UNIVERSIDAD POLITÉCNICA DE MADRID

POLITÉCNICA

INDUSTRIALES ETSII | UPM

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Industriales

Máster Universitario en Automática y Robótica

CAR CENTRO DE AUTOMÁTICA Y ROBÓTICA

Trabajo Fin de Máster

Evaluation of a Hybrid Localization Algorithm on a Public DataSet of Indoor Environments

Autor: **María Fernanda Rodríguez Eguíbar**

Director: **Paloma de la Puente Yusty**
PhD

Co Director: **María Guadalupe Sánchez-Escribano**
PhD

February, 2022

Index

Figure Index v

Table Index vii

1 Introduction 1

1.1 Localization 1

1.2 Motivation 2

1.3 Objectives 3

1.4 Project Outline 3

2 State of the Art 5

2.1 Introduction 5

2.2 Indoor Localization 6

2.3 Localization Algorithms 7

2.4 Public Data Sets 8

2.5 Evaluation Metrics 9

3 Algorithm 11

3.1 Monte Carlo Localization Algorithm 11

3.1.1 MLC 11

3.1.2 KLD MLC 11

3.2 ROS Implementation 11

3.3 Hybrid Localization 13

3.4 Detector Node 13

4 Public Dataset 15

4.1 Robo@Home 15

4.1.1 Robot Model 15

4.1.2 Sensors 16

4.1.3 Scenarios 17
4.1.4 Data Structures 18
4.1.5 Dataset Analysis 19

5 Tools 21

6 Software Development 23
6.1 Challenges 25
6.2 Laser Data Adapter 26
6.3 RGB-D Data Adapter 26

7 Evaluation 27
7.1 Process 27
7.2 Steps Summary 27

8 Results 29
8.1 Laser Tests 29
8.1.1 Scenario 1: Alma 30
8.1.2 Scenario 2: Anto 30
8.1.3 Scenario 3: Pare 30
8.1.4 Secnario 4: Rx2 30
8.1.5 Scenario 5: Sarmis 30
8.1.6 Laser Results Summary 30
8.2 RGB-D Tests 30
8.2.1 Scenario 1: Alma 30
8.2.2 Scenario 2: Anto 30
8.2.3 Scenario 3: Pare 30
8.2.4 Secnario 4: Rx2 30
8.2.5 Scenario 5: Sarmis 30
8.2.6 RGB-D Results Summary 30
8.3 Comparison 30

9 Conclusions 31

10 Future Lines 33

11 References 35

A User Manual: Evaluate Hybrid Algorithm in ROS 39

Figure Index

3.1 ROS navigation stack elements [23] 13

4.1 Giraff robot with the sensors mounted [15] 16
4.2 Maps of the 5 apartments. Estimated robot path in doted line [15] 17
4.3 Scenario Rx2 has geometric patterns interesting for this project [15] 19
4.4 Scenario Sarmis has variations in light conditions [15] 19

5.1 The tool rqt_graph allows data inspection in .bag files 21

Table Index

2.1 Datasets Comparison 9

4.1 Robo@Home Scenarios of 5 different apartments [15] 17
4.2 ROS AMCL Requirements Analysis 19

6.1 Steps to evaluate and adapt data set 24
6.2 Robo@Home needs to be adapted in ROS 25

Chapter 1

Introduction

1.1 Localization

The localization problem in mobile robotics consists of estimating the pose of the robot, its position and orientation, relative to a known map of the environment from multiple sensor readings over time [1]. Having an adequate localization of the pose of the robot is fundamental to many robot tasks such as robot navigation. When the initial robot pose is known the localization problem becomes pose tracking. If the initial pose is not known the problem is known as global localization and represents a bigger challenge than position tracking.

Probabilistic localization algorithms, based on the Bayes filter, commonly known as Markov localization are solutions for the mobile robot localization problem. A widely used localization algorithm is the extended Kalman filter EKF. However as it is a linearized Gaussian technique it tends to work well only if the position uncertainty is small [1]. Another limitation when using the EKF for localization is that it needs a feature map which is not always possible to obtain.

Alternative localization techniques that can deal with these limitations use different representations of the robot's belief which is the estimate of the current robot pose. Such is the case of the Adaptative Monte Carlo Localization AMCL which is capable of solving global localization and dealing with uncertainty. This algorithm solves localization by using a particle filter that estimates the probability density by a collection of random particles that represent the desired distribution [2].

The work presented in this project is focused on an algorithm based on the AMCL for indoor localization, which estimates the pose of the robot in environments such as houses, museums, or offices. Indoor environments present challenging situations depending on the sensor used. Each type of sensor has its limitations additional to the error and noise that all sensors have. An approach that improves the problems in sensing is using a hybrid localization algorithm where more than one type of sensor is used compensating the advantages and disadvantages of each.

1.2 Motivation

This project aims to continue the work initiated in three previous projects:

- ”Particle filter localization using visual markers based omnidirectional vision and a laser sensor” by Patricia Javierre del Río.

- ”Evaluación de un algoritmo de localización híbrida para un robot móvil dotado de láser y cámara omnidireccional” by Elena Martín Arias.

- ”Localización híbrida de robots móviles utilizando sensores RGB-D: Desarrollo y evaluación de entornos complejos” by Miguel Beteta Fernández

In the work by Patricia Javierre the hybrid algorithm was developed, implemented and tested using a laser and an omnidirectional camera with artificial landmarks. It was tested in simulation and then in a real robot in a corridor. The robot used was Doris, a mobile robot equiped with a laser and omnidirectional camera developed in the Centro de Automática y Robótica from the Universidad Politecnica de Madrid. It was concluded that when using the hybrid algorithm the camera can operate where the detection range of the laser may not be enough, and the laser can be useful when there are not many visual landmarks available.

The work done by Elena Martín also evaluated the algorithm with laser and an omnidirectional camera. Tests where made both in simulation and with the robot Doris mostly in corridors that represent challenging environments. The results in simulation were successful but not on the real robot. It was observed that the localization with the camera was not precise due to illumination conditions and distance between markers [3]. As a future line it was proposed to develop an observation model that can work with any rectangular marks.

