Simultaneous Localization and Mapping: Military Applications - Student Experience

Suhair Amer, Deanna Tatum, Jesse Hankins, and Kyle Meystedt

Department of Computer Science,
Southeast Missouri State University, Cape Girardeau MO

Abstract - This project explores the concept of using a general tree to solve the problem of simultaneous localization and mapping as it pertains to autonomous robots traversing unknown, potentially hostile environments. This project was completed as part of a sophomore programming C++ class required by computer science students.

Keywords: simulation localization, trees, C++

1. Introduction

Autonomous robots can take on many real-world roles, from domestic assistants to search and rescue and military applications. This project will simulate a robot traversing and mapping a hostile structure, tracking hostile entities without engaging them. This type of application is useful in military situations when prior knowledge about an enemy structure would aid in planning tactical maneuvers. This robot must be able to discreetly traverse and store relevant information concerning potential obstacles and pathways within the structure.

This utility is not new; for any autonomous robot to perform a function, it must have the ability to create and maintain accurate models of their environments and relative positions in them. This problem, which has been prevalent in robotics for decades, is often referred to as SLAM, or Simultaneous Localization and Mapping (Rogers III, 2013) (Corcoran, Bertolotto, & Leonard, 2014). The solution to this problem is dependent upon several factors, including the types of sensors used by the robot to obtain information about its environment, the limitations of those sensors, the physical limitations of the robot if it must traverse and actively explore to acquire information, what sort of map is required, etc. (Thrun, 1998).

Several types of sensors have been used, often several in conjunction, to help solve these problems when exploring an environment. Odometry readings, i.e., readings from internal sensors measuring distance traveled or the robot’s angular rotation, by themselves are often quite unreliable. External sensors are required to compensate for this inadequacy (Patki, 2011). A robot will typically use ocular (vision) sensors in addition to range sensors like sonar sensors and laser scanners (Lee, Cho, & Song, 2012). Sonar or auditory sensors can allow for a robot to evaluate and localize events such as humans talking or doors slamming, but they are difficult to calibrate such sensors to dynamically adjust for the robot’s own motion, as well as ambient sounds (Sasaki, Kagami, & Mizoguchi, 2009). Laser scanners are often much more precise, but they are often cost prohibitive.

Another alternative is to simply use visual cues for all sensors and rely on a binary search pose estimation technique. This technique uses visual landmarks to iteratively converge towards the best estimation of the camera’s position relative to the points (Ross, Martchenko, & Devlin, 2013). This visual approach to the localization problem is both cost effective and subtle. Such approach would be required of a robot with real-world applications in the military, where discretion is of the utmost importance. One factor to consider, however, is the relatively enormous processing power required to store the images (Mendes, Coimbra, & Crisostomo, 2011). Once the sensors have detected information about the environment, the robot will need a way to process the objects discovered. These objects can be classified in a multi-level hypothesis hierarchy, much like the robotