LITERATURE REVIEW
CO-OPERATIVE MAPPING AND LOCALIZATION OF AUTONOMOUS ROBOTS

LYNTON DICKS
Supervisor: DR. KAREN BRADSHAW
Department of Computer Science, Rhodes University

March 2011
## Contents

1 Introduction .............................................. 4
2 Simultaneous Localization and Mapping .......... 4
3 Co-operative Mapping and Localization ........ 4
4 Hardware .................................................. 5
   4.1 Encoders ........................................... 5
   4.2 Sensors ............................................. 6
5 Odometry Data ............................................. 7
6 Landmark Extraction .................................... 7
   6.1 Unique versus Non-unique Landmark Observations 7
   6.2 Time between Landmark Observations ............ 8
   6.3 Static versus Dynamic Environments ............ 8
7 Data Association ......................................... 9
   7.1 Multi-Hypothesis Tracking with Pruning ........ 9
   7.2 Joint Compatibility Branch and Bound test .... 9
8 Localization Algorithms ............................... 9
   8.1 Kalman Filter (EKF) .............................. 10
   8.2 Monte Carlo Localization ......................... 10
   8.3 Markov Localization ............................... 11
9 Map Merging ............................................. 11
## List of Figures

1. The Fischertechnik robot that is used in this project complete with Controller, two ultrasonic sensors, two encoder motors, and Bluetooth. (This image is taken from [3]) ........................................ 6

2. The Fischertechnik Encoder Motor used to record the rotation of the motor’s output shaft to use in odometry. (This image is taken from [3]) ........................................ 7

3. The Fischertechnik ultrasonic range finder with an arc width of 60 degrees, maximum range of 4 meters, and a scan resolution of 1 centimeter (This image is taken from [3]) .................. 8

4. Maps created from a dynamic, populated environment without people filtering. (Image taken from [6]) ............................. 12

5. Maps created from a dynamic, populated environment using people filtering. (Image taken from [6]) ............................. 12
1 Introduction

This paper reviews various literature concerning Co-operative Mapping and Localization of Autonomous Robots in order to garner the knowledge necessary to implement and test a working framework that provides a team of robots with the ability to accurately map an environment by working together. The paper begins with a discussion of Simultaneous Localization and Mapping (SLAM) and how the framework for a single robot to position itself and, at the same time, map its environment can be extended to allow a team of robots to do the same. During the course of this discussion, the various steps required during the process are expanded upon, along with the available algorithms and their respective advantages and disadvantages.

2 Simultaneous Localization and Mapping

SLAM is a framework of algorithms that provides a robot with the ability to build a map of an environment for which no prior information exists and, at the same time, determine its location within that environment [2]. Using a map with other means of recording movement, such as odometry and environmental sensing devices, gives the robot the ability to detect and recover from incorrect sensor data and more accurately plot its position within an environment [3]. SLAM has produced significant results in the past decade, but mainly for single robots [10]. The problem of single robot localization has been solved in many different ways by posing it in a probabilistic state estimation framework which combines the position and bearing (Pose) of the robot, relative to its starting position, with its environmental observations (Landmarks) to estimate the current robot’s position and bearing in absolute world co-ordinates [18].

3 Co-operative Mapping and Localization

A substantial amount of effort has been expended on solving the problem of localization of a single robot, but localization of a team of robots is still a relatively new field [18].

Co-operative Mapping and Localization introduces additional problem on top of SLAM such as each robot’s role, centralization, aggregation, and communication methods [3]. Regarding the robot’s role in Co-operative Mapping and Localization, each robot in a team can be an independent entity running its own instance of a localization algorithm or, as in the case of [18], there can be master and slave robots where the master robots perform SLAM and the slave robots act as extended eyes and ears. The slave robots feed sensor data back to the master robots to be used in the localization algorithm, but do not perform SLAM themselves. The master robots have better odometry and environment sensors and must have the ability to make inter-robot measurements as well as a means of communicating with the other robots in the team [18].
Centralization/Convergence concerns the question of how the robots will converge upon each other once they have completed mapping their respective sections of the environment.

Aggregation is concerned with how to combine the different sections of the map, which have been gathered by each of the robots, so that each section is in its correct place.

The communication methods used will depend on what roles are performed by each of the robots in a team. Whether all the robots communicate with only a single, controlling robot (The Master robot as mentioned in [18]) or whether each robot is independent and communicates with every robot in its team to build its portion of the map if and when it observes a robot.

Co-operative Localization and Mapping allows larger areas to be mapped in parallel, which makes exploration more efficient since robots do not have to map areas that have been mapped already by other robots [17]. It also gives robots the ability to map environments that have limited prior information available efficiently and reliably [18].

