Coverage Efficiency in Autonomous Robots With Emphasis on Simultaneous Localization and Mapping Algorithms
TJHSST Senior Research
Computer Systems Lab 2009-2010

Mo Lu

January 26, 2010

Abstract

Coverage efficiency is a major goal of certain autonomous robotic systems. In the field of robotic lawnmowing, coverage efficiency has yet to be fully developed and there are different methods to approach coverage efficiency. The solution this paper covers is uses Simultaneous Localization and Mapping, known as SLAM. Using a laser scanner, SLAM algorithms create a map detailing the obstacles of the enviornment. Once obstacles are mapped, the algorithm process the map, and dictates where the robot can move, where it has moved, and where it currently is in relation to the obstacles. This data will enable the robot to cover the entire lawn.

Keywords: map processing, area efficiency

1 Introduction

Today, automated systems have supplemented humans in previously labor-intensive tasks. Automated lawnmowers are an example of these systems, but the currently avaliabe technology in automated lawnmowing is inefficient and primitive. This paper will propose and implement an alternate method to automated lawnmowing, known as Simultaneous Localization and Mapping, then report back the results.

2 Background

Commercial autonomous lawnmowers today do not have processing systems appropriate for efficient coverage. Current approaches to commercial robotic lawnmowing operate under the idea that if a lawmower is constantly mowing the lawn, then the lawn stays constantly mowed[1]. This is done by a series of random cuts and turns, which if given enough time, theroetically could cover an entire unmowed lawn[1]. Another aspect of this method is the use of "bump-and-go" technology. The system does not recognize the presence of obstacles until it actually hits it, and when it does hit obstacles, it does not store their locations for future use. This method is horrifically innefficient in terms of time and energy, when backtracking is taken into consideration. Random cuts also contain the possibility that a certain section of the lawn will never get mowed. This project proposes a different approach to this method: use of mapping tecniques to recognize landmarks, avoid obstacles, and naviagate an enviornment[4]. This method consists of three parts: 1) Use of a constantly updating laser scanner to recognize obstacles, 2) Creation of obstacle map using the laser data, and 3) Processing that obstacle map for runtime efficiency[2]. Success is determined by how effectively the robot avoids the obstacles and how quickly it runs through the lawn.

3 Development

3.1 Theory

SLAM theory is centered around the mapping process. A laser scanner is mounted on the robot, and pings out laser data in a 180 degree angle. The time it takes for the laser to hit an obstacle determines how far the obstacle is. These values are tracked by the system while the scanner is constantly working, and repeated obstacle values signify an obstacle, which the robot maps in relation to its current position. Once the obstacles are mapped, the robot will be able to process the most viable and efficient route through the lawn, taking into consideration the obstacles, terrain, and boundaries of the lawn. The end result will enable the robot to navigate and mow the lawn.

3.2 Project Work

Before the SLAM algorithims can be implemented into a physical robot, it must first run in a simulation. The current version of the simulation consists of a pre-created matrix based enviornment where the obstacles and terrain have been set. The robot is placed in the environment and keeps track of its position and obstacles, via the use of a coded coordinate system, a scanner mimic, and a blank obstacle map. As the robot moves and scans through the enviornment, obstacles are recognized, and the robot begins to build on its own independent matrix enviornment. The output of this mapping process matches the locations of the obstacles in the enviornment, and gives the robot an idea of where it can and cannot move in future mowings. Once the simulation best reflects a live environment, the program will be adapted for use with a laser rangefinder/laser scanner.

4 Testing and Analysis

The most general test of the performance of the system is if it mows the lawn. This depends on wether or not it maps the environment accurately. When efficiency is taken into account, three new categories for testing arise:

- Time efficiency
- Coverage precentage
- Backtracking

These testing categories are dependent on obstacle and boundary recognition, obstacle mapping, location tracking, and unmowable terrain recognition. Current focus is on the testing of obstacle/boundary recognition and obstacle mapping. Testing for the current focus is determined by how accurate the obstacle map is when compared to the environment. Later, the simulation will be adapted for testing in a random matrix enviornment, and then a non-matrix based enviornment. The non-matrix based enviornments will be able to function in a physical enviornment. Many conditions must be met for success of this project, if the original goal is to be met. Future testing will address the processing aspect of the program, with sucess determined by coverage and time efficiency. Current analysis of the project is determined by the correlation of the obstacle map with the enviornment.

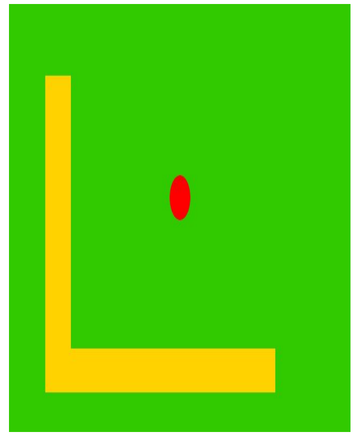

Figure 1: Environment

5 Results

The robot is correctly placed in the enviornment, and obstacles are generated. See Fig. 1. Red represents the lawnmower, yellow represents the boundaries.

