Simultaneous Localization and Mapping in Maze Solving

Divya Bhaskaran, Charushree MB, Abhijith M , Dr Jitendranath Mungara

RV College of Engineering, divyabhaskaran93@gmail.com,+919886219466

Abstract— Simultaneous Localization and Mapping is the process by which a robot is able to place itself in, and map out its environment. This twofold solution has proved to be a breakthrough in the quest to attain an autonomous robot. However the solution pertains to only a small set of spaces. To improve upon this, efforts are being made to cover not just localization, but also global localization and the kidnapped robot problem. The Kalman Filter algorithm and the Markov Localization algorithms are being replaced by the Monte Carlo Algorithm which represents a belief as a set of particles instead of approximating posteriors in parametric forms. The Rao-Blackwellised particle filter further reduces the particle space .Scan mapping and improving the odometry can further improve this filter method.

.**Keywords**—. Automated System, Simultaneous Localization and Mapping(SLAM), Maze Solving, Posteriors,Particle, Rao – Blackwellized Particle Filter (RBPF) ,Odometry.

INTRODUCTION

The rapid growth in the technology in our times is greater than ever. More and more autonomous robotic devices are infiltrated in the lives of people making theirs easier. The research of building smart robotic vehicles is a largely funded initiative. In many cases, autonomous robots can be the best option for specific missions. Conservative ways of rescuing survivors are time consuming and harmful for the survivors. Instead, we can have unmanned autonomous robotic vehicles which carry out rescue operations. Being equipped with the necessary sensory devices, unmanned autonomous robotic vehicles can scan the environment sending precious information to the rescue teams about the location of survivors. Additionally, there are also places where use of robots is the only way to achieve a work. Space exploration, nuclear plants, chemical factories, or any environment unreachable for humans, could be explored by an autonomous robot. So, independent mapping and localization for a robot became one of the main goals in robotics technology. This is a complex problem, and is not totally solved today. The main difficulties are the accuracy of measurements and the real-time processing in tandem with the minimum processing power available in most of the embedded systems. The entire above problem can be subdivided into three sub problems: Localization, Mapping and Path Planning.

Simultaneous localization and mapping, or SLAM[1], is a process very central to optimize and automate the system. The name refers to the twofold solution whereby a robot is able to map out its environment and also place itself in that environment. In other words, to learn a map of the robot's environment using the robot's sensors. For autonomous mobile robots, learning maps is often essential. Being able to automatically navigate in an environment is dependent on having a map, and manually creating this map is often a hard and labor intensive effort. Maintenance can prove costly enough to render the robot unusable. Equipping the robot with sensors and software enabling it to solve this task by itself can be of great importance to the success of the robot system. Autonomous mobile robots also need to be able to localize themselves in their environment. Some sensor arrays could provide a full state estimate, such as an overhead camera combined with computer vision software. This solution is used primarily when the environment restricted to a small surface, such as in the Micro Robot World Cup Soccer Tournament (MiroSot) [2]. In such applications, the full robot position can be computed directly. However, this does not account for changes in the size and nature of the environment. For a robot exploring unprepared indoor environments, its location most often has to be computed from several sensor scans, and is dependent on a map. Importantly, the problem of SLAM consists of two mutually dependent sub problems. If a complete and accurate map existed, simpler algorithms could have been utilized for generating position estimates. Likewise, if a complete history of accurate positions existed for the robot, map learning would be reduced to writing sensor data to a map representation. It is hard to estimate which process comes first. For this reason, the problem is recognized to be hard, and it requires a search for a solution in a high-dimensional space of possible locations.

LITERATURE SURVEY

The maze solving problem is one of the most popular trends in the field of robotics. In fact IEEE designed a “Micromouse” competition dating back to the 70s, which has become such a craze,that it is now held at regional, national and international levels all over the world. It is an amalgamation of elements of electronics, control, mechanical and software engineering and provides the perfect platform to teach mechatronics to students across all the fields.

In the late 1940’s, Claude Shannon built a Maze Solving Mouse. His maze consisted of a 25 square checkerboard maze with aluminum walls which were removable. A motorized carriage was underneath the board of the maze which provided power to the mouse through magnetic coupling. Relay circuits provided logic and memory to navigate the maze and remember the solution. The downside to this method was that the path determined was not the optimum path and the mouse performed very badly in a maze with no walls at all.