Finally, the work done by Miguel Beteta consisted in adapting the algorithm to be used with a RGB-D sensor where the depth measurements from the sensor are interpreted as laser data and the camera images used for the detector of marks. Test were made in simulation using only the depth sensor for localization, then just using the camera and finally using both sensors with the hybrid localization algorithm. In this version the algorithm is capable of detecting rectangular marks on the environment.

In all of these previous works it was shown that the hybrid localization algorithm performs better than traditional localization algorithms both in simulation and with the robot Doris. To further validate the performance of the algorithm it is now proposed to test it on a public data set, with natural rectangular marks for the detector and real-world data, which is more realistic than simulations and an alternative to test the algorithm in other robot models without the need of having them physically and avoiding the physical difficulties of replicating scenarios. Another benefit of using public datasets to evaluate the hybrid algorithm is that this allows to compare results with the work of others.

1.3 Objectives

The global objective of this project is to evaluate the performance of the hybrid localization algorithm on a public dataset, containing more than one sensor, to validate a hybrid approach in localization has better performance than single-sensor approaches. The ideal dataset must have real world data of laser and RGB-D sensors in indoor environments, preferably with light variations, rectangular marks and challenging rooms like hallways and long corridors.

The specific objectives are the following:

- Select Public Dataset
- Analyze dataset and identify challenges
- Adapt data to use in ROS
- Test algorithm with laser and RGB-D data
- Test evaluation metrics
- Compare and conclude

1.4 Project Outline

The following project is divided in 10 chapters. The introduction, this chapter, and the second chapter, the State of the Art, present the problem this project is looking to solve.

The third chapter, Algorithm, presents a description of the algorithm evaluated in this project and its implementation in ROS. The fourth chapter presents a description of the dataset selected and its key aspects.

The fifth, sixth, seventh and eighth chapters, Tools, Software Development, Evaluation Process and Results, describe the practical part of this project, the challenges faced and how they were solved to achieve the main objective which is evaluating the hybrid localization algorithm.

Finally, chapters 9 and 10, Conclusion and Future lines, present the conclusions obtained by the work realized in this project and the possible work lines that can be derived from the results.

Additionally, a step-by-step guide on how to use the code developed is presented in the appendix: "User Manual: Evaluate Hybrid Algorithm in ROS". The source code developed is available at [4].

Chapter 2

State of the Art

The purpose of this chapter is to present an overview of the existing work related to the mobile robot localization problem in indoor environments. An overview of the available public robotic data sets, evaluation metrics, and their relevance to this project is presented.

2.1 Introduction

The reproduction of experimental results is key to obtaining advances in a research field. In robotics, reproducing experimental results is hard to achieve due to the difficulties of replicating the exact environment of the previous test and having access to the same robot model, tasks which can be costly and time consuming. Indoor robot navigation is closely related to robot vision where robots are still missing good perceptual systems, which prevents deployment into real-world applications [5]. Robotic sensing data and images are hard to obtain as data acquisition is time consuming and robot platforms are expensive. The progress is slowed down by the absence of common data to precisely represent the world perceived by the robots in real-world applications.

An alternative to reproduce robotic experimental results without having to use a physical model consists of using simulation environments that allow to recreate tests by simulating a representation of the original environment. The problem with simulation is that there is usually a significant gap in realism between real-world scenes and simulated environments [6], which leads to errors at the moment of implementation of a physical model in real world testing.

An alternative to solve this problem is open-source, low-cost, large-scale datasets and benchmarks [7]. Especially datasets of real world data which represent all challenges the robot experiences when navigating in the wild. Datasets can allow to evaluate other robot controlling systems without really constructing the robot, a task which is economically convenient and time efficient [7]. This allows for comparison and sharing of results, which is key to obtain advances in the research field .

2.2 Indoor Localization

In the mobile robot field, indoor navigation presents different challenges than exterior navigation. While exterior robots can handle localization with sensors like GPS, sensors more suitable for indoor mobile robots are cameras and lasers. As indoor navigation is closely related to robot vision, one of the main challenges for indoor localization is light variations in the environment, poor illumination, varying camera perspectives and occlusions [8].

Another of the great challenges of indoor navigation is the high variability in the environment structure. Indoor navigation seeks to implement robot navigation tasks within buildings such as homes, offices, museums, hospitals, shopping malls, etc. where there is a high variability inside the environment structure. This presents a problem as many robot navigation systems rely on a static representation of the environment such as occupancy grid maps or topological maps [9].

Indoor mobile robots must operate autonomously in changing environments over an extended period of time where the structure of the scene is dynamic and changing daily. In real world applications, these changes can happen both by human activities and natural factors. Inside a home scenario, most objects are movable, replaceable, or deformable, and the visual features may be significantly different in successive days [10]. These dynamic variations in the environment pose significant challenges to the pose estimation accuracy and to the long-term deployment and correct functioning of the mobile robot.

It is important to use data sets that correctly represent the real world to further deploy successfully the robotic application. In the robot grasping field, the study presented in [6], demonstrated that using real world data from other scenarios rather than labs to train robots can improve the performance even compared with lab training, as they obtained a 43.7% improvement over a model trained with data collected in the lab.

2.3 Localization Algorithms

The localization problem in robotics is the estimation of the robot pose: its location and orientation relative to a reference frame. The difficulty of the localization estimation can vary given the conditions of the environment and robot. If the initial robot pose is known, the localization problem becomes position tracking where the errors in odometry must be compensated as the robot moves through the environment to know its location. If the initial pose is not known, then the robot must determine it and this is called global localization. In this case, the error in the robot's estimate can not be assumed to be small and the robot must be able to handle multiple, distinct hypotheses of its location [2]. In dynamic environments, where the structure of the environment is constantly changing, as it is the case for indoor mobile robots applications, the localization problem becomes even harder.

The most used localization algorithms work with probabilistic approaches based on the Markov Localization [11]. Two very used probabilistic algorithms are the *Extended Kalman Filter* and the *Monte Carlo Localization*. Kalman filters estimate posterior distributions of robot poses based on sensor data. This algorithm makes restrictive assumptions such as considering the noise in the sensors is Gaussian distributed. While Kalman filters offer elegant and efficient solutions for localization, its restrictive assumptions make it unusable for the global localization problem [2].

The Monte Carlo Localization algorithm, also known as *Particle Filter*, was introduced in 1993 as a numerical approximation to the nonlinear Bayesian filtering problem [12]. The MCL is a recursive Bayes filter that estimates robot poses based on sensor data. The key idea of a Bayes filter is to estimate the state of a dynamical system from sensor measurements. It must estimate a probability density over the state space conditioned on the data. The MCL describes this probability density by a collection of random particles that represent the desired distribution [2].

The MCL algorithm consists of two phases. To initialize the filter, it starts at time 0 with a random sample. The first phase, the prediction phase, an initial set of random particles is computed and the motion model is applied to each of this particles. In the second phase, the update phase, the sensor measurements are taken into account and each of the particles is given a weight depending on the likelihood of the measurement. A new set of samples is generated where the samples with the higher probability are selected. These prediction and update phases are repeated recursively [13]. The advantages of the MCL are that it can handle non-gaussian noise in sensor readings, it can adapt to the available computational resources by controlling the number of samples, and it is easy to implement.

2.4 Public Data Sets

There are many available datasets for indoor robotics. For a dataset to be useful to evaluate and compare results, it must have a reliable ground truth. A trustworthy and accurate ground truth is essential for benchmarking self-localization and mapping algorithms [14]. Additionally for a dataset to be interesting for this project, it must have both laser and camera data or RGB-D with depth perception measurements.

An interesting computer vision dataset and benchmark for indoor navigation is the NavigationNet: A Large-scale Interactive Indoor Navigation Dataset [7]. This dataset consists of photos captured by 8 cameras in a human-sized robot which covers $1500m^2$ of indoor area with bedrooms, studies, meeting rooms, etc. and its main purpose is to train deep reinforcement learning on scene understanding based indoor navigation. However, this dataset is not interesting for this project as it does not contain laser information.

A dataset which has both laser and camera data is the Robot@home, a robotic dataset for semantic mapping of home environments [15]. This dataset presents raw and processed sensory data from home settings consisting of 87,000 time-stamped observations gathered by a mobile robot with four RGB-D cameras and a 2D laser scanner. The data was taken in five apartments and comprises a total of 36 rooms rich in contextual information. These characteristics make this dataset interesting to evaluate the common challenges of indoor navigation: light variations, occlusions, viewpoint variation, and cluttered and dynamic rooms.

A dataset that has both laser and omnidirectional camera is the one from the Bicocca Indoor Dataset [16] from The Rawseeds Project. This dataset is taken in a pair of buildings from the Universita di Milano-Bicocca. The scenarios taken in this dataset are the floor of an office building and a bridge connecting both buildings. This dataset has a rich source of features: corridors, hallways, doors, passageways, etc. and has ground truth data.

Another interesting dataset is COLD: The Cosy Localization Database [17]. This dataset consists of data gathered in three indoor laboratory environments from the University of Ljubljana, Slovenia, the University of Freiburg, and the German Research Center for Artificial Intelligence in Germany. The data was recorded using various mobile robots, perspective cameras, omnidirectional cameras, laser scans, and odometry data. The images were acquired under different illumination conditions.

A dataset that has RGB-D data with depth perception is An indoor RGB-D dataset for evaluation of robot navigation algorithms [18]. This dataset was taken with a Kinect sensor using a WiFiBot mobile robot. There was special care in this dataset to make sure that the data registered was Kinect friendly and that objects in the scene appear at no more than a 6m distance to assure depth perception data is available. This dataset also has a reliable ground truth as special care to synchronize the ground truth data from the overhead cameras and the robot-mounted sensors was taken having a 1ms accuracy.

Below, a table with a summary of the discussed datasets and its main attributes.

Dataset	**Environment**	**Sensors**	**Ground truth**
NavigationNet	Bedrooms, studies, meeting rooms	Cameras	Yes
Robo@home	36 rooms in 5 apartments	RGB-D and laser	Yes
Rawseeds	Office building and hallways	Laser and omnidirectional camera	Yes
COLD	Indoor laboratory	Laser and omnidirectional camera	Yes
Indoor RGB-D	Indoor laboratory	RGB-D	Yes

Tabla 2.1 Datasets Comparison

2.5 Evaluation Metrics

Another key aspect to create advances in the field is to be able to measure results and compare data on the same metrics between projects. It is important to have common evaluation tools to compare when using public datasets. In the evaluation of robot navigation algorithms, it is important to compare the estimated trajectory with the real trajectory, which means comparing the output of the system with the ground truth, hence the importance of having reliable ground truth.

Some of the most common evaluation metrics in mobile robotics are the absolute trajectory error ATE and the relative pose error RPE. The RPE measures the difference between the estimated motion and true motion. It is useful to evaluate odometry systems or drift. On the other hand, ATE does not evaluate relative pose differences but first aligns the two trajectories and then evaluates directly the absolute pose differences [19].

Other existing evaluation metrics are the auto-localization error and the complexity estimation error [3]. The auto-localization error evaluates the localization algorithms building a map and then compares the estimated position with the ground truth. The complexity estimation evaluates how the computing time of an algorithm increases when the amount of entry data to be processed also increases.

Chapter 3

Algorithm

3.1 Monte Carlo Localization Algorithm

The Monte Carlo Localization algorithm, as was introduced in chapter 2, is based on particle filters and represents the belief bel(x_t), which is a hypothesis of the current location of the robot, as a set of particles. The algorithm has two main phases prediction and correction.

3.1.1 MLC

3.1.2 KLD MLC

This is the version implemented in ROS and the version in which the hybrid algorithm is based.

3.2 ROS Implementation

The Robot Operating System, commonly known as ROS, is an open source meta-operating system. It works as a framework on top of Linux such as Ubuntu and Debian. It allows to abstract the hardware from the software to make it possible to create programs for robots without having to deal with the specifics of hardware. The main purpose of ROS is to create code that can be shared and reused among different robots [20]. It provides libraries, drivers and tools such as visualizers for debugging to help developing robotic applications.

