CIMSA 2008 – IEEE International Conference on
Computational Intelligence for Measurement Systems And Applications
Istanbul – Turkey , 14-16 July 2008

An Occupancy Grid Based SLAM Method

Ozan Özışık, Sırma Yavuz
Yıldız Technical University
Computer Engineering Department

Abstract – *Simultaneous localization and mapping (SLAM) is an active area of research. SLAM algorithms should allow the robot to start its movement from a random position in an unknown environment and to build the map of the area while knowing its own position relative to the map. Thus, at the end of the mapping task robot should be able to return where it has started. Especially in real time applications, using limited sensor data, there are still many problems to be conquered. In this study a probabilistic occupancy grid approach is proposed to build the map of an unknown environment. The method tested both in a simulation environment and on a real robot. Although there are some improvements to be made, the initial results are promising.*

Keywords – *SLAM, occupancy grid mapping, probabilistic robotics*

I. INTRODUCTION

Robotic Mapping has been a highly active research area for the last two decades [1]. Mapping is the process of the robot to model the environment it exists using the control signals and the sensor data it gets. There is an important problem in application; sensor data is usually noisy and also the robot may not be moving precisely as the control signals tell it to do. In order to solve this problem, probabilistic methods have been developed. Smith, Self and Cheeseman developed a powerful statistical framework, for simultaneously solving the mapping problem and the problem of localizing the robot relative to its growing map, in 1990’s [2,3]. Subsequent to this progress, robotic mapping has commonly been referred to as simultaneous localization and mapping (SLAM).

There are many SLAM algorithms, and Kalman Filter Method is probably the most commonly used. Kalman Filters are Bayesian filters that represent posterior pose estimation $p(s_t,m \mid z^t, u^t)$ with Gaussian distribution. There are some drawbacks associated with Kalman Filter; it needs artificial landmarks, it can not solve the data association problem, sensor data has to be preprocessed, and noise in sensor data must have a Gaussian distribution. Another popular algorithm is Expectation Maximization which is developed using the maximum likelihood model. As it doesn’t use an incremental model while processing the data, the same data is processed again and again; the method doesn’t only depend on the last step but it depends on all the previous steps. Expectation maximization algorithm can solve the data association problem but it can not work in real time. Incremental Maximum Likelihood (IML), is a method that creates the map step by step while it gets the sensor data. As it doesn’t consider the uncertainties, it is unsuccessful in mapping cycling environments. But there are some studies [4, 5, and 6] that use IML while keeping track of the uncertainties, and they can correct the map backwards in time whenever an inconsistency is detected.

II. PROPOSED APPROACH

In this study, a method that incrementally builds an Occupancy grid map is proposed. The map is initialized with an *occupancy value* in all the grids. The initial *occupancy values* are selected as 0.5 and are updated in every step that sensor data is taken. Then the control signals are created depending on this up to date map. In the map, occupancy is represented by a rational number greater than 0 and in every step that an obstacle is observed in a point’s neighborhood, occupancy value of the point is increased. Occupancy update is done according to equation (1). In this equation, P is the point whose occupancy value will be updated, occ_p is the old occupancy value of P, occ_p' is the new occupancy value of P, α is the degree of increment, X is the point that an obstacle is observed, and R is a neighborhood function that is inversely proportional to the distance between X and P.

$$\mathrm{occ_p}' = \mathrm{occ_p} * (1 + \alpha * \mathrm{R}(\mathrm{X}, \mathrm{P})) \tag{1}$$

As it is seen in the equation, the method doesn’t only update the occupancy value of the point that an obstacle is observed, it also updates the occupancy values of all the points in a neighborhood. While the robot is moving, the points whose occupancy values are above a threshold are considered as obstacles and decisions are taken depending on these points.

A point’s occupancy value is updated in the following circumstances:

1) When an obstacle is observed in point P, P’s occupancy value is increased.

2) When an obstacle is observed in point P, occupancy values of the points in the neighborhood of P are updated in a degree that is inversely proportional to their distances.

3) When an obstacle is observed, occupancy values of the points between the obstacle and the robot are decreased.

4) When sensors don't observe any obstacle, occupancy values of the points that are on the direction of the sensors are decreased.

5) Occupancy values of the points that are on the path that robot is moving on are decreased.

III. STRUCTURE OF THE MOBILE ROBOT

The three wheeled mobile robot used in this study is constructed in Computer Engineering Department of Yıldız Technical University[1].

While the SLAM problem may be considered as a solved problem with sophisticated sensors, it is very challenging if the robot has limited sensing capabilities. Therefore one of the important properties of the robot used in this study is being equipped with only 6 infrared sensors for proximity detection. Each of these sensors takes a continuous distance reading and returns a corresponding analog voltage with a range of 10cm to 80cm.

An optical encoder is used to calculate the distance the robot has traveled, based on the diameter of the wheel. The information obtained from IR sensors and optical encoder is used to localize the robot relative to its growing map.

The robot is shown in Figure 1.

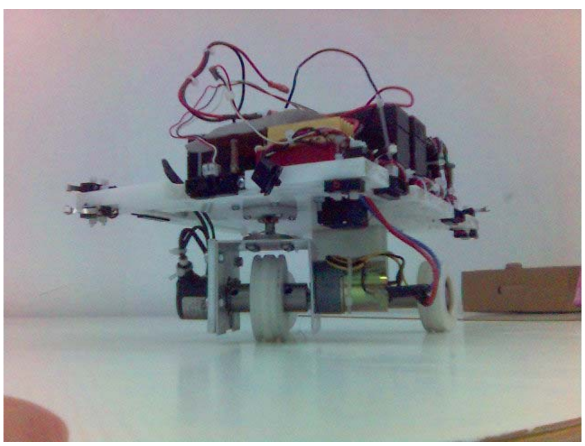

Figure 1 - The three wheeled mobile robot used in the study

IV. RESULTS

Initially the proposed algorithm is tested on a simulation environment to adjust the parameters. The simulation environment is completely imitates the real robot including the sensor readings and localization of the robot. The graphical user interface of the simulation environment used in our studies is shown in Figure 2. It is possible to place the robot at any point to start with and create obstacles on desired positions. The algorithm will control the simulation and it is possible to follow the trajectory of the robot during the run.

The final map determined by the algorithm is exported to an image file as seen in Figure 3.

[1] This research has been supported by Yıldız Technical University Scientific Research Projects Coordination Department. Project Number: 27-04-01-01.

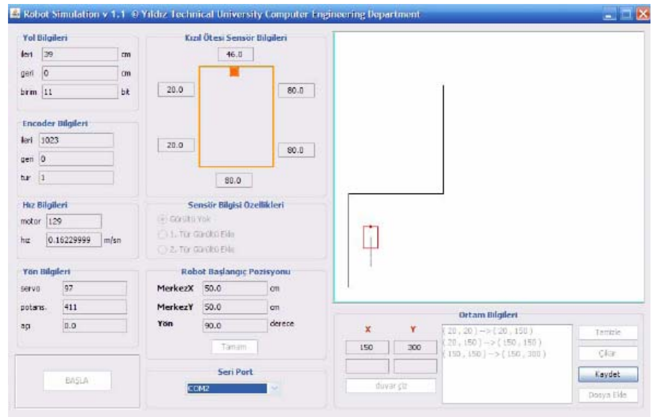

Figure 2 - The graphical user interface of the simulation environment

There is one important difference in simulation environment which is that the sensor readings are ideally correct or can only include a pre defined gauss or exponential noise. While using the simulation environment where the IR sensor readings are assumed to be correct the proposed algorithm is able to produce correct map of the environment.

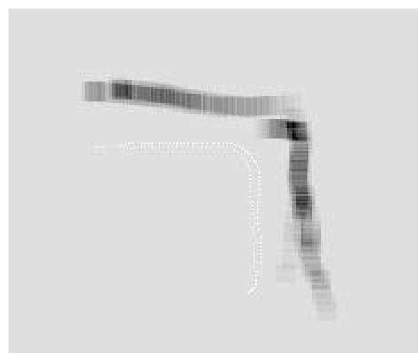

Figure 3 – Map for an L shaped corner obtained by using the simulation environment

In second part of the experiments the algorithm is tested on real robot, thus having noisy IR sensor readings. Some of the results obtained for a rectangular room and an L shaped corner are shown in Figure 4 and Figure 5.

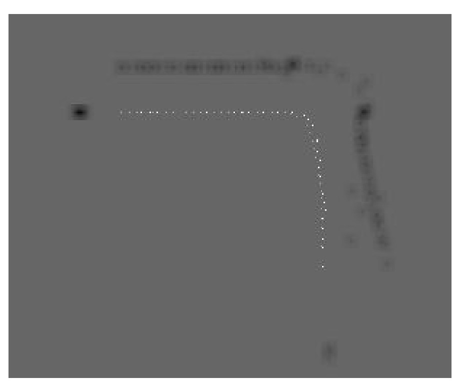

Figure 4 - Map for an L shaped corner obtained by using the real robot

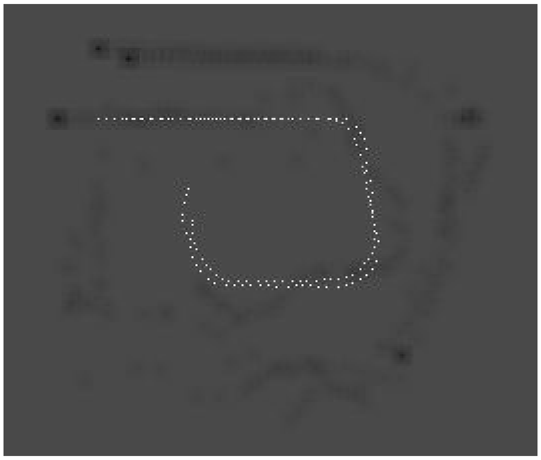

Figure 5 - Map for a rectangular area obtained by using the real robot

As it can be seen from the figures, light gray grids that are inconsistent with the trajectory of the robot are the result of noisy sensor readings. These noisy data may lead to wrong identification of obstacles (walls). As a solution currently it is planned to calibrate the sensors via software to remove the sensitivity differences between them. Upon experiments also the threshold values will be determined more accurately.

REFERENCES

[1] S. Thrun, (2002), Robotic mapping: A survey, Exploring Artificial Intelligence in the New
Millenium, Morgan Kaufmann.

[2] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfong, editors, Autonomous Robot Vehnicles, pages 167–193. Springer-Verlag, 1990.

[3] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. Technical Report TR 4760 & 7239, SRI, 1985.

[4] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), 2000.

[5] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. International Journal of Robotics Research, 20(5):335–363, 2001.

[6] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), San Francisco,
CA, 2000. IEEE.

[7] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, The MIT Press, 2005.