Perception and decision making for intelligent robots

Oscar Hjertqvist, Martin Lamm, Hasan Sharineh, Yaochen Song, Joel Trollheden

***Abstract*—For Autonomous Robots to make informed decisions, perform actions autonomously, and maneuver in complex and dynamic environments, they require accurate perception systems to perceive their surroundings. This article investigate how the different sensors on a TIAGo 101 robot can be used for autonomous navigation, and describes the implementation of a navigation, perception and decision making system on the physical robot. A map was created of a lab and corridor environment which the TIAGo robot successfully could use to localize itself. Furthermore the object detection could detect humans and obtain the distance between the human and the TIAGo robots camera. The obtained distance was used to take decisions based on a state machine, where the robot slowed down the closer it got to a human and completely stopped once it was within 1m. The estimated distance to a human was also tested to be within an accuracy of 0.03m and a standard deviation of 0.004 m.**

I. Introduction

Autonomous Robots are already an essential part in many personal assistance and manufacturing applications, and the usage of Autonomous Robots in social indoor environments such as offices or homes are expected to increase in the coming years [1]. In order to enable Autonomous Robots to make informed decisions, perform actions autonomously, and maneuver in complex and dynamic environments, the robots require rigorous perception systems to perceive its surroundings. Some examples of commonly used sensors for autonomous navigation are: Absolute positioning sensors (GPS/GNSS), visual sensors (Cameras), range sensors (Radar/Lidar/Ultrasonic sensors), and relative positioning sensors (IMU) [2].

This article describes the steps taken to investigate how the different sensors on a TIAGo 101 robot can be used for autonomous navigation in an indoor environment, as well as implementing a navigation, perception and decision making system on a physical robot. The research was limited to three areas of investigation:

- The implementation of a off-the-shelf localization and mapping software provided by PAL-Robotics that allows the TIAGo robot to map its surroundings and to identify its location in a map.
- The implementation of an off-the-shelf perception system that allows for the TIAGo robot to identify objects ahead of it. Examples of identifiable objects are persons, tables and chairs.
- Implementation of a basic navigation and decision making algorithm that can use the information from the sensors, the map and its current location to take decisions such as stop, drive and slow down.

The testbed used in this study was a TIAGo 101 robot from PAL-robotics, visulized in figure 1. It is a multi-functional mobile research robot with several different sensors, and actuators [3]. The TIAGo robot utilizes ROS for managing its sensors, actuators and the communication between its different parts. The available sensors on the TIAGo robot were the following [3]:

- A laser range-finder in the front of the base.
- Sonars in the back of the base.
- IMU in the base.
- Motors current feedback in arm/wheels
- Stereo microphones in the torso.
- Force/torque sensors in the wrist.
- RGB-D camera in the head.

Fig. 1. Overview of TIAGo robot, Source: [3]

However, as this article is focused on object detection, mapping and localization as well as decision-making, only the laser range-finder, odometry data from the wheels and RGB-D camera were used and considered in this study. Other limitations included that the testing only was performed in a controlled indoor lab/corridor environment which meant that other more dynamic and challenging environments such as offices or factories were not considered. Due to time restraints, the neural network used for the object detection was not fine tuned for our environment. The decision making system was built on a simple state machine where only the closest detected human were considered when making decisions. All other objects were ignored. The laser range finder mounted at the base of the robot was used to localize and plan the robot's navigation path by detecting walls and objects. However obstacles which primarily are above the floor, like tables, could therefore not be detected and put into the map. Because of this the navigation goal was carefully chosen to avoid such situations with risk of collisions, i.e. before letting the robot move to the navigation goal any object that was not detectable for the laser range finder was removed.