All ROS applications are handled in modular packages that allow its portability. Inside the packages there are nodes, which are programs, that are executed individually through special files called launch files and communicate between each other through channels called topics. Several nodes can be sending messages through several topics at the same time working as a multi thread communication. The main program where all of this happens is the roscore where all the nodes, topics and communication between them are handled. ROS must work within a specific workspace in the system generally called the catkin workspace.

The version of ROS that this algorithm is built on is ROS kinetic [21] released on 2016. One of the main advantages of using ROS are its great debugging tools like Rviz, a 3D visualization tool, which allows to visualize data in a graphic environment, a task that is extremely relevant when working with a dataset where there is no access to the robot model or test environment. Several packages of ROS are used in this project, described in chapter 5, and the hybrid algorithm is based upon the localization node of the ROS navigation stack.

Navigation Stack
The ROS navigation stack is a collection of packages with the implementation of several navigation related algorithms, than can be used for autonomous navigation in mobile robots [22]. It is meant for 2D maps and can execute algorithms for slam, path planning, localization, map handling and obstacle avoidance that take information from odometry, sensor data and a goal pose to output safe velocity commands sent to a mobile base with differential or holonomic drive.

The navigation stack set up can be a challenging task as ROS assumes that the robot already has certain characteristics. In figure 3.1 the input and output configuration of the navigation stack can be seen where the blue components must be configured for every robot. The hybrid algorithm is built based on the AMCL node.

AMCL
The AMCL node is a probabilistic localization system based on the Adaptative Monte Carlo, or KLD-Monte Carlo, localization for tracking robots moving in a 2D known environment through the use of a particle filter. It uses the laser, odometry and Monte Carlo algorithms described in [1]. The ROS AMCL current implementation only works with data from laser scans [24], but has the possibility to be extended to work with other sensor data, which is the purpose searched with the implementation of the hybrid algorithm

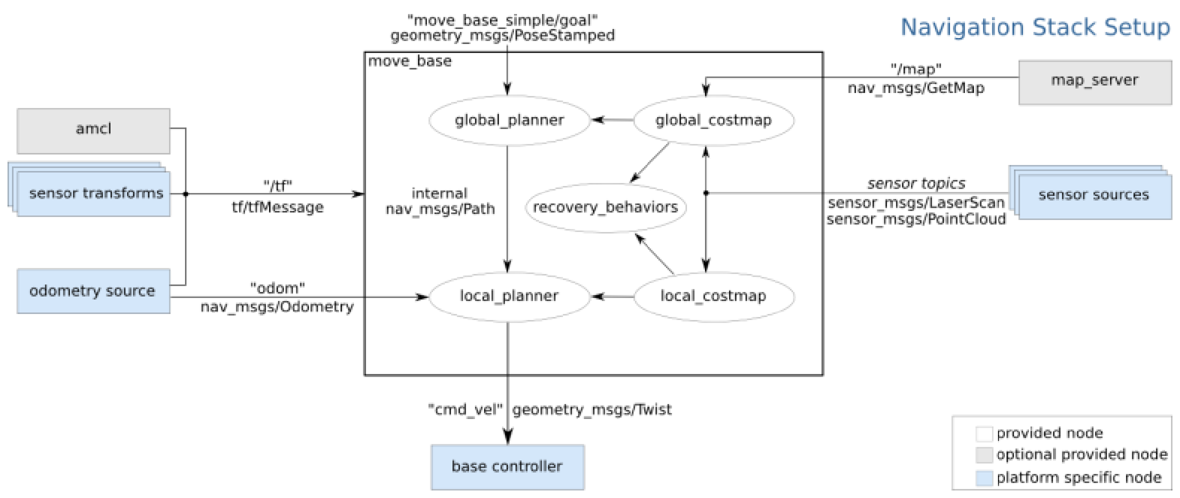


Figure 3.1 ROS navigation stack elements [23]

evaluated in this project to improve localization quality.

The inputs for this node are an occupancy grid map, laser scans, and transform messages, information about relationships between coordinate frames of the robot elements, including odometry data. The outputs are pose estimates represented with a particle cloud. The parameters to configure the node are set in the launch file and there are three main parameter categories: overall filter, laser model and odometry model; the particle filter will be initialized according to the parameters provided. The four important pre-requisites to be fulfilled for running the AMCL are the following:

- The transform frame information between world, odometry, robot and sensors.
- Odometry data for the transform frame tree.
- An occupancy grid map of the environment.
- Laser data.

3.3 Hybrid Localization

TO BE FINISHED

3.4 Detector Node

TO BE FINISHED.

Chapter 4

Public Dataset

4.1 Robo@Home

The Robo@Home data set is the public dataset selected to be used in this project. This dataset is a collection of raw and processed sensory data collected by a mobile robot in indoor domestic settings. The sensors used to collect the data were 4 RGB-D cameras and a laser mounted in the robot platform. The data contains over 87,000 timestamped sensor observations which are about 75 min of recorded data. This dataset is originally designed for semantic mapping algorithms through the categorization of objects or rooms but can also serve for typical robotic problems like localization as the robot poses can be accurately estimated by the sequence of 2D scans.

4.1.1 Robot Model

The commercial robot Giraff, from Giraff technologies [25], was the robot platform used to collect the data. This robot consists of a motorized wheeled platform with a videoconferencing set designed to assist the elderly in indoor environments. The kinematic model of the robot is a differential drive mobile system. The sensors used to take the data were mounted in the Giraff robot as can be seen in the figure 4.1. The origin of the robot frame was considered as the center of the robot base [15].

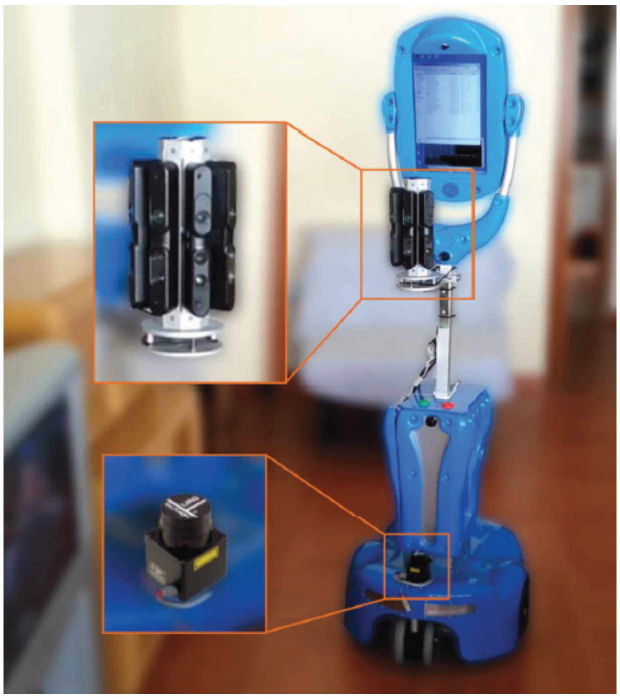


Figure 4.1 Giraff robot with the sensors mounted [15]

4.1.2 Sensors

RGB-D Cameras

The RGB-D cameras used to take the data where Asus XTion Pro Live RGB-D. Each one of these with a 58° vertical and 45° horizontal field of view, are capable of providing synchronized intensity and depth images with a resolution of (640X480) VGA. The cameras were mounted vertically on a rig with a difference of 45° between each other which allows no overlap between the field of view of the four cameras avoiding interferences and creating a 180° field of view when combining the images of the four cameras. The rig where the cameras are mounted was placed in the front part of the robot at a height of 0.92m.

Laser Scanner

The laser used to take the data was a Hokuyo URG-04LX-UG01 scanner. This laser is capable of surveying 2D planes with a field of view of 240° and an angular resolution of 0.352°. The laser was mounted at a height of 0.3m above the ground, at this height it does not perceive the robot while being able to survey an horizontal plane to the floor at its maximum field of view.

The sensors where both intrinsically and extrinsically calibrated and the methods used are described in [15].

4.1.3 Scenarios

A total of 36 rooms in 5 different apartments were fully inspected teleoperating the robot through the houses, at a maximum speed of 0.1m/s and 10deg/s for linear and angular velocity, while the sensors recorded. The scenarios representing each one of the apartments were called: Alma, Anto, Pare, Rx2 and Sarmis. Maps of each one of these appartments can be seen in figure 4.2. The data was gathered in 7 sessions, 3 sessions for the scenario "Sarmis" and one for each other of the scenarios. Each session contains a number of sequences of RGB-D and laser data. The scenarios include characteristics such as cluttered rooms, occlusions, changing viewpoints, variations in the environment set up and distinctive patterns were placed for each of the scenarios as can be seen in table 4.1

Scenario	# Rooms	Rooms	Patterns Placed
Alma	5	Bathroom, 2 bedrooms, kitchen, living room	Teddy bears
Anto	9	2 Bathrooms, 3 bedrooms, corridor, dressing room, Kitchen, Living room	Fruits
Pare	10	2 Bathrooms, 3 bedrooms, hall, corridor, kitchen, 2 living rooms	Numerical patterns
Rx2	4	Bathroom, bedroom, kitchen, living room	Geometric figures
Sarmis	8	2 Bathrooms, corridor, kitchen, living room, 3 bedrooms	Variations in light

Tabla 4.1 Robo@Home Scenarios of 5 different apartments [15]

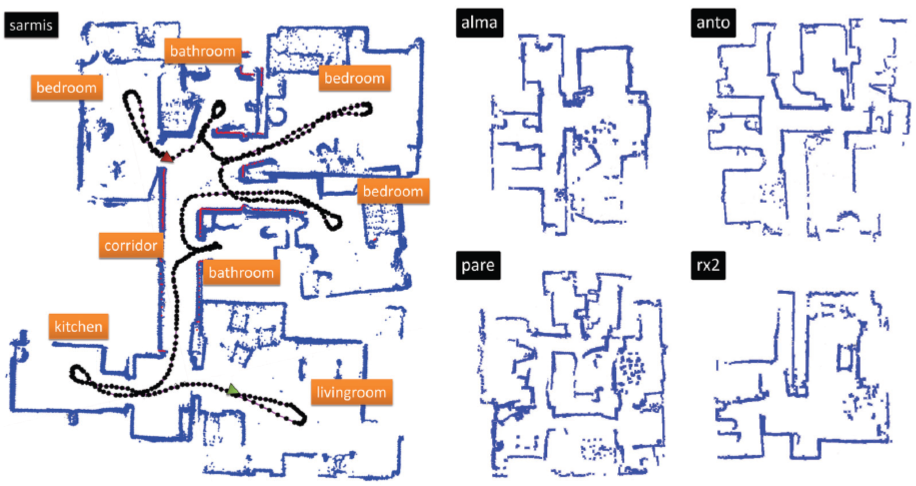


Figure 4.2 Maps of the 5 apartments. Estimated robot path in doted line [15]

4.1.4 Data Structures

MRPT

The Mobile Robot Programming Toolkit or MRPT is a set of libraries and applications for robotics written in C++ [26], and it is the set of tools that was used to record and process all the dataset data. MRPT defines some file formats with the purpose of creating and exchanging robotic raw datasets. The native MRPT files that are used in this dataset are .rawlog and .simplemap. The rawlog files are a binary format that can store raw observations from multiple sensors for posterior processing. The simple map files are a set of pairs of poses and their associated observations [27].

The data available in this dataset is divided in raw and processed data, where the sensor readings are the raw data and the processed data are the scene maps. The data taken from the 5 sensors was stored simultaneously with the rawlog-grabber application from the Mobile Robot Programming Toolkit which time stamps the data and compresses it into a single binary file, a rawlog file, and automatically translates it to human-readable information (plain text files for the laser and png images for the RGB-D cameras).

Raw Data

- Laser Data
  The laser data used in this project is the one in human readable files format, plain text files with a sequence of laser readings. The observations were saved at a frequency from 1.25Hz to 10Hz for the 2D laser scanner
- RGB-D Data
  The RGB-D data used in this project is the one in the human readable files format, depth and intenstiy png files. The frequency that these images were taken varied from 1Hz to 11Hz for each camera.