The findings of Shannon instigated Moore[3] to come up with a solution to optimize the path through a maze. He reconstituted the problem as an abstract one as a number of points that could be visited and available passageways between them.

Moore’s algorithm operated as follows. From the starting position, all points that could be reached in one step were marked by “1”.From there, all the yet unmarked places that could be accessed in one additional step were marked by “2”. From there the yet unmarked places were marked by “3” which could be reached in yet another additional step. This iterative process continued until the final point was reached. This method however, worked in abstract mazes with fewer positions, connected by an irregular array of passageways.

*Ivan E Sutherland’s-“**A Method for Solving Arbitrary-Wall Mazes by Computer”** IEEE member (1969)* talks about solving mazes with extended open areas where arbitrary walls are placed. It reduces large open areas containing many parts to a small set of shortest paths. Then Moore’s algorithm is used. However, this paper covers only a computer simulation of a vehicle and not an actual real world thing.

Jianping Cai, Xuting Wan, Meimei Huo and Jiazhong Wu’s-“***An Algorithm of Micromouse Maze Solving”** Zhejang University City College, Hangzhou, China(2010),*proposes a maze solving algorithm called “Partition central Algorithm”, which is used to find the shortest path in a micromouse competition maze. A standard 16*16 maze was divided into 12 partitions. Depending on the absolute direction of the micromouse and the partition location exploring rules alter when the micromouse walks to optimize the maze exploration process.

A generic 3-D simulation platform with multiple with multiple viewpoints was presented that could be used for education and research purposes by *Jianpang Cai, Meimei Huo, Jiazhong Wu, Bin Song’s-“**Micromouse Competition Training Method-Based on 3D Simulation Platform”** Zhejang University City College, Hangzhou, China(2010)* offer many characteristics that differentiate it from its 2D version. It offered a platform for students to participate in the IEEE Micromouse competition based on the 3D simulation platform. It offered a complete example suitable for large scale student participation. This proved the popularity of the competition.

Apart from the micromouse competition, various methods were developed for solving mazes using different techniques. One such one was by *Adil M.J Sadik, Maruf A Dhali, Hasib M.A.B Farid,Tafhim U Rashid,A Syeed’s-“**A Comprehensive and Comparitive Study of Maze Solving Techniques by Implementing Graphy Theory”**(2010),* state how without the use of Artificial Intelligence,the maze solving algorithms are most ineffective. Graphical and non graphical solutions were developed to increase the efficiency. In that realm however, graph theories triumph non graph theories. By proper interpretation and mathematical modeling, it is possible to find the shortest distance between any two nodes. Some algorithms employed were *Depth First Search, Breadth first search, Flood Fill or Bellman Ford algorithm*.

The solution to SLAM-Simultaneous Localization and Mapping has been one of the notable successes of the robotics community. *Hugh Durrant-Whyte, Fellow and Tim Bailey’s-“**Simultaneous Localization and Mapping (SLAM) Part I The Essential Algorithms”*** and *“**Simultaneous Localization and Mapping Part 2”*** covers the probabilistic form of the SLAM problem, essential solution methods and significant implementation. Part II discusses the recent advances and new formulations of the SLAM problem for large scale environments.

Problems with current methods are:

1. They are not always designed for a dynamic and constantly changing environment.
2. The constant problem of whether mapping should come before localizing or the other way around. Likewise, if a complete history of accurate positions existed for the robot, map learning would be reduced to writing sensor data to a map representation [4]. It’s a chicken and egg problem [5].

METHODOLOGY

For the proper implementation of SLAM in maze solving, the following are crucial:-

III.I Map Representations

Several map representations are recognized for SLAM purposes, and most of them can be put in one of two categories: landmark-based maps and occupancy grid maps [5].

III.I.I Landmark-based maps

Landmark-based maps are based on landmarks, which are features in the environment. Landmarks can be corners, line segments or points. The landmarks are seen as distributed in a continuous space. In other words, each landmark is associated with a position in space. The landmarks are assumed to be relatively sparse and unambiguous [6].

III.I.II Occupancy Grid maps

Occupancy grid maps can be seen as regarding everything as landmarks. The individual sensor measurements are assumed to be individually not very distinctive, but dense [6]. Grid maps discretize the environment into a grid – for regular maps a grid of two dimensions. They can have a variety of different ways of representing a cell, everything from a simple binary bit to tree structures [6].

III.II Sensors

The choice of sensors for performing SLAM is large, and different types of sensors are used in different contexts. Autonomous underwater vehicles can use sonar, while unmanned aerial vehicles can use radar systems, infrared cameras or other means of sensing. The focus of this paper is on autonomous indoor wheeled robots

The system consists of:

III.II.I Range Sensors:

While it is not strictly required, most autonomous mobile robots have some form of range sensor. A range sensor can tell the distance to the nearest object in a given direction or sector. Ultrasonic, IR and laser based systems (LIDAR) are common, while vision systems based on digital cameras represent another alternative.

The range sensors form the method of detecting obstacles, facilitating the generation of a map and location tracking of the robot relative to them, whether the map be landmark or grid map based. The range sensors also play a role in correction of odometric errors where odometric sensors are present.

III.II.II Odometric sensors:

In order to efficiently and correctly cope with more situations, many robot designs and SLAM approaches also include odometric sensors, that is, sensors which measure the distance traveled. A naive SLAM algorithm could be solely based on odometry and dead reckoning where range sensor readings are written to the map based on the position deduced by odometric measurements. Such mapping with raw odometry most often gives inconsistent results, since errors accumulate and are never corrected. For these reasons, LIDAR often is preferred.

While the individual distance measurements of a LIDAR might have higher uncertainty, there are usually so many of them, allowing for a more precise estimate to be deduced through scan matching, comparing several scans and inferring their relative positions. For instance, [7] reports that the probability distribution $p(z|x)$ is much more peaked than $p(x|x', u)$, meaning the probability of obtaining the reading z given a position x is less uncertain than the probability of the position x given the previous position x' and an odometric reading u.

III.III Scan Matching

Scan matching is a concept frequently used in SLAM algorithms. For some algorithms, scan matching is the most central aspect. Combining range sensor measurements form one range sensor revolution, hereafter called scans, can be used to estimate the movement of the robot between these scans. Measurements originating from other range sensors are represented in a coordinate system fixed to the sensor unit. When the robot and sensor move, the coordinate system in which the range scans are given is moved relative to world and map coordinates. Scan matching is the procedure of aligning different range scans to a world or map coordinate system, based solely on the scans themselves, or with the help of other inputs. In this view, scan matching aligns range scans to each other, or to a previously obtained or static map [8].

III.IV Particle Filters

The most effective solution to the mobile robot localization problem is *position tracking* .In this field an effective algorithm would be the Kalman filter algorithm where the posterior distribution of the robot poses conditioned on sensor data. However in the *global localization problem* and the *kidnapped robot problem*, this algorithm is ineffective. Instead, a better algorithm would be that of *Monte Carlo localization algorithm* (MCL). It can accommodate arbitrary noise distributions (and nonlinearities in robot motion and perception). Thus, MCL avoids a need to extract features from the sensor data.

The key idea of MCL is to represent the belief by a set of samples (also called: particles), drawn according to the posterior distribution over robot poses. In other words, rather than approximating posteriors in parametric form, as is the case for Kalman filter and Markov localization algorithms, MCL represents the posteriors by a random collection of weighted particles which approximates the desired distribution .

International Journal of Engineering Research and General Science Volume 3, Issue 2, Part 2, March-April, 2015
ISSN 2091-2730

DESIGN AND IMPLEMENTATION

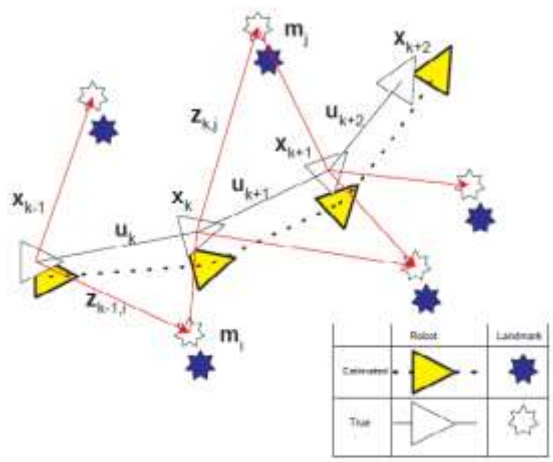

Fig 4.1: The essential SLAM problem.

