Artificial Intelligence Autonomous Vehicle for the Blind

Promsurin Phutthammawong [1], Phurichaya Angbunthorn[2]
Phisan Kaewprapha[1], Dulyachot Cholaseuk[2]

[1]Department of Electrical Engineering and Computer Engineering, Faculty of Engineering,
Thammasat University, Thailand, promsurinm@hotmail.com,
[2]Department of Mechanical Engineering, Faculty of Engineering,
Thammasat University, Thailand

Abstract

Artificial Intelligence Autonomous Vehicle was aimed as a new mean of transportation for blind or visually impaired people, to help them become more independent in their daily life.

This vehicle was designed based on a simple electric scooter configuration that is easy to use. Artificial Intelligence was utilized for car recognition, traffic light recognition, and voice recognition. With autonomous features driven by Simultaneous Localization and Mapping technology (SLAM) and real-time anti-collision sensors along with a 3D depth camera, the scooter provides dependable self-driving vehicle to transport a blind person to a mapped location accurately and safely. The vehicle can be operated in both autonomous and manual modes. A prototype was developed and tested. The test result showed that the autonomous vehicle can be safely operated at 35% of the full speed.

Keywords: Autonomous, Artificial Intelligence Vehicle, Blind

1. General Information

Artificial Intelligence Autonomous Vehicle for the Blinds was developed for blind people to allow them to travel through non-Braille block pathways that are not specifically designed for the blinds. A blind person traveling alone with just a blind stick on an unfamiliar non-Braille block route could pose a risk from accidents. Therefore, blind people normally rely on assistances, such as a guide dog, a personal assistant, or guidance staff, which is, sometimes inconvenience. Our vehicle can help them become more independent.

This vehicle was designed based on a commodity scooter configuration available on market. Artificial Intelligence was utilized for car and traffic light recognition, and voice recognition. With autonomous features driven by SLAM and real-time anti-collision sensors with an additional 3D depth camera, the scooter becomes a reliable self-driving transportation to take a blind person to a mapped location accurately and safely. The vehicle can be operated in both autonomous and manual modes.

2. Theory & Related Literature

2.1 Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) is a technique for identification and localization the position where the robot is located. This technique is widely applied in many applications, such as military operation or industrial application. SLAM predicts the pose of a robot and can identify the direction, mapped into the environment by using odometry combined with other sensor components based on algorithms like Adaptive Monte Carlo Localization (AMCL). By linking with 2D Navigation stack and 3D mapping or alternative methods, one can make the SLAM produces positions accurately under environment limitation. SLAM also weighs to trust any devices at the programmer required and can provide feedback control. The above context will be integrated with visualization software for display and stack other technologies.

2.2 Adaptive Monte Carlo Localization

AMCL is a probabilistic localization system for a robot moving in 2 Dimensions. It implements an Adaptive Monte Carlo Localization algorithm for tracking the pose of a robot against a known map. AMCL takes a number of parameters. AMCL takes the laser-based map, laser scans, transform messages, and pose estimates. The standard AMCL algorithms are already available for two drive configurations, differential drive and omni-directional drive. In this work, one of the major contributions is that we have designed some compensates methods to let the tri-cycle wheel or three wheels scooter possible to drive along a planned path by 2D navigation stack [1].

2.3 Robot Operating System

The robot operating system is a flexible framework for writing robotic application. ROS consists of vast selections of tools and libraries. As a result, ROS was built from the ground up to encourage collaborative robotics software development. For example, one laboratory might have experts in mapping indoor environments and could contribute a world-class system for producing maps. Another group might have experts at using maps to navigate, and yet another group might have discovered a computer vision approach that works well for recognizing small objects in clutter. ROS was designed specifically for groups like these to collaborate and build upon each other's work [2]. Our vehicle uses ROS to implements navigation autonomous and try to cross-platform development with Artificial Intelligence.

2.4 Mechanical Design

Figure 1 shows a diagram with all modules connected to the scooter and show the prototype scooter.

CP-5

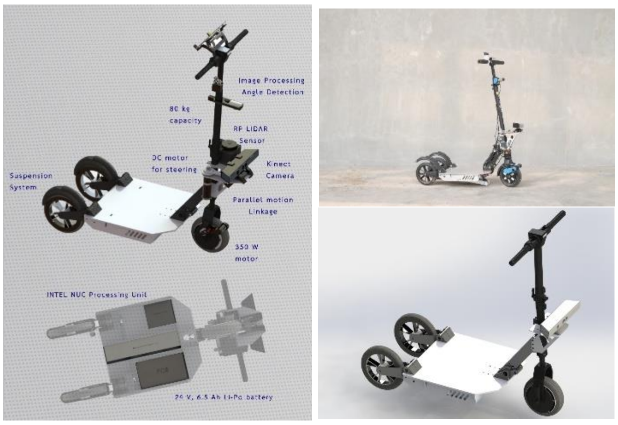

Figure 1. a diagram with all modules connect to the prototype scooter.

The vehicle was designed as a tricycle scooter with a 350W hub motor driving front wheel. The front steering column is actuated by a servo motor connected to the column by a parallel motion linkage mechanism. Image processing angle detection system generate steering angle feedback data for odometry. The data is processed in the robot operating system.

The design speed was specified based on walking speed walking of people around transportation. We find out that the most appropriate speed is should be under 1 meter per second [3]. We take into account the anatomy of human [4] resulting in the average of length between foot while standing is 271.78 millimeters and the longest length of feet in the united states of America is size 13 US or 297.89 millimeters. The information was used to design the platform. Aluminum alloy type 6061-T6 which has high strength-to-weight ratio is selected as the frame material. The prototype was designed, and the 3D CAD was generated for converting to URDF and implement into Robot operating system.

3. Implement and Development

Figure 2 shows how the mechanical system and the control system are integrated.

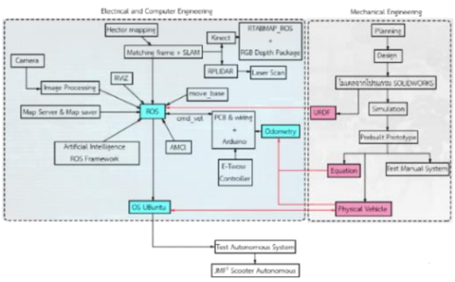

Figure 2. Integration between mechanical and electrical & computer engineering works.

The front wheel drive part was adopted from an existing commercial scooter (E-TWOW Scooter driving part) and integrated with an embedded system that utilized Arduino mega to control while communicating with Intel NUC to process data that retrieve from sensors such as Lidar, Kinect Depth Camera, Encoder.

Figure 3 shows the concept of autonomous operation. First, the pre-mapping process is conducted.

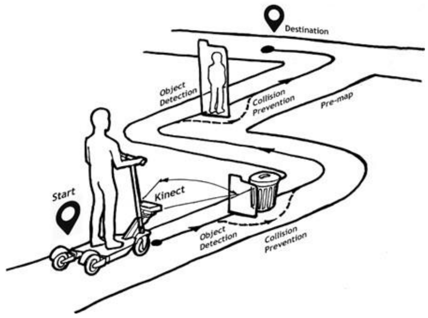

Figure 3. Artwork concept planning how this vehicle could working

We use a 3D Kinect camera and Lidar Sensor to start mapping by using Hector Slam mapping [5]. We also tried to generate map with RTABMAP (Real-time Appearance-based mapping) [6]. The result between 2 options of mapping software were shown in Figure 4.

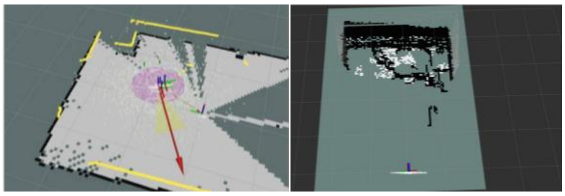

Figure 4. Using Hector Mapping with RP-Lidar and Kinect camera

If we consider the result, Kinect camera provides more data but less angle of environment, on the other hand, RP-Lidar provides a better environment degree but less accuracy. This is because Kinect uses Depth Camera which gives point cloud and converts to laser-scan data. We can fusion these 2 data sources but this method we still working on the development, so Kinect was primarily used in this current prototype. Figure 5 shows how Kinect Camera sees the environment by using a depth camera feature. Figure 6 shows Kinect camera that used on this vehicle.

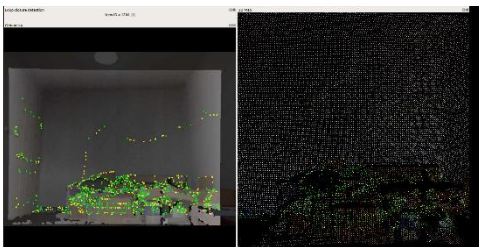

Figure 5. The Vision of Kinect Depth Camera sees the environment.