II. Related work

The creation of the map, the localization and autonomous navigation system proposed in this paper was largely built from work made by PAL-Robotics [4], [5]. Using the TIAGo robot's laser range finder sensor at the base of the robot, together with the odometry from the motor feedback sensors, a map of the area could be created utilizing a Gmapping ROS Package. Gmapping is a highly efficient SLAM (Simultaneous Localization and Mapping) algorithm that utilizes a Rao-Blackwellized particle filter to learn grid maps from laser range data [6]. Gmapping has good capacity for robustness, which enables it to generate accurate maps in various environments[7].

The localization and pose estimation was done using the AMCL ROS package [5]. AMCL (Adaptive Monte Carlo Localization) is a probabilistic localization algorithm for a vehicle moving in a two-dimensional occupation grid map [8]. It uses a particle filter to estimate the location and pose of the robot with the range-finder and odometry data. Initially the robot is uncertain of its location with particles, containing the robot's state, all over the map. When the robot moves the particles are updated based on the sensors data which eventually leads to the particles converging to the most likely position of the robot.

The autonomous navigation is done by placing a `tiago_2dnav` in rviz [5], which the robot tries to create a path to with the help of two costmaps, a global and local costmap. The global costmap is based on the created map and it is used to avoid static objects which were found during the mapping process, such as walls, lockers or other easily seen objects. The local costmap uses the TIAGos laser range finder in real time to detect and avoid new obstacles. The path planning algorithm will try to create a path to a specific position and also minimize the "cost" of going there by not colliding with objects on the map or identified later using the laser [5].

The object detection was based on a ROS implementation of Ultralytics YOLOv8 [9]. YOLOv8 (You Only Look Once) is a pre-trained computer vision model on the famous data set COCO [10]. The model is able to recognize objects and then apply image segmentation and/or tracking among other functions [11].

III. Method

The following section describes the method, including the simulation environment and the tools used in the study. The section also provides a detailed description of how the object detection, decision making, and localization was implemented on the robot.

A. Simulation environment

During the project Gazebo [12] was used in parallel to the physical robot to test the different software packages. In Gazebo there are multiple different scenarios or virtual worlds that can be utilized for testing different things, such as an office environment or a space with many different rooms which would be suitable for testing. The two scenarios mainly used in this project were the `simple_office_with_people` and `small_office`. In figure 2 the former of the two can be viewed.

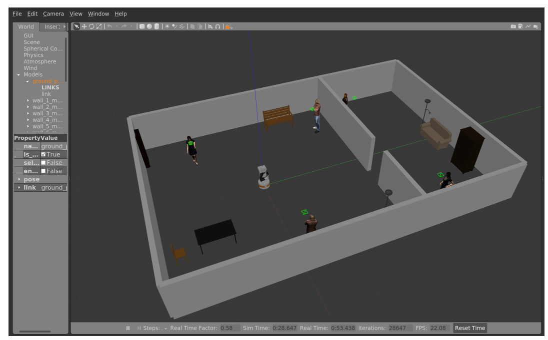

Fig. 2. The environment simple_office_with_people in Gazebo

The reason for using simulation instead of only implementing it on the real robot is due to safety concerns, both for the humans around the robot as well as the robot itself. Furthermore testing is easier and more efficient in simulation when compared to the real robot as the environment is completely known, can easily be reset and manipulated.

B. Object detection, segmentation & depth estimation

As already mentioned, the objects were detected and tracked by utilizing YOLOv8 in the form of a ready-to-use ROS package [9]. The given solution was adapted to not only do object detection, but also instance segmentation and depth estimation of the objects. The object classes, the bounding boxes, and the object masks were estimated from the RGB-D image data using the YOLOv8n-seg model [9]. An example of a detected object and mask is visualized in figure 3. The distance to the different objects were then derived by using the depth image 4 and calculating the average depth of all the valid registered pixels in the corresponding object mask. The detected object classes and 3d positions (bounding boxes and depths) were then published as an array in a custom ROS message to be used in the decision-making.

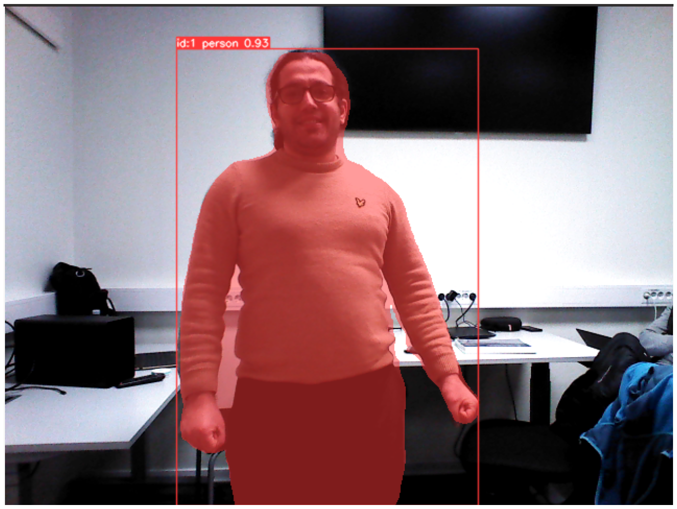

Fig. 3. Object detection and segmentation of a person from Yolov8 model.

Fig. 4. Example of a depth image, given by the RGBD-camera. Pixels with unregistered depth are blacked out in the image.

C. Mapping & Localization

Along with the position of the objects, the state of the robot is also necessary to make informed decisions [13]. A mapping and localization method was therefore implemented for the TIAGo robot. Using the TIAGo robot's laser range finder sensor at the base of the robot, together with the odometry from the motor feedback sensors, a map of the area was created utilizing the Gmapping ROS package [4].

By manually navigating a room with a joystick controller, a map of the room could be created utilizing a SLAM algorithm simultaneously as it kept track of the position of the robot. For this project, a map of a lab and corridor environment was created and used in the pose estimation, path planning and the autonomous navigation of the robot.

D. Navigation & Decision Making

The autonomous navigation was done by utilizing the package `tiago_2dnav` to execute navigation to a specific location, along with two custom ROS nodes for the object detection and decision making In figure 5 the entire ROS architecture that was created for the project can be viewed. With all the nodes, input and output.

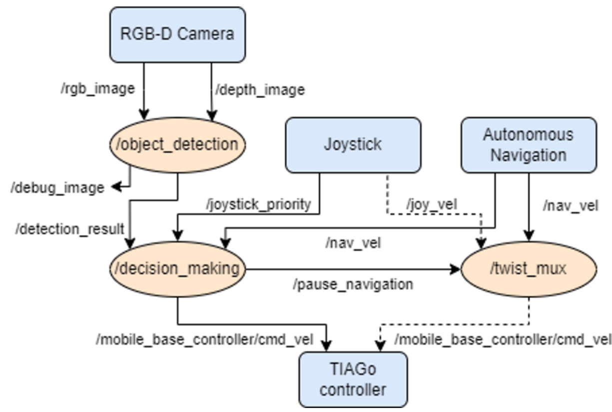

Fig. 5. The ROS architecture of the project.

The nodes, `object_detection_node` and `decision_making_node`, allow for a human to be detected, along with the distance to the human from the camera. The decision making node is based on a simple state machine that takes decision based on the distance to the closest person. The different decisions is visualized by a decision tree in figure 6 that describes what the TIAGo robot should do depending on if there is a human within 2 m, within 1 m or not at all. Also if the joystick controller is on then it should not make any decisions.

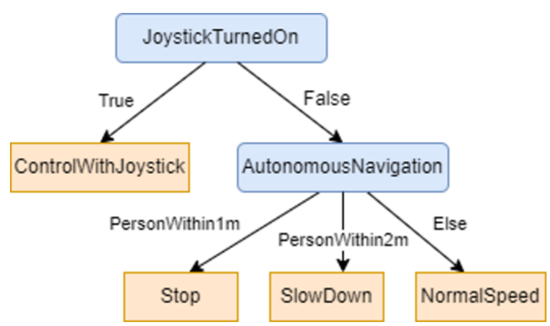

Fig. 6. A decision tree that describes each of the actions that the TIAGo robot takes depending on various factors.

More specifically the `decision_making_node` only modifies the message sent by the autonomous navigation if the identified person is closer than 2 meters. If a human is closer than 2 meters it starts to linearly decrease the velocity until it reaches a distance of 1 meters where the robot stops completely. Equation (1) describes it further.

$$v_{\text{dec}} = \begin{cases} v_{\text{nav}} & \text{if } d_{\text{pers}} > 2 \\ (d_{\text{pers}} - 1) \cdot v_{\text{nav}} & \text{if } 2 > d_{\text{pers}} > 1 \\ 0 & \text{if } d_{\text{pers}} < 1 \end{cases} \tag{1}$$

IV. RESULTS

In figure 7 the created map of the lab room as well as the surrounding corridor can be observed. Note the rather convoluted lab room with several objects of some kind in blue or pink can be seen. These are objects that have been moved since the creation of the map chairs, tables or people that are picked up by the laser range finder.

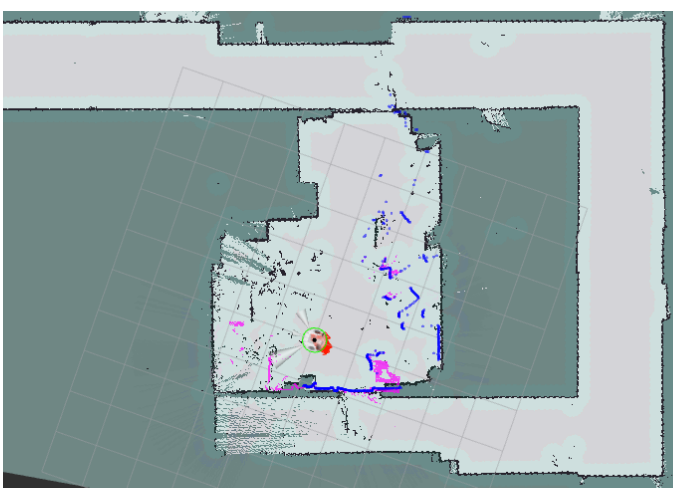

Fig. 7. The map that was created around the lab room and the surrounding corridors, along with the current location of the TIAGo robot.

Figure 8 shows the robot navigating in a corridor with a door opened in front of it. The purple lines represent what the laser range finder is seeing at the moment while the black lines is the saved map. The short red lines on the TIAGo robot are the particle filter from the AMCL algorithm that is used to localize the robot on the map.

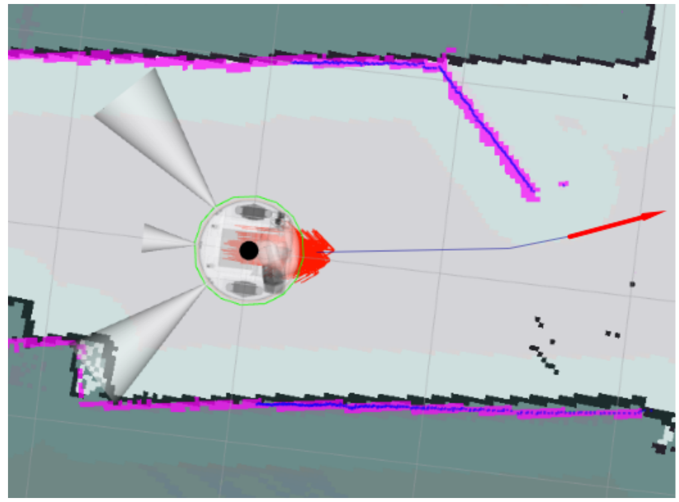

Fig. 8. The TIAGo robot navigating in a corridor with an open door in front of it.

To test the accuracy of the depth estimation, one of the conducted tests was with the TIAGo robot standing still with a human about 1 m in front of it for about 1 min. Creating a histogram of the distance perceived from the robot's Camera to the human was done, which can be seen in figure 9. As the figure shows, the histogram is close to a bell curve and the span is about 0.970 m to 0.995 m. Furthermore the mean and standard deviation was calculated for the test giving $\mu = 0.979$m and $\sigma = 0.004$ m.

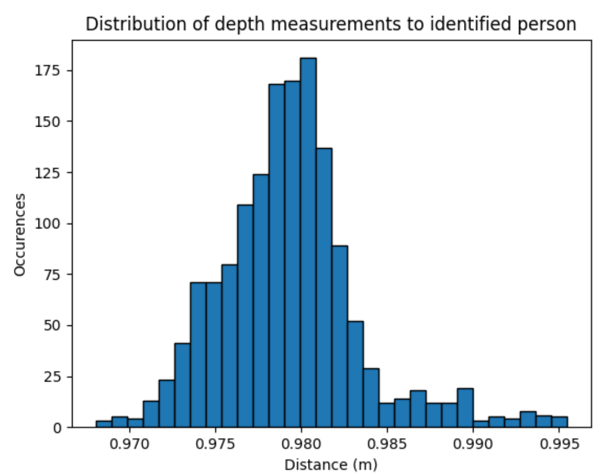

Fig. 9. A histogram of the distances from the TIAGo head to the human of the test when a human stood still in front of the TIAGo for about 1 min with a distance around 1 m.

Another test was conducted where the TIAGo robot was placed in front of a flat wall for around 10 seconds. The purpose of this test was to check the accuracy of the depth sensor and see if the depth perception still shows the same distance from the camera at the center of its view as on the edges of its view. An image of the result of this test can be viewed in figure 10. All in all the noise of the depth image had a standard deviation of $\sigma = 0.00026$ m, when the robot was placed 0.65 m from the wall.

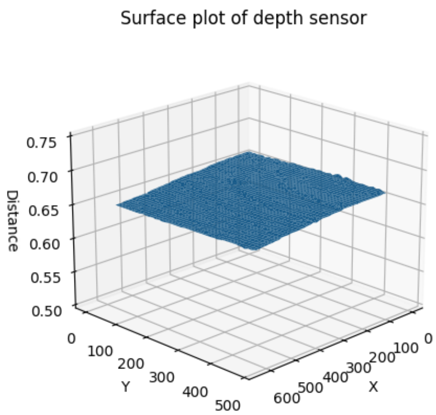

Fig. 10. The test result in the form of a surface plot when the TIAGo robot stood still in front of a wall to investigate its depth perception in the center and the edges of its camera view.

To test the autonomous navigation and decision making, the TIAGo was given a goal on the map where a human stood in its path. The closer the TIAGo got to the the human the slower it went until it was within 1 m and it completely stopped. Figure 11 shows a photo from this test. Note that the lockers to the right are not close enough to affect the movement of the TIAGo.

Fig. 11. The navigation test where it has stopped as a human is observed within 1 m.

Furthermore from the same test, data about the linear velocity was also taken. This can be seen in figure 12 and 13. Figure 12 shows in blue the new velocity and the orange is the reference velocity that the TIAGo navigation wants to follow originally. Figure 13 shows the distance to the person from the camera on the TIAGo. The figures makes it evident that as the TIAGo gets closer it slows down until it completely stops when it is 1 m to the target and then when the human move out the cameras field of view, the TIAGo continues to follow the reference velocity.

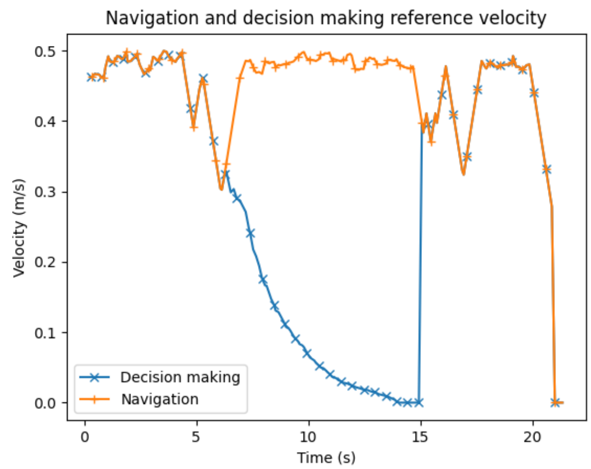

Fig. 12. The reference x velocity in orange and the new reduced velocity in blue from the navigation test as it sees a human. When within 1 m it stops and then continues after human has moved out of view.

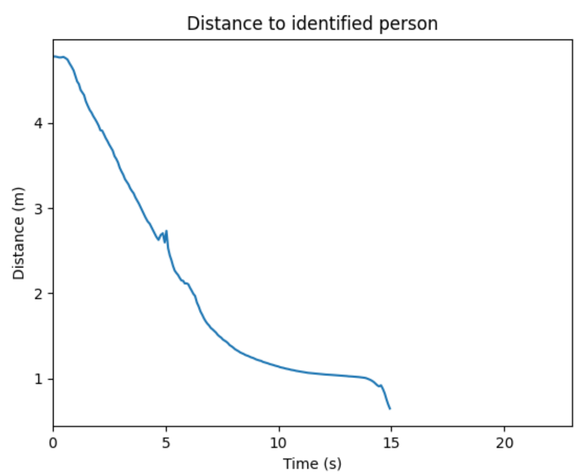

Fig. 13. The distance observed by the depth camera on the TIAGo to a human. Note that after 15 seconds the human moves out of view.

A demo video of the navigation test can be accessed by the following link: https://www.youtube.com/watch?v=hLsdNSf9o_8 [14].

V. Discussion

As could be seen in figure 7, tables, chairs and human legs were not that obvious to discern in the map. This could pose an issue as the TIAGo robots laser range finder is close to the ground and it may not see certain objects well. Therefore the navigation was implemented in the corridor instead of inside of the lab. This could potetially also be resolved by also using the RGBD camera to augment the map with all the objects that the laser range finder is not able to see quite as well. Though this was not deemed necessary as the project goal did not require perfect navigation in a cluttered lab room.

Furthermore the localization as seen in figure 8 operates well as it can see new changes in the world such as the open door. The localization still has the same concern regarding table legs and chairs but in the corridor the localization performs as expected and required.

The current decision making system is rather simple as stopping if the TIAGo robot sees a human within a certain distance of the camera was considered the most appropriate decision to implement. This could be further developed by increasing either the complexity or the number of decisions possible depending on more factors. For example allowing for the robot to judge if it is able to drive past a person or if it has to wait and involving more objects than just humans.

Currently, the distance to an object is being calculated as the mean pixel depth for every pixel in the detected object. This might cause problems if an object has an uneven shape, i.e if a person put out its arm, as it would lead to a faulty estimation of the object distance. A simple improvement could be to instead take decisions based on the closest distance to the objects, but in that case potential outliers would have to be filtered out to avoid taking decision overly reactive based on noise.

The fact that the histogram in figure 9 almost follows a bell curve probably means that the human stood rather still and that the depth perception is acceptable. The span is from about 0.970 m to 0.995 m. Furthermore the mean and the very small standard deviation of 0.004 m along with the span of about 2.5 cm means that tells us that the distance measurement is quite precise.

The surface plot 10 is almost flat, which indicates that the distance to objects on the edges of the field of view of the camera, is the same as if the object was in the center. The horizontal field of view of the depth camera is $\theta = 58°$ [15], which means that in reality the distance ratio to objects on the edge of the field of view would be $1/cos(\theta/2) = 1.14$. This means that the real distance to objects on the edge of the robots vision would be at most 14% further away than the identified distance used in the decision making.

When the TIAGo navigation test was conducted it could be observed that the velocity was reducing the closer it got to the human as expected. When it got within 1 m it stopped completely. Furthermore when the human moved out of view it then immediately went back to the reference velocity. This behavior was exactly what was intended and this part could be seen as a success.

VI. Conclusion

Creating a map of the environment around the lab room and corridor as well as localizing the TIAGo robot inside of this map was successful. Furthermore the object detection could also detect a human and obtain the distance between the human and the camera of the TIAGo robot. The decision making worked as intended, where the robot slowed down when it detected a person within 2m and stopped completely when a person was within 1m. Although care should be taken when a human is standing at the very edge of the field of view since there could be an error of up to 14 %.
Future research building on this project could focus on further developing the decision making and navigation system to be able to handle more complex scenarios. This could be finding a new trajectory to the navigation goal instead of only slowing down or stopping the robot if it sees a human. Another could be to further develop the object detection to detect more objects and train the network for our specific environment.

Overall the project has seen moderate success though with room for future work to improvements.

References

[1] M. Misaros, O.-P. Stan, I.-C. Donca, and L.-C. Miclea, “Autonomous robots for services—state of the art, challenges, and research areas,” *Sensors*, vol. 23, no. 23, p. 4962, 2023. [Online]. Available: https://doi.org/10.3390/s23104962.

[2] S. Nahavandi, R. Alizadehsani, D. Nahavandi, *et al.*, “A comprehensive review on autonomous navigation,” *arXiv preprint arXiv:2212.12808*, vol. 1, no. 1, p. 48, 2022.

[3] “Tiago robot.” Last accessed on 2023-12-07, Pal Robotics. (), [Online]. Available: https://pal-robotics.com/robots/tiago/.

[4] J. Pages. “Create a map with gmapping.” Last accessed on 2023-12-07. (), [Online]. Available: http://wiki.ros.org/Robots/TIAGo/Tutorials/Navigation/Mapping.

[5] J. Pages. “Localization and path planning.” Last accessed on 2023-12-07. (), [Online]. Available: http://wiki.ros.org/Robots/TIAGo/Tutorials/Navigation/Localization.

[6] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with rao-blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, pp. 34–46, 2007.

[7] S. K. NT *et al.*, “Mobile robot terrain mapping for path planning using karto slam and gmapping technique,” in *2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT)*, IEEE, 2022.
[8] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte carlo localization: Efficient position estimation for mobile robots,” *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1999.
[9] Alpaca-zip. “Ultralytics_ros.” Last accessed on 2023-12-07. (), [Online]. Available: https://github.com/Alpaca-zip/ultralytics_ros.
[10] T. Lin, M. Maire, S. J. Belongie, *et al.*, “Microsoft COCO: common objects in context,” *CoRR*, vol. abs/1405.0312, 2014. arXiv: 1405.0312. [Online]. Available: http://arxiv.org/abs/1405.0312.
[11] “Yolov8 documentation.” Last accessed on 2023-12-07, Ultralytics. (), [Online]. Available: https://docs.ultralytics.com/.
[12] “Gazebo simulator.” Last accessed on 2023-12-11. (), [Online]. Available: https://gazebosim.org/home.
[13] I. Dianov, K. Ramirez-Amaro, and G. Cheng, “Generating compact models for traffic scenarios to estimate driver behavior using semantic reasoning,”
[14] O. Hjertqvist. “Ssy226 group 5 result.” Last accessed on 2024-01-11. (), [Online]. Available: https://www.youtube.com/watch?v=hLsdNSf9o_8.
[15] “Xtion camera specification.” Last accessed on 2024-01-11, ASUS. (), [Online]. Available: http://xtionprolive.com/asus-xtion-pro-live?fbclid=IwAR23P6NbmgZa4WSD0bJO833dWQfp71_up8v6Mr2W4-xhgvLbiFpLsKg02o0.