Consider a mobile robot moving through an environment taking relative observations of a number of unknown landmarks using a sensor located on the robot as shown in Figure 2. At a time 'k' we have:

X_k=State vector describing the location and orientation of the vehicle.

U_k=Control vector used to drive the vehicle to state x_k at time k. It is applied at time k-1.

M_i=A vector describing the location of the i^{th} landmark whose true location is time invariant.

Z_{ik}=Observation taken by the vehicle of the i^{th} landmark at the time 'k'. If multiple observations of the landmark are taken or if the landmark is irrelevant to the problem, it is z_k.

In addition we have:

$X_{0:k}=\{x_0,x_1,x_2......x_k\}=\{x_{0:k-1},x_k\}$ which is the history of locations.

$U_{0:k}=\{u_0,u_1,u_2......u_k\}=\{u_{0:k-1},u_k\}$ which is the history of control inputs.

$m_{0:k}=\{m_0,m_1,m_2......m_k\}=\{m_{0:k-1},m_k\}$ which is all the landmark observations.

IV.I. Rao –Blackwellized Particle Filter (RBPF) based SLAM.

The high dimensional state space of the SLAM problem makes the application of particle filters computationally infeasible. However it is possible to reduce the sample space by using RBPF [9]. SLAM posterior is as follows:

$\mathbf{P(x_{1:t},m|z_{1:t},u_{0:t-1})=P(x_{1:t}|z_{1:t},u_{0:t-1})*P(m|x_{1:t},z_{1:t})}$

Where,

$\mathbf{P(x_{1:t},m|z_{1:t},u_{0:t-1})}$ is the SLAM posterior,

$\mathbf{P(x_{1:t}|z_{1:t},u_{0:t-1})}$ is the robot path posterior. $X_{1:t}$ is localization which is carried out using MCL.

$\mathbf{P(m|x_{1:t},z_{1:t})}$ is the mapping with known poses.'M' is the pose estimate from MCL.

The particle filter is used to represent the potential trajectories of the robot. Each particle carries its own map and we now have joint posteriors of the poses of the robot and the map.

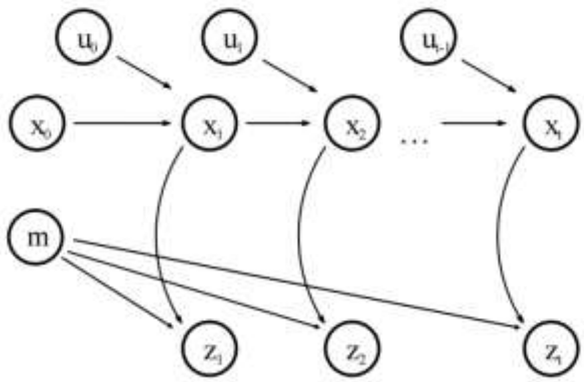

Fig 4.2 A graphical model of the Rao-Blackwellized Mapping.

IV.II. Pose Correction using Scan Mapping

In the case of Rao Blackwellized mapping in grid maps, the maps are quite big and since every particle has its own map, the number of particles needs to be contained. To improve this, we improve on the pose estimate before particle filtering.

We maximize the position of the ith pose relative to the (i-1)th

$\mathbf{X_t^{*}=arg\ _{xt}max\ P(z_t|x_t,m_{t-1})*P(x_t|x^{*}_{t-1},u_{t-1})}$

Where:

$\mathbf{z_t}$ is the current measurement,

$\mathbf{(x_t,m_{t-1})}$ is the map constructed so far

$\mathbf{U_{t-1}}$ is the robot motion.

IV.III. RBPF SLAM with Improved Odometry

Scan matching provides a locally consistent pose correction. Pre correct short odometric sequences using scan matching and then using them as input to Rao Blackwellized Particle Filter. The advantage is that fewer particles are needed since the error in the input is reduced.

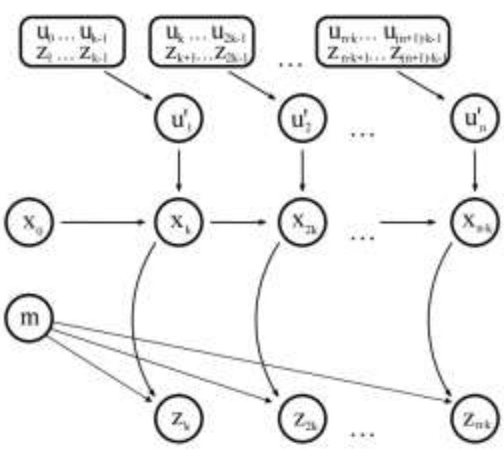

Fig 4.3 Graphical model for mapping with improved Odometry.

This method uses Odometry as the input as opposed to scan matching in the standard RBPF. The number of partitions also varies from 500 to 2.000.

EXPERIMENTAL RESULTS

The following are the outputs produced from implementing the methods provided:

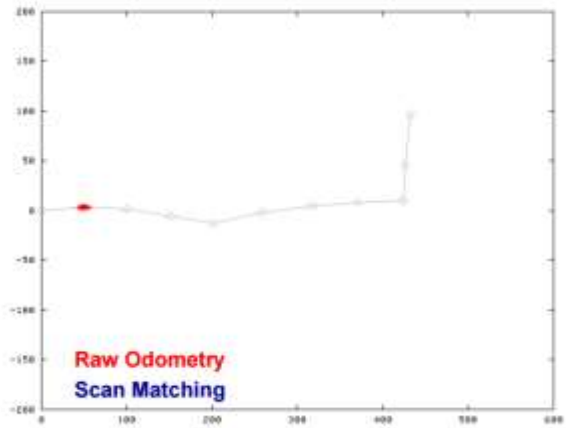

Fig 5.1 Motion Model for Scan Matching

Fig 5.2 RBPF Slam with Improved Odometry

CONCLUSION

In the field of motion, Simultaneous Localization and Mapping was a breakthrough in terms of reaching the goal of autonomy.

The SLAM method provides a solution to the key problem of mapping and localization for any autonomous robot. The past decade, in particular, has seen substantial progress in our understanding of the SLAM problem and in the development of efficient and robust algorithms. However in large-scale mapping, global localization, kidnapped robot problem, problems involving many vehicles and in mixed environments with sensor networks and dynamic landmarks, SLAM still has a long way to go. The delayed data-fusion concept complements batch association and iterative smoothing to improve estimation quality and robustness. Appearance-and pose-based SLAM methods offer a radically new paradigm for mapping and location estimation without the need for strong geometric landmark descriptions. These methods are opening doors to new ideas and making links back to fundamental principles in robotics.

FUTURE WORKS

The key challenges for SLAM are in larger and more persuasive implementations and demonstrations. Some of the problem include reducing the number of particles, covering not just indoor localization but also global localization and kidnapped robot problem, which is perhaps most applicable in war fields.

Even though the progress has been substantial, the scale and structure of many environments are limited. The challenge now is to demonstrate SLAM solutions to large problems and scope where robotics can truly contribute: driving hundreds of kilometers under a forest canopy or mapping a whole city without recourse to global positioning system (GPS), to create smart cars and to demonstrate true autonomous localization and mapping of structures such as the Barrier Reef or the surface of Mars. SLAM will contribute in making all this possible.

REFERENCES:

[1] Hugh Durrant-Whyte, Fellow, IEEE and Tim Bailey "Simultaneous Localization and Mapping Part 1 The Essential Algorithms".

[2] Prahlad Vadakkepat, Tong Heng Lee, and Liu Xin. Application of evolutionary artificial potential field in robot soccer system. In IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, pages 2781–2785. IEEE, 2001

[3] E.Moore, "Shortest path through a maze", Annals of the Computation Laboratory of Harvard University, Vol 30,Cambridge Mass:Harvard University Press,1959,pp,285-292

[4] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. In Proceedings of the National conference on Artificial Intelligence, pages 593–598. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002

[5] S. Thrun, W. Burgard, D. Fox, et al. Probabilistic robotics, volume 1. MIT press Cambridge, MA, 2005.

[6] Austin I Eliazar and Ronald Parr. Dp-slam 2.0. In Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, volume 2, pages 1314–1320. IEEE, 2004

[7] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. Robotics, IEEE Transactions on, 23(1):34–46, 2007..

[8] Stefan Kohlbrecher, Oskar von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable slam system with full 3d motion estimation. In Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on, pages 155–160. IEEE, 2011