Mapping algorithims print out a matrix-based map. See Fig. 2. [1] represents an unmoveable zone, and [0] represents moveable zones.

Current inputs include diagional, vertical, horizontal, and circular obstacles. See Fig. 3 and 4.

Outputs work fine for the most part, but errors occur with circular obstacles and diagional obstacles. Since the program scans using angles, the 45 degree scan assigns a block with an obstacle value, even though that does not exist. Diagionals always return one layer too thick. This is problematic when the diagional obstacle is touching the edge of the map. See Fig. 5.

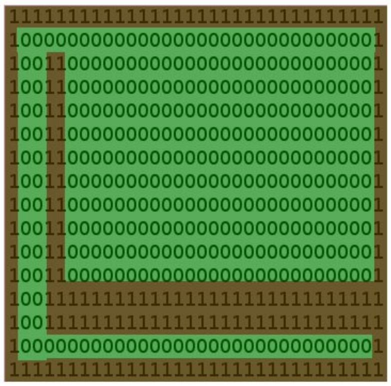

Figure 2: Modified Environment

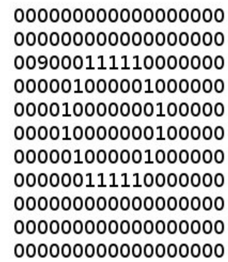

Figure 4: Input Environment: Circle

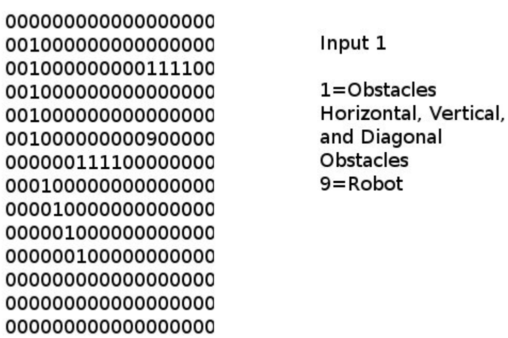

Figure 3: Input Environment: Diagional, Vertical, and Horizontal

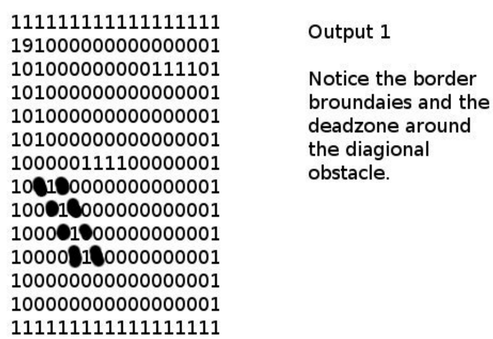

Figure 5: Output: Diagionals

6 Discussion

Before the SLAM algorithms can be implemented into a physical robot, it must first run in a simulation. The current version of the simulation consists of a pre-created matrix based environment where the obstacles and terrain have been set. The robot is placed in the environment and keeps track of its position and obstacles, via the use of a coded coordinate system and a scanner mimic. The robot moves and scans through the environment so long as obstacles are a certain distance away, and the environment map does not equal the obstacle map. Obstacles are recognized, and the robot begins to create its own independent matrix environment. Since the output of this mapping process matches the locations of the obstacles in the environment, it can be concluded that the scanning and obstacle recognition works for certain obstacles. That, along with the robot's ability to keep track of its position gives all the nessacary data to begin optimization algorithims. However, before that can happen, the simuilation must be tested for randomly-generated matrix environments and non-matrix based environments (graphic-based). Also, non-matrix based enviornments mean that the program cannot have a coordinate system, and process its location based off odemetry (wheel movement calculations) and its last known position. One problem that needs to be address in the current code is the tendency to re-scan already known obstacle locations. Future versions will need to reflect more realistic conditions such as terrain types and power-sources.

7 Conclusion

The current version of the program gives all the nessacary data for optmization processing to begin. However, non-matrix based environments must still be testested before this phase can begin, in order to fully mimic live runs, and current issues such as the diagonal and circular errors must be fixed before the program can be incorperated into a laser scanner.

References

[1] Husqvarna, "Husqvarna Automower", http://www.automower.us, 2009.

[2] Sren Riisgaard and Morten Rufus Blas, "SLAM for Dummies", pp. 1-44, 2003.

[3] Ian Schworer, "Navigation and Control of an Autonomous Vehicle", pp. 1-84, 2005.

[4] Dustin Bates and Evan Dill, "The Ohio University Autonomous Lawnmower", pp. 1-21, 2009.