johor Bahru, Malaysia.
Royyan@graduate.utm.my

Zair Asrar bin Ahmad
Applied mechanics and design
Department Faculty of Mechanical
Engineering Universiti Teknologi
Malaysia (UTM) Johor Bahru, Malaysia.
zair@utm.my

***Abstract*— Robotics has become very popular in this era, with high precision and complex functionality. One novel industrial application in robotics is its use to create 3D point cloud data on infrastructure to inspect any defects in need of maintenance, this is normally done with 3D LiDAR sensors. However, cameras are also capable of creating point cloud data with textures, using a method known as photogrammetry. Hence, the aim of this project is to develop a teleoperated (master-slave) mobile robot that is capable of taking image data and sending the information automatically to another compter. The image is then processed into point cloud and 3D mesh with texture. In this paper, we implemented ROS2 control framework and v4l2 capture interface on a differential drive robot and USB webcam. The mobile robot uses an existing chassis design on the market. The experimental results show that our system was able to perform teleoperation and photogrammetry of target objects in controlled lighting condition.**

Keywords*— *Photogrammetry; Mobile robots; Teleoperation

I INTRODUCTION

In recent years, mobile robots have become more accessible and are widely used to replicate human's task in industry; namely service, logistics, automated warehousing and infrastructure sectors. Moreover, the current trend which is Industry 4.0 involves use of software and robotics to replace humans in specific tasks. Unmanned ground vehicles are robotics systems which operate on land without human operators on-board, often controlled through teleoperation. It is usually used in small or unsafe environments where human supervision is important.

Photogrammetry is an established technique in the mapping industry, as it provides accurate relative dimension of the target being scanned. With the recent rise in online retail, there is demand in providing a 3D model of high ticket goods, this can be done if the manufacturer provides the CAD model of the item, photogrammetry provides a solution to bypass this, allowing retailers to 3D scan their goods and display it as a digital model. The method of photogrammetry to scan object is not the same as mapping, photos need to be taken around the target with optimal overlap of 5 degrees between each photo. Image resolution effects mesh polygonal density.

II PROJECT METHODOLOGY

The flow of this project system could be seen in Fig. 1 below. After preliminary experiments on photogrammetry and software for point cloud generation, software and hardware development is conducted, which is then integrated into a finished robot. Preliminary study was done using a turntable and fixed camera, this was done to evaluate the viability of consumer grade cameras in producing 3D models from images. Software development was done using ROS2 on Linux ubuntu 20.04, as Microsoft's WSL2 gave graphic driver incompatibilities when using ROS2 foxy.

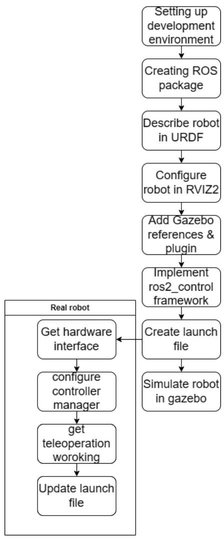

Fig. 1. Diagram of software workflow

Software development was done by creating a ROS2 package and first simulating the mobile robot in Gazebo and RVIZ2, the code is then modified to work with the hardware interface rather than Gazebo simulation. The main controller of the robot is a raspberry pi, which is running a headless setup of ubuntu 20.04 server to minimize processing draw from a GUI. This is programmed via SSH from the development machine, the main package can be pulled by the raspi from a github repository of the project, and modified using VS code remote extensions. ROS2 control package for differential drive robot was used as the main controller, tuned for the robot's dimensions.

Camera feedback was received from the raspi by running a ros video for linux node using the function;

ros2 run v4l2_camera v4l2_camera_node --ros-args -p image_size:="[1280,720]" -p camera_frame_id:=camera_optical_link

This broadcasts camera data to the network. Upon testing it can be concluded that the optimal camera resolution is 1280x720 compressed, as above this the framerate stutters, giving imprecise results. The camera feed is saved in the development machine by creating a ros image_view subscriber node;

ros2 run image_view image_saver image:=image_raw _save_all_image:=false _filename_format:=foo.jpg __name:=image_saver

This saves the images according to the frequency of ros2 run function, thus the frequency has to be reduced to artificially "time" when the photos are taken.

The main components for the robot hardware is the raspi as the microprocessor which provides target joint velocities to a connected arduino uno as a microcontroller which then provides target rotations to an L298N motor driver.

The ros_arduino_bridge sketch is flashed on the arduino, this provides a base controller for a differential drive robot that accepts ROS Twist messages and publishes odometry data back to the PC. The package needs to first be modified to work with the arduino pinouts used in the real robot.

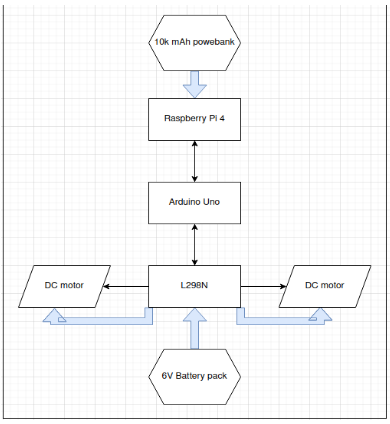

Fig. 2. Electronics layout

Two power sources were used for this robot, a 10,000 mAh powerbank powers the raspberry, arduino and camera. A secondary AA battery pack powers the motor driver and motors. Fig 2 shows the electronic layout of the robot, blue arrows represent power exchange while black arrow is communication between components.

A Mobile robot Design

After testing the operation of the electronic components by providing target PWM movement to motors from development machine, the robot is assembled. The electrical components are stack on top to minimize vibration damage.

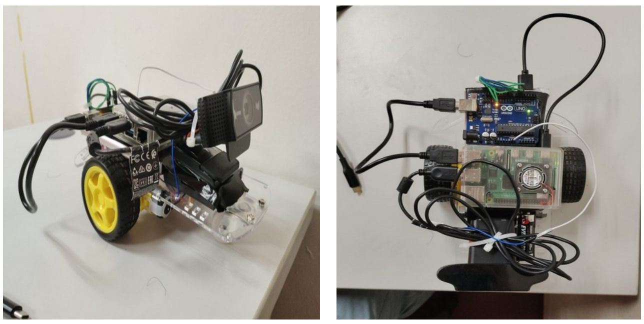

Fig. 3. Assembled robot

III RESULTS AND ANALYSIS

A. Gazebo & Odometry Results

Odometry provides estimated positional data of the robot, which provides a tool for analytics of its performance. Before implementing the code on the real robot, gazebo simulation was tested, controlled using a teleop twist keyboard nod on ROS2, this allowed control of the robot using a keyboard. Fig 4 shows gazebo simulation and Fig 5 shows its odometery on RVIZ2.