CP-5

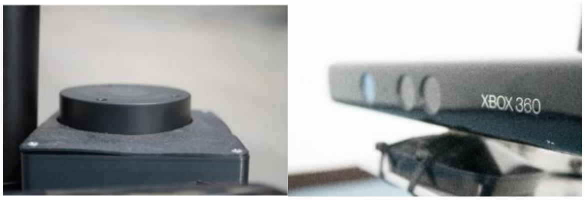

Figure 6. RP-Lidar sensors module and Kinect camera.

Figure 7. Intel NUC under vehicle for processing.

Next, the data from odometry sensor is required to obtain actual travelling path. The data consists of distance, speed, and direction. An encoder is installed to the front wheel to measure distance and speed, as shown in Figure 8. Image processing angle detection system is installed on the steering column to measure direction the data is then sent to ROS.

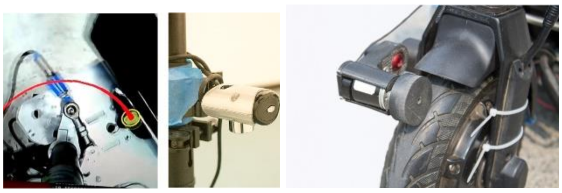

Figure 8. A camera installed on a steering column for image processing angle detection and Encoder device on the wheel to measure speed and distance.

All the data will get to process with Robot operating system, and we can render a map. We also can weight each sensor confident to use those data. After data has processed, we can launch a map node that contains all the pre-map data and continue to deal with the localization algorithm. Figure 9 will show some of the results that came from parameters configured in a safe area.

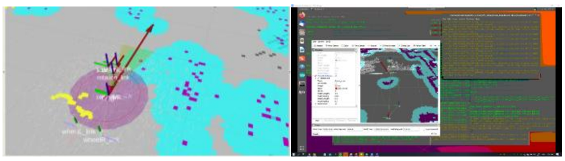

Figure 9. Map with Safe area configured with Global Planner and Local planner with ROS on Ubuntu.

The parameters such as Global Planner and Local Planner, Max velocity in axis must be set manually adjusted and these parameters affect another node such as control node linked to the Arduino board, Adaptive Monte Carlo Localization Algorithm, 2D Navigation Stack, etc. Localization is operated by AMCL and is linked with the Navigation node. Some of the navigation were already provide by the ROS. The artificial intelligence engine such as darknet deep learning algorithm performs cars detection as a safety measure.

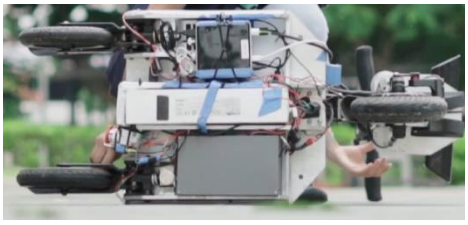

Figure 10. Devices under the vehicle.

Figure 10 depicts how components are placed under the vehicle which includes Intel NUC, microcontroller and the battery.

Darknet YOLO Algorithm [7] was implemented to a made decision based on 70% confident level to send signals for brake the vehicle. When the vehicle starts to break, it will provide digital signal into the controller and let the vehicle to stop running by communicating on a relay switch under the vehicle. Figure 11 shows how artificial intelligence predict from scooter vision.

Figure 11. Deep Learning A.I. with YOLO model predict car detection.

4. Experimental and Setup

We perform an experiment observing how well the vehicle operates both manual and automatic mode. On the autonomous mode maximum current using for a drive is 10.59 Ah and the maximum distance to travel in a laboratory test is 2.75 km when operating continuous run at 3.6 Km/h. In practice, the load will not take the full load so the distance can go farther away than the testing system in the laboratory. From the test result, we can load a maximum of 80 kilograms. In a manual mode system, the scooter can drive on a 7-degree horizontal slope without flip. The maximum vertical degree angle can do by run-upward is 16.7 degree and the maximum angle without flip run-downward is 22.2 degree. We have results that we adapt our new methods to the Adaptive Monte Carlo Localization and navigation move base. By our experimental setup, we have set by putting the chair as an obstacle that needs to avoid and let the scooter start position away from obstacle estimate to 1.5 meters. We let the autonomous navigation system to reach destination point which far from obstacle approximately 3.5 meters away at different speeds. The selected speed is 35%, 75%, and 100% of minimum acceleration and operates without a human on board to test which speed is most stable and accurate from current power processing. The Intel NUC have Core i5 processors and ram at least 4GB.