Processed Data

The processed data are the 2D geometric maps of the 5 apartments. The dataset contains a total of 41 maps, 36 for the rooms and 5 for the full apartments. These maps were created registering the observations from the laser scanner with the ipc-SLAM application within MRPT. The logs generated during the SLAM process are included in the dataset and these include additional information such as the estimated path of the robot, which will be used in this project as ground truth for the localization.

4.1.5 Dataset Analysis

This dataset contains laser data that can be used to test the traditional AMCL and RGB-D data that can be used to test the hybrid AMCL so it is a great option for comparing the performance between algorithms. However more details regarding the data have to be considered to determine if the dataset is useful. The table 4.2 shows the basic requirements to run the AMCL algorithm in ROS and how it matches with the dataset files:

AMCL Requisite	ROS Format	Dataset	Format
Laser Data	ROS laser msg	Hokuyo Laser data	.txt and .rawlog
Odometry Data	ROS odom msg	Not available	Not available
TF information	ROS TF msg	Sensor position info.	data in cm
2D Grid map	.yaml and .pgm	2D Maps	.png and .simplemap

Tabla 4.2 ROS AMCL Requirements Analysis

The disadvantages this dataset has are the missing odometry data and the file formats needed in ROS that are different than the formats provided in the dataset. The process of converting the provided data into usable ROS format is described in the next chapter. Regarding odometry, data from the encoders of the Giraff robot is not included in the dataset as the encoders were giving data with excess of noise [28]. An alternative it to use an existing ROS node called laser_scan_matcher capable of generating odometry data from laser readings.
The advantages of this dataset are the geometric patterns placed in the Rx2 scenario, as can be seen in figure 4.3, that can be tested with the detector algorithm which is capable of detecting natural rectangular marks. Also the ligth variation in the Sarmis department, as can be seen in figure 4.4, represent one of the greatest challenges when using cameras in robotic applications.

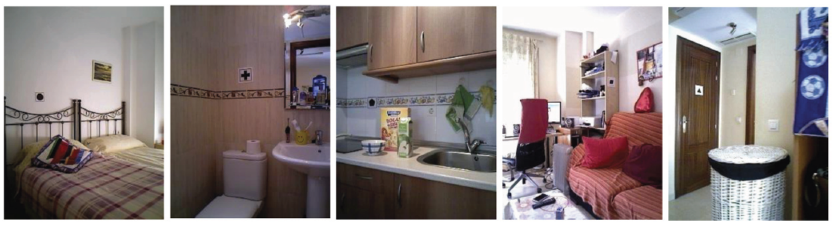


Figure 4.3 Scenario Rx2 has geometric patterns interesting for this project [15]

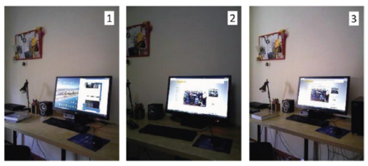


Figure 4.4 Scenario Sarmis has variations in light conditions [15]

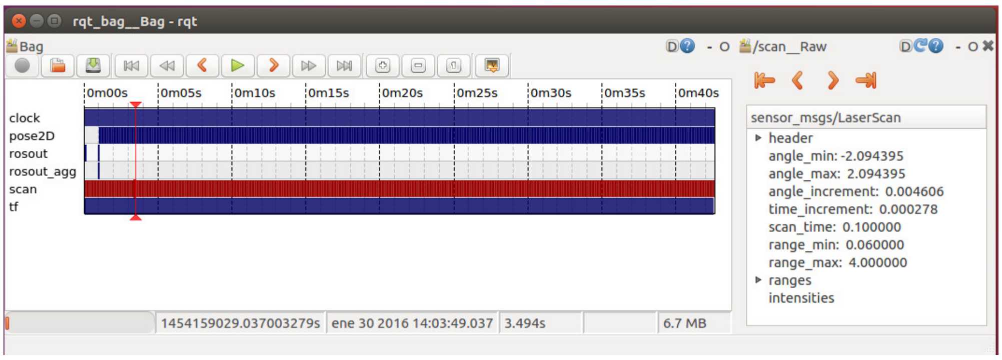
Chapter 5

Tools

In the process of adapting the dataset to be used in ROS some pre-existing tools were used, in addition to the software developed described in chapter 6.

Rosbag

The rosbag node is a set of tools to record and play data in ROS topics. [29]. This is achieved when working with *bags*, a ROS file format for storing message data. This files have as extention .bag and are created using the rosbag tools, that subscribes to a ROS topic and stores the serialized message data in a file as it is received. These files can be played back to topics as if the sensor data was being streamed. It is a very common tool used in ROS, so it ensures portability. There are visualization tools like rqt_bag which allow visualization of the data in the bag files which makes it a great, documented, flexible and supported set of tools. This was the tool selected and used to convert and manage the data from the dataset.



```
sensor_msgs/LaserScan
▸ header
  angle_min: -2.094395
  angle_max:  2.094395
  angle_increment:  0.004606
  time_increment:  0.000278
  scan_time:  0.100000
  range_min:  0.060000
  range_max:  4.000000
▸ ranges
  intensities
```

```
1454159029.037003279s | ene 30 2016 14:03:49.037 | 3.494s | | 6.7 MB
```

Figure 5.1 The tool rqt_graph allows data inspection in .bag files

Laser Scan Matcher

The laser scan matcher is a package in ROS that works as an incremental laser scan registration tool. It can serve as a stand-alone odometry estimator as it does not need external input from odometry data. The package allows to scan match between consecutive sensor_msg/LaserScan messages, and publish the estimated position of the laser as a geometry_msgs/Pose2D or a tf transform [30]. This package is really useful to generate odometry data, which is a pre-requisite to run localization in ROS, from laser scans when there is laser data available and no odometry. In this project this node was used to create the odometry data not available in the public data set.

Pointcloud to LaserScan

This ROS package is able to convert a 3D point cloud to a 2D laser scan. This can make depth data from RGB-D devices appear like a laser scanner for 2D applications. It creates a virtual laser scan from the RGB-D pointcloud [31]. This package is useful to simulate a laser and a camera when using a single RGB-D sensor. In this project it was used to adapt the RGB-D depth info to be used with the AMCL which accepts laser data messages.

Chapter 6

Software Development

?? The next section presents the process and challenges during the adaptation of the dataset to be compatible with ROS and fulfill the requirements needed by the algorithm as well as a description of the software developed to cover these needs. The procedure to evaluate if a dataset is ready to be used with ROS can be summarized in the following steps:

1- Verify the robot model
It is important to make sure you have a general knowledge of the robot being used and its kinematic model, as the ROS AMCL only works with differential and holonomic models. Any other type of robot will not be correctly localized with the algorithm. This information is required to set up the parameters in the configuration file of the AMCL.

2- Verify robot setup
It is important to have a clear understanding of the sensors the robot used, how many there were, in what part of the robot they were mounted as this information will be needed to set up the transform frames TFs in ROS which is a requisite to run the AMCL, as well as the position of the base frame of the robot base.

3- Verify sensor data
The sensor data available in the dataset must be able to be played in topics. If the data is not in a ROS format like rosbag, then it will need to be pre-processed with the software created in this project. It is possible to transform data in .txt files to ROS format. One of the requisites of the AMCL is having laser data or any other data that can be processed

to be interpreted as a laser like RGB-D depth information.

4- Verify odometry

Another of the pre-requisites of the ROS AMCL is to have odometry data. It is commonly generated with wheel encoders. If the case appears where no odometry data is available, like in the Robo@Home dataset, it will need to be generated as the AMCL can not run without this data.

5- Verify Occupancy Grid Map

Finally, another of the fundamental pre-requisites for running the localization algorithm is counting with a good quality occupancy grid map of the environment to be tested. The available map must be transformed into the acceptable ROS format, a .pgm and .yaml files, if it is not in this format.

A summary with all the steps to evaluate a dataset and identify its needs to be adapted to the ROS AMCL hybrid algorithm can be seen in table 6.1:

Steps to Verify Dataset	**Description**
1. Robot Model	Define kinematic model of robot: differential or holonomic can work with the AMCL.
2. Robot Setup	Define robot base frame and sensors relative position to this frame.
3. Sensor Data	Make sure there is laser data available, or other sensor data that can be interpreted as depth.
4. Odometry	Make sure there is data from wheel encoders or similar. Define odometry frame.
5. Map	Make sure there is an occupancy grid map of the environment available. Define map frame.

Tabla 6.1 Steps to evaluate and adapt data set

6.1 Challenges

An analysis of the dataset was carried according to the steps described in tables 4.2 and 6.1. The combined information in this tables with the identified needs of the Robo@Home dataset to be used in ROS can be seen in table 6.2.

Steps to Verify Dataset	**Description**
1. Robot Model	Jiraff robot has a differential drive kinematic system. This info. will be used in the .launch file parameter configuration for the AMCL.
2. Robot Setup	As described in [15] the robot base frame was considered the center of the robot base. So the position in meters (x,y,z) and orientation in degrees (yaw,pitch,roll) for all the robot elements are the following [32]: • Robot base (0,0,0)m (0,0,0)° • Laser (0.205,0,0.31)m (0,0,0)° • RGB-D1 (0.285, 0.0, 1.045)m (0, 0, 90)° • RGB-D2 (0.271,-0.031,1.045)m (-45,0,90)° • RGB-D3 (0.271, 0.031,1.045)m (45, 0,90)° • RGB-D4 (0.240,-0.045,1.045)m (-90, 0,90)°
3. Sensor Data	Hokuyo laser scan data is available in .txt files that will be converted into ROS laser msgs using the software described in 6.2. The RGB-D data available in .png images for the depth and intensity sensors will be converted to ROS depth and camera msgs using the software described in 6.3.
4. Odometry	The odometry data will be generated from the laser data using the tool 'Laser Scan Matcher' described in 5. The odometry frame will be defined as the laser frame, (0.205, 0, 0.31)m, (0,0,0)°.
5. Map	The .png images and .simplemap files from the dataset will be adapted to .pgm and .yaml. The map frame will be defined in the origin (0,0,0)m.

Tabla 6.2 Robo@Home needs to be adapted in ROS

The results and challenges faced in each one of the steps described in table 6.2 is described below:

TO BE FINISHED.

6.2 Laser Data Adapter

6.3 RGB-D Data Adapter

Chapter 7

Evaluation

The next chapter presents the process done in this project to test and evaluate the algorithm with the public dataset Robo@Home as well as the tools used.

7.1 Process

Once all the setup, configuration files and data from the dataset was adapted to ROS the next step was to run the AMCL and Hybrid AMCL and evaluate the results with the metrics described in the state of the art, ATE and RPE.
TO BE FINISHED.

7.2 Steps Summary

TO BE FINISHED.

Chapter 8

Results

8.1 Laser Tests

A total of n test were done following the procedure described in Chapter 6. The results are the following:
-enlistar los resultados según escenario

8.1.1 Scenario 1: Alma

8.1.2 Scenario 2: Anto

8.1.3 Scenario 3: Pare

8.1.4 Secnario 4: Rx2

8.1.5 Scenario 5: Sarmis

8.1.6 Laser Results Summary

8.2 RGB-D Tests

A total of n test were done following the procedure described in Chapter 6. The results are the following:
-enlistar los resultados según escenario

8.2.1 Scenario 1: Alma

8.2.2 Scenario 2: Anto

8.2.3 Scenario 3: Pare

8.2.4 Secnario 4: Rx2

8.2.5 Scenario 5: Sarmis

8.2.6 RGB-D Results Summary

8.3 Comparison

Chapter 9

Conclusions

Chapter 10

Future Lines

Test this algorithm with a public dataset with omni directional cameras.
Adapt to latest version of Ubuntu and ROS.

Chapter 11

References

[1] S. Thrun, W. Burgard, and F. Dieter, *Probabilistic Robotics*, 1st ed. MIT Press, 2006.

[2] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots", *Artificial Intelligence*, vol. 128, no. 1, pp. 99–141, 2001, ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(01)00069-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370201000698.

[3] E. M. Arias, "Evaluación de un algoritmo de localización híbrida para un robot móvil dotado de láser y cámara omnidireccional", 2020, [Online]. Available: http://oa.upm.es/58273/.

[4] M. F. Rodríguez, *Ros amcl hybrid localization repository*, 2021. [Online]. Available: https://github.com/fernandaroeg/ROS_AMCL_Hybrid_Localization.

[5] M. R. Loghmani, B. Caputo, and M. Vincze, *Recognizing objects in-the-wild: Where do we stand?*, 2018. arXiv: 1709.05862 [cs.RO].

[6] A. Gupta, A. Murali, D. Gandhi, and L. Pinto, *Robot learning in homes: Improving generalization and reducing dataset bias*, 2018. arXiv: 1807.07049 [cs.RO].

[7] H. Huang, Y. Shen, J. Sun, and C. Lu, *NavigationNet: A large-scale interactive indoor navigation dataset*, 2018. arXiv: 1808.08374.

[8] S. Lee, A. M. Naguib, and N. U. Islam, "3d deep object recognition and semantic understanding for visually-guided robotic service", in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 903–910. DOI: 10.1109/IROS.2018.8593985.

[9] L. Nardi and C. Stachniss, *Long-term robot navigation in indoor environments estimating patterns in traversability changes*, 2019. arXiv: `1909.12733 [cs.RO]`.

[10] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song, Y. Guo, Z. Wang, Y. Zhang, B. Qin, W. Yang, F. Wang, R. H. M. Chan, and Q. She, *Are we ready for service robots? the openloris-scene datasets for lifelong slam*, 2020. arXiv: `1911.05603 [cs.RO]`.

[11] P. Javierre, B. P. Alvarado, and P. De La Puente, "Particle Filter Localization Using Visual Markers Based Omnidirectional Vision and a Laser Sensor", *Proceedings - 3rd IEEE International Conference on Robotic Computing, IRC 2019*, pp. 246–249, 2019. DOI: `10.1109/IRC.2019.00045`.

[12] F. Gustafsson, "Particle filter theory and practice with positioning applications", *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, pp. 53–82, 2010.

[13] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots", in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, 1999, 1322–1328 vol.2. DOI: `10.1109/ROBOT.1999.772544`.

[14] S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti, and P. Taddei, "Rawseeds ground truth collection systems for indoor self-localization and mapping", *Autonomous Robots*, vol. 27, no. 4, pp. 353–371, 2009, ISSN: 09295593. DOI: `10.1007/s10514-009-9156-5`.

[15] J. R. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez, "Robot@home, a robotic dataset for semantic mapping of home environments", *International Journal of Robotics Research*, 2017.

[16] Rawseeds Project, *Dataset "Bicocca (indoor)"*, http://www.rawseeds.org/rs/datasets/view/6.

[17] A. Pronobis and B. Caputo, "Cold: The cosy localization database", *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 588–594, 2009. DOI: `10.1177/0278364909103912`. eprint: `https://doi.org/10.1177/0278364909103912`. [Online]. Available: `https://doi.org/10.1177/0278364909103912`.

[18] A. Schmidt, M. Fularz, M. Kraft, A. Kasiński, and M. Nowicki, "An indoor rgb-d dataset for the evaluation of robot navigation algorithms", in *Advanced Concepts for Intelligent Vision Systems*, J. Blanc-Talon, A. Kasinski, W. Philips, D. Popescu, and P. Scheunders, Eds., Cham: Springer International Publishing, 2013, pp. 321–329, ISBN: 978-3-319-02895-8.

[19] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems", in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 573–580. DOI: `10.1109/IROS.2012.6385773`.

[20] *Ros introduction.* [Online]. Available: `http://wiki.ros.org/ROS/Introduction`.

[21] *Ros kinetic.* [Online]. Available: `http://wiki.ros.org/kinetic`.

[22] *Ros wiki: Navigation.* [Online]. Available: `http://wiki.ros.org/navigation`.

[23] *Ros wiki: Navigation stack setup.* [Online]. Available: `http://wiki.ros.org/navigation/Tutorials/RobotSetup`.

[24] *Ros wiki: Amcl.* [Online]. Available: `http://wiki.ros.org/amcl`.

[25] *Giraff robot*, http://www.giraff.org/.

[26] *Mobile robot programing toolkit.* [Online]. Available: `https://www.mrpt.org`.

[27] *Robotic file format standards in mrpt.* [Online]. Available: `https://www.mrpt.org/robotics_file_formats/`.

[28] Robo@HomeMeetingPoint, *Issue 4*, 2020. [Online]. Available: `https://github.com/jotaraul/robot-at-home_meeting-point/issues/4`.

[29] G. Staples, *Ros wiki: Rosbag*, 2020. [Online]. Available: `http://wiki.ros.org/rosbag`.

[30] N. Varas, *Ros wiki: Laser_scan_matcher*, 2019. [Online]. Available: `http://wiki.ros.org/laser_scan_matcher`.

[31] P. Bovbel, *Ros wiki: Pointcloud_to_laserscan*, 2015. [Online]. Available: `http://wiki.ros.org/pointcloud_to_laserscan`.

[32] *Robo@home dataset.* [Online]. Available: `http://mapir.isa.uma.es/mapirwebsite/index.php/mapir-downloads/203-robot-at-home-dataset.html`.

Appendix A

User Manual: Evaluate Hybrid Algorithm in ROS

Pre-requisitos
1. Sistema operativo Ubuntu 16.04 Xenial
link a instalación
2. Instalación de ROS kinetic
link a instalación

Environment Setup
1. crear catkin workspace
2. clonar algoritmo dentro de catkin_ws/src
3. catkin_make desde catkin_ws

Additional Libraries and Dependencies Installation
1. Install laser_scan_matcher

Hybrid Localization Algorithm Compilation
1. Clonar repositorio dentro de catkin_ws/src
2. Catkin make

Algorithm Evaluation
1. Inizializar ros, roscore

2. Abrir Rviz