4 Hardware

The robot used for the development and testing of the framework is a Fischertechnik robotic kit with a Fischertechnik ROBO TX Controller. The controller consists of a 32-bit ARM 9 processor operating at 200 MHz, a display, and an integrated Bluetooth radio interface operating at 2.4 GHz with a range of 10 m [4]. The robot has two ultrasonic sensors used to make observations of the environment and this sensor data forms part of the Measurement Model and is used to extract objects (Landmarks) from the environment [9]. The Plant Model contains measurements of how the robot’s position changes over time relative to its starting position by using encoders to measure the rotations of the motors [9]. Such relative position data is known as odometry data. The robot is equipped with two such encoders that provide it with the necessary odometry data that it needs to localize itself (Determine its position).

4.1 Encoders

Encoders are specialized motors that are used to capture odometry data by recording the rotations performed by a motor. The robot’s controller operates the motor via the controller’s internal PWM (Pulse Width Modulation) circuitry to control its forward or backward speed. Relying on the amount of time that the motor’s output shaft turns is unreliable since the speed that the motor turns is dependent on the battery’s voltage levels as well as the resistance that the motor encounters through surface friction on attached wheels [3]. A decrease in battery voltage or increase in resistance on the motor will cause the motor to spin slower and thus overestimate the distance traversed by the robot.

There are many methods that encoders use to record the rotation of the motor’s output shaft. Mechanical encoders use magnets or metal brushes around
the shaft in regular positions that generate a tick each time they pass a stationary contact in the motor casing. The number of ticks per second are recorded and represent the amount of rotation in terms of angular ticks [12].

Another method mentioned in [12] is the use of Optical encoders. These encoders have a patterned/punctured disc attached to the output shaft of the motor and a stationary contact in the motor encasing will record the number of ticks.

4.2 Sensors

Observations of the environment are captured by sensors which provide range and bearing data for observed Landmarks (Objects in the environment). These landmark observations are run through a landmark association algorithm and then combined with the odometry data, provided by the encoders, so that a localization algorithm can localize the robot in its environment as well as generate new map readings. There are many types of sensors that can be used in the measurement model, two of which are laser sensors and sonar scanners. Laser sensors are widely used these days as they provide accurate measurements at the speed of light and their output is easily processed [3]. The main problem with laser sensors mentioned in [14] is their price. Sonar scanners use a range measurement technique known as Time of Flight (TOF). The time taken between issuing the sound wave and recording the sound wave is recorded and this measurement is used to provide a range estimation, bearing in mind that sonar
scanners work at the speed of sound.

5 Odometry Data

Odometry data is captured by Dead reckoning sensors. These sensors provide data about the robot’s current position relative to its starting position. There are many types of sensors that provide dead reckoning data. For ground based robots Encoders are used, and for aerial vehicles GPS (Global Positioning System) is used, whereas for AUVs (Autonomous Underwater Vehicles) Gyros, INS (Inertial Navigation Systems, and DVL (Doppler Velocity Logs) are used [8].

6 Landmark Extraction

Landmarks are observed objects in the environment whose co-ordinates have been recorded by environmental sensing devices. A localization algorithm will use this data to match landmarks against the map if they exist or add the landmarks to the map if they have not been previously observed [14].

6.1 Unique versus Non-unique Landmark Observations

Normally, as described in [14], landmarks should be unique so that that do not get miss-associated with other landmarks in the map and they should be
adequately spaced to prevent them being interpreted as a single landmark. However, as described in [13], it is possible to localize with non-unique landmark observations by using an Importance Sampling based approximate solution and implicit hypothesis pruning. This works by removing as much ambiguities, caused through non-unique landmarks, as possible before the landmark data is used by the localization algorithm. A simpler method of removing landmark ambiguities mentioned in [13], is by using a compass to provide directional context to non-unique landmarks.

6.2 Time between Landmark Observations

Another requirement for successful SLAM, is that there must not be a long time period between observations of landmarks since this would cause the algorithm to reach high levels of uncertainty and get lost [14].

6.3 Static versus Dynamic Environments

Reference [14] mentions that another requirement for successful SLAM as that landmarks should remain stationary since a moving landmark would cause uncertainty in the localization algorithm, however, the authors in [6] explain how they implemented a probabilistic method of tracking multiple people in a populated environment and how they incorporated the the results of the tracking into the mapping process. They explain that the resulting maps are more accurate since the number of random or non-landmark objects in the environment are reduced and corrupted readings are flagged.
7 Data Association

Data Association is the ability of a robot to match a set of measurements, made by itself or other robots, against other robots and known features already in the map in order to determine new landmarks [18].

7.1 Multi-Hypothesis Tracking with Pruning

The authors in [13] use a method called Multi-Hypothesis Tracking (MHT), where the possible data associations form hypotheses which are tracked with a Kalman Filter after first being pruned to reduce complexity. This allows non-unique landmarks to be used in the environment since it reduces the ambiguities caused through non-unique landmarks, such as the poles of a goal post, before the landmark data is used in the localization algorithm.

7.2 Joint Compatibility Branch and Bound test

Reference [11] proposes the use of a data association method called the Joint Compatibility Branch and Bound test (JCBB). JCBB is more robust to quick, incorrect pairings even if the pose uncertainty is high and JCBB takes the compatibility of the observations made by more than one robot into account and it does this by performing a depth first search for the correct hypothesis [11].