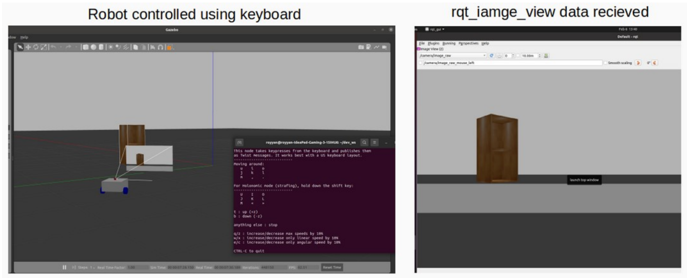

Fig. 4. Gazebo simulation with camera feed received

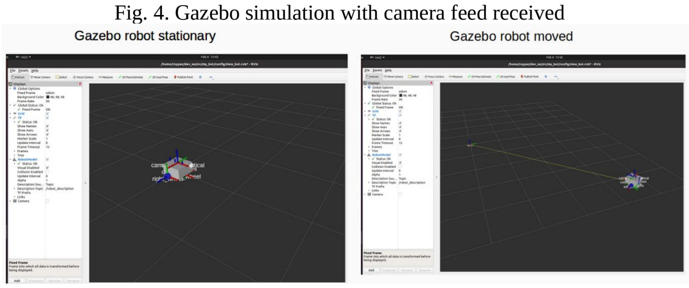

Fig. 5. RVIZ odometry as gazebo simulation is moved

B. Image & 3D model Results from robot

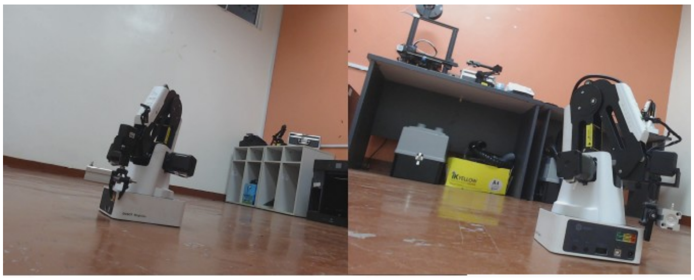

Fig. 6. photo taken by robot

From the simluation is was found that the robot turning radius can be controlled by adjusting its angluar velocity, the robot is then tested in Lab setting, to circle a dobot magician robotic arm and take pictures of it. A total of 45 images were taken at compressed 1280x720 resolution. The images are stored on the computer and then run through meshroom, and opensource 3D reconstruction software to produce the result in Fig 7, it was converted to a .obj format and tested to work with unity engine.

Fig. 7. Resulting 3D model from photos taken by robot

IV Conclusion

In conclusion, the 3D model produced by the robot is satisfactory. Odometry failed to work on the real robot as the motors used did not have built in encoders. Image quality was severely limited by the processing power of the raspberry pi, which bottle necked the webcam quality to compressed 720p images. The viability of conducting photogrammetry with mobile robot has been explored.

V Acknowledgement

I would like to express my gratitude to my family, friends, and all the knowledge that I have received. Special thanks to my friend Wessam Hamid for the guidance with ROS, and my supervisor, Dr. Zair Asrar for the feedbacks, information, guidance through the whole progress of my project.

References

1 Chen, M.-C. and Huang, S.-H. (2003) ‘Credit scoring and rejected instances reassigning through evolutionary computation techniques’, Expert Systems with Applications, 24(4), pp. 433–441.

2 Chen, M.-C. and Huang, S.-H. (2003) ‘Credit scoring and rejected instances reassigning through evolutionary computation techniques’, Expert Systems with Applications, 24(4), pp. 433–441.

3 T.B. Asafa, T.M. Afonja, E.A. Olaniyan, H.O. Alade (2016) ‘Development of a vacuum cleaner robot’, Mechanical Engineering Department, Ladoke Akintola University of Technology

4 Olaf Hallan Graven (2018) ‘An Autonomous Indoor Exploration Robot Rover and 3D Modeling with Photogrammetry’, University College of Southeast Norway

5 M. Z. H. Noor, S. A. S. M. Zain, L. Mazalan (2013) ‘Design and Development of Remote-Operated MultiDirection Unmanned Ground Vehicle (UGV)’, Faculty of Electrical Engineering Universiti Teknologi Mara

6 Abhishek Roya , Mathew Mithra Noel (2014) ‘Design of a high-speed line following robot that smoothly follows tight curves’, Bosch Engineering and Business Solutions, Bangalore

7 Joshua J. Medina, James M. Maley, Siddharth Sannapareddy, Noah N. Medina, Cyril M. Gilman, John E. McCormack (2020) ‘A rapid and cost-effective pipeline for digitization of museum specimens with 3D photogrammetry’, Moore Laboratory of Zoology, Occidental College, Los Angeles, CA

8 Adrian Savari Thomas, Mohd Fahrul Hassan1, Al Emran Ismail1, Reazul Haq Abdul Haq, Ahmad Mubarak Tajul Arifin1, Md Fauzi Ahmad (2019) ‘Portable Mini Turntable for Close-Range Photogrammetry: A Preliminary Study’, Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia

9 Hardware Interface: https://github.com/joshnewans/diffdrive_arduino

10

11 Ros arduino bridge: https://github.com/hbrobotics/ros_arduino_bridge