5. Result and Discussion

After a test to avoid obstacle with move navigation stack an autonomous, result at 35% speed of minimum acceleration as Figure 12 and the result at 70% speed of minimum acceleration. and the result at a 100% speed of minimum acceleration as Figure 13. we can conclude that the result at 35% acceleration is the best and most stable operated. We can also make a conclusion that the processing power is required more than the current computing node or we can run more just a single computer for computation.

Figure 12. Testing of the prototype scooter at 35% and 70% of minimum speed.

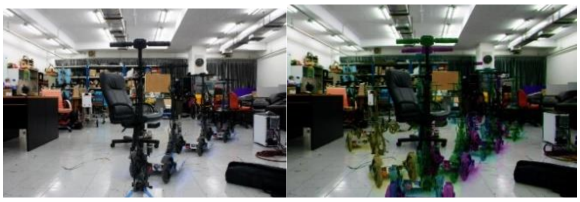

Figure 13. Testing of the prototype scooter at 100% of minimum speed and comparison all of three testing result.

6. Conclusion

Artificial Intelligence Autonomous Vehicle for the Blinds has been developed. The vehicle took the form of a three-wheeled scooter with front wheel hub motor drive. Autonomous system is based on a pre-mapping information. The pre-mapping is done with Kinect camera. Intel NUC is used as the brain of the vehicle. ROS is utilized to process the sensor data that consists of distance speed and direction. It is then controlling the drive motor and front wheel steering servo to navigate the vehicle autonomously to a predetermined destination. Along the route, active local object detection is performed, and local route is adjusted to avoid collision.

Preliminary test result showed a promising potential. The vehicle was able to perform autonomous operation at. 30% of minimum speed movement and Figure 13. shows comparison each speed percent used (yellow & green is 35%, blue is 70% and purple is 100% of minimum speed). Future improvement includes raspberry pi devices for improve processing part of sensors, voice recognition for command a scooter, fine-tuning parameters to make more stable and accurate while navigation and planning, increase to full speed at minimum speed movement.

Acknowledgement

I would like to extend my sincere thanks to Asst. Prof. Dr. Bunyong Rungroungdouyboon for suggestion, valuable advice and support this research, and I also wish to thank Asst. Prof. Dr. Pradya Prempaneerach and Asst. Prof. Dr. Photchara Ratsamee, Cybermedia Center, Osaka University for helpful advice.

This research was supported by Electrical and Computer Engineering, and Mechanical Engineering, Faculty of Engineering, Thammasat University

References

[1] Ewing Kang (2019) Adaptive Monte Carlo Localization, Available at: http://wiki.ros.org/amcl (Accessed: 6th July 2020).
[2] ROS Developer team (2020) About ROS, Available at: https://www.ros.org/about-ros/ (Accessed: 7 July 2020).
[3] M. F. M. Ali, M. S. Abustan, S. H. A. Talib and I. Abustan, A Case Study on the Walking Speed of Pedestrian at the Bus Terminal Area, 2018.
[4] Henry Dreyfuss (2001) Anthropometric Data, Available at: http://www.arch.mcgill.ca/prof/castro/arch304/winter2001/dander3/frame/Homepage8.htm (Accessed: 18 July 2020).
[5] Stefan Kohlbrecher (2012) Hector-mapping, Available at: http://wiki.ros.org/hector_mapping (Accessed: 18th July 2020).
[6] Mathieu Labbé (2019) RTAB-Map Real-Time Appearance-Based mapping, Available at: http://introlab.github.io/rtabmap/ (Accessed: 18th July 2020).
[7] Qi-Chao Mao, Hong-Mei Sun, Yan-Bo Liu, Rui-Sheng Jia (2019) 'Mini-YOLOv3', Real-Time Object Detector for Embedded Applications, IEEE Access, PP (99):1-1(10.1109/ACCESS.2019.2941547), pp. 2.

About the authors:
Promsurin Phutthammawong
Bachelor's Degree of Electrical and Computer Engineering, Faculty of Engineering Thammasat University Year 2020, Interesting topic: Autonomous, IoT and Artificial Intelligence

CP-5