POLITECNICO DI TORINO

Master of Science Degree in Mechatronic Engineering

Indoor SLAM and Room Recognition with Deep Learning at the edge

Supervisor: prof. Marcello Chiaberge

Candidate: Andrea Eirale

Introduction

With the advancement of technology, a series of new possibilities have emerged in the field of service robotics. One of these concerns the development of SLAM (*Simultaneous Localization and Mapping*), which in the last years allowed integration with numerous hardware and software components, in order to accomplish the most varied tasks. This project, developed with PIC4SeR (PoliTo interdepartmental centre for service robotics), aims to implement an algorithm able to lead an unmanned ground vehicle in an unknown, closed domestic environment, mapping it and classifying each room encountered in the process. Low cost components and free, open-source software are used to achieve the final result. The robot employed is a TurtleBot3 Waffle, modified to accommodate a Nvidia Jetson AGX Xavier Developer Kit unit, used as main computational platform and on which resides the whole algorithm. An Intel RealSense D435i Depth Camera is used for SLAM. It posses an integrated IMU (*Inertial Measurement Unit*), useful for localize the robot, an infrared depth module, which can visualize the environment and create a map from it, and a color camera for the acquisition of frames of the environment. These frames are then used by a Convolutional Neural Network model to recognize rooms encountered. Since the IMU of the camera just mentioned is not so precise and often leads to lost of tracking, a Intel RealSense T265 Tracking Camera is integrated. Its VIO (*Visual Inertial Odometry*) allows a much more precise localization and tracking of the robot, and is able to greatly improve results provided by the D435i. Actually, the joint use of these two camera is employed, and also encouraged, by Intel for SLAM projects. Initially no other sensors were used, but after several implementations and tests, it is observed that the depth module of the D435i camera does not allow a sufficiently accurate reconstruction of the environmental map for the purposes of this project. In order to achieve better results, a 360 Laser Distance Sensor LDS-01, the default TurtleBot3 lidar unit, is employed. On the software side, advanced SLAM algorithms like *Kimera* and *RTAB-Map* were considered, but after the integration of the lidar unit the final result is obtained with a much more simple program like *Gmapping*. For the implementation of the neural network model *TensorFlow* is used alongside its API *Keras*, while the control of the robot is made through ROS (*Robot Operating System*).

Objectives

The main objectives of this thesis project consist in:

- Implement a Convolutional Neural Network model able to recognize rooms and suitable to run on hardware at the edge;
- Implement a SLAM algorithm able to generate a map image of the environment, then apply a post-process operation on this image in order to obtain an enhanced representation;
- Integrate these two algorithms into one, capable of performing an operation of simultaneous localization, mapping and classification;

The expected result should be a representation of the environment's map, presenting localized labels which indicates types of rooms recognized.

Room Recognition Neural Network

The Convolutional Neural Network model, used for Room Recognition, has at its base a *MobileNet* pre-trained model, with some additional layers in order to adapt it to the new case of study. A dataset of more than 7000 images is gathered and then split into training set and validation set, with the latter containing a fifth of the total. Each of these sets is further divided in the six desired classes of rooms: *Bathroom*, *Bedroom*, *Closet*, *Dining room*, *Living room*, and *Kitchen*. In a pre-processing operation, these images are rescaled, resized and gathered in batches. The dataset is then used to re-train the model and after a series of trial and error attempts, parameters and hyperparameters are tuned and optimized. The final model has an accuracy on the validation set over 85%, and presents an overfitting problem. These results can be considered acceptable given basically two factors: first, the scenes shown in the images do not depict single objects but whole rooms, which are quite complex and, in general, difficult to be analyzed by a neural network. Secondly, for the purpose of this project a number of instances potentially infinite can be run in every room. This means that a single prediction has little importance compared to the total set.

Simultaneous Localization and Mapping

The SLAM algorithm implemented is so simple that a series of programs could be equivalently adopted. How said before, for this project *Gmapping* is chosen, but other algorithms such as *Hector* or *Cartographer* can be used instead, without real differences in the final outcome. Gmapping is a mapping software based on a particular particle filter, called Rao-Blackwellized particle filter, in which each particle carries an individual map of the environment. It takes as input the scan message from the lidar unit and odometric information from the T265 VIO, and provide as output a 2D Occupancy Grid Map. Moreover, in order to work, It requires the three reference frames of lidar, robot and map (that is the one of the physical world or environment in which the robot has to navigate), and the transformations between them. Additionally, one of these reference frames has to track the robot during the navigation. Since the T265 camera and the lidar unit are integral with the robot, these three frames can be considered coincident. Then, it is sufficient to define the reference frame of the map equal to the initial position of the robot, and all the subsequent poses assumed by the robot during navigation will be automatically computed by the SLAM algorithm.