8 Localization Algorithms

While a robot is moving, the odometry and landmark data that is collected is used by the localization algorithm, which maintains the uncertainties of landmark and position estimates to update the robots position in absolute world co-ordinates and map the locations of those landmarks if they have not already been seen [14]. The odometry data and the landmark data must be collected simultaneously or time stamped in order for it to be useful as input to the localization algorithm [14]. Localization algorithms combine odometry data with environment observations (Landmarks) to determine the robot’s position and orientation in absolute world coordinates (The robot’s pose). Localization algorithms can be divided into those that do not use Landmarks commonly observed by other team robots to improve their localization calculations and those that take advantage of environmental observations to improve each robot’s pose estimates, while simultaneously mapping the environment [10]. The localization algorithm can be distributed over all the robots in a team as demonstrated in [15] or each individual robot can run their own instance of the localization algorithm.

Reference [7] uses a stereo-vision based method that uses landmarks extracted by one robot to help other robots localize themselves. Reference [10] claims that the stereo-vision based method is focused on a specific application (stereo-vision-based) and is therefore unsuitable as a general localization method.
that makes use of landmarks in its pose estimation, but they retain the idea of using landmarks extracted by one robot to help another robot to localize itself. Their method is described in further detail in the Monte Carlo Localization section below.

8.1 Kalman Filter (EKF)

Reference [15] presents a method which use a single Kalman Filter distributed among the robots in the group. This approach does not use commonly observed landmarks to improve the accuracy of its localization calculations and addresses multi-robot localization as the relative localization of a robot based on the inter-robot distance measurements [10].

Similarly, the authors in [18] investigate a method of Co-operative Mapping and Localization, known as Moving Baseline Navigation, in which only one robot is responsible for maintaining estimates of the map and poses for each robot. Inter-robot measurements are combined with observations of the environment made by each robot in the team to determine each robot’s pose [18].

Reference [16] implements cooperative localization within a Kalman Filter Framework using the relative positions of the robots as the observations of the filtering part of the algorithm, and the state includes the positions of all the robots. This algorithm does not use commonly observed landmarks to improve the accuracy of its localization calculations [10].

8.2 Monte Carlo Localization

Reference [5] uses the Monte Carlo Localization algorithm to represent the probability distributions of the robots in a group and uses inter-robot observations to improve state estimation, but claims that the Monte Carlo Localization algorithm ignores the inter-robot correlation which result in the algorithm suffering from overconfidence.

A modification of the Monte Carlo Localization (MCL) algorithm is demonstrated in [10] where particles are not spread uniformly in the state space when a robot is lost, but rather according to information on the location of a commonly observed object whose distance and bearing is measured by the lost robot and whose absolute world coordinates are provided by other robots via direct wireless communication. This algorithm does use commonly observed landmarks to improve the accuracy of its localization calculations and takes advantage of the features exposed by particle filter algorithms and provides global robot and object localization [10]. The modification is said to decrease the time taken for a lost robot to recover and, due to the extra landmark information provided by other robots in the team, is robust to perceptual aliases, which occurs when environments have symmetries [10]. Each robot runs the Monte Carlo Localization algorithm to localize itself and is able to determine when the uncertainty about its localization drops below a given threshold and that uncertainty is determined through the existence of an observation model that enables the level
of confidence on the position of the object to be determined [10]. This modification to the MCL algorithm relies on a common object being observed by many robots in the team, which is, in this case, a soccer ball that all robots of the team can observe and provide position estimates to be used to localize the lost robot.

8.3 Markov Localization

Reference [5] uses an extended version of the general Markov Localization algorithm where the relative distance and bearing of two robots is inserted as an extra step in the belief update algorithm, which is based on the Bayes filter. The Monte Carlo Localization sampled version of the Markov Localization algorithm was used to influence the weights of the observed robot’s particles over the observing robot’s particle sampling of the inter-robot distance and bearing measurements.

9 Map Merging

Each robot builds its own map of the section of the environment that it is exploring and, when a robot observes another robot, attempts to combine those individual sections of the environment into a single complete map which works in absolute world co-ordinates. The authors in [1] present an algorithm for merging the occupancy grid maps produced by the multiple robots exploring the environment. Their algorithm produces a set of weighted possible transformations (Rotations and Translations) needed to merge two maps by extracting spectral information from the maps. Since each possible transformation is weighted, the algorithm is able to continue when ambiguities arise. The authors make the claim that their approach is deterministic, non-iterative, and fast (merging 250 thousand cells in 500 ms on common desktop PCs). The authors in [6] discuss an algorithm to build maps in dynamic, populated environments using a probabilistic method to track people and to incorporate the tracking technique into the mapping process. Their algorithm could be useful as a first step in co-operating mapping and localization since it can be modified to track other robots in the environment and thus filter them out of the mapping process and use their positions as reference points for merging their map sections.
Figure 4: Maps created from a dynamic, populated environment without people filtering. (Image taken from [6])

Figure 5: Maps created from a dynamic, populated environment using people filtering. (Image taken from [6])
References