Integration

The final algorithm, which gather the two just described with further implementations, operate in two distinct steps: the Real Time step and the Post Processing step. In the first step, the robot navigate within the environment. The navigation is performed manually, using teleoperation provided by ROS, from an external device. During this phase, the SLAM algorithm is run, in order to construct a first raw map. Simultaneously, at each interval of time, a frame of the environment is acquired by the D435i RGB camera, alongside with its corresponding pose, retrieved from the T265 VIO. At the end of this step, the map image and the frame images are saved in a specific directory, while the corresponding positional information are saved in a CSV file. It is worth noting that during this phase, executed entirely in real time, the only resource consuming operation is represented by the SLAM program. This ensure a pretty lightweight algorithm, able to run on most embedded hardware, without demanding many resources. In the second step, frame images saved before are used by the convolutional neural network model in order to obtain predictions of rooms. These information are then used to update the CSV file. Then, the raw map constructed before is loaded and, in an operation of post processing, a series of computer vision technique and filters are applied to enhance the quality of the visualization and eliminate any gross error. The map is then enlarged in scale, to allow a better and user-friendly visualization. The CSV file is imported and a temporal mean is performed, so that for every n position/prediction samples, a new position/prediction sample is obtained, which is the mean of these n samples. Each prediction of these newly obtained samples is printed on the processed map, in the position associated with it.

Results

In the construction of the environment's map, results obtained from the main test can be seen in Figure 1. The figure to the right is a map reconstruction of the building used for the test: black lines indicate walls, while gray lines represent main obstacles. The center figure depicts the raw occupancy grid map obtained by the Gmapping SLAM algorithm. In an occupancy grid map, black pixels correspond to walls and obstacles,

white pixels to the ground, zones the robot can access and where it can navigate freely, while gray pixels represent unknown portions of the map, zones not yet visited by the robot or simply areas outside the map's limits. The left figure is the final result of the map, obtained from the raw map through a post-process operation, which applied a threshold, a gaussian blur and a Canny filter. How well evident, the final result is much more clear and defined compared to the raw outcome, presents far fewer biased pixels, and allows a better representation of the environment.

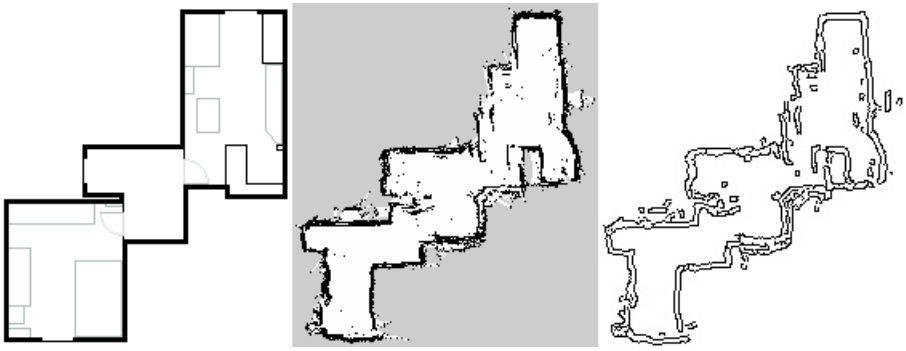

Figure 1: Map visualizations comparison.

In the prediction of frame images, the training set used on the model is a collection of photography of domestic rooms taken from a man-height, and the set used to test the accuracy of such model contains similar images. Testing this model on the field, if the D435i camera is maintained to a man-height, the results are pretty accurate, consistent with the accuracy values found with the validation set. On the other hand, during the navigation of the robot, frames are acquired at a ground-height, and this inevitably influence negatively the accuracy of the model. This problem can be simply solved by retraining the model with a ground-height dataset. The final result can be observed in Figure 2. To the left is shown the actual final labeled map. To the right, for comparison, is presented the same map, with labels obtained using frame images taken maintaining the camera at man-height.

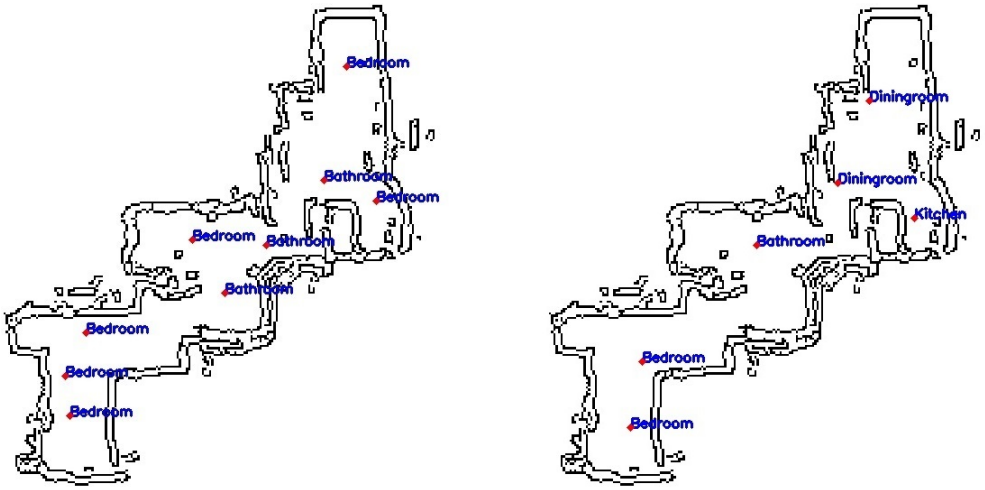

Figure 2: Labeled Map.

Conclusions

This project opens up to a series of future implementations: the algorithm can be further improved by adding an application able to segment rooms using only the occupancy grid map, so it can place a single label for each room, which is the mean value of all the predictions taken in that room. Successively it can be integrated with other hardware and software components, to allow autonomous operations within the environment, for assistance to disabled or elderly users. Furthermore, it is worth noting that, from a user privacy perspective, all the images and information gathered are fully processed right on the robot, and are not shared, by any means, with external entities. This makes it optimal for working in any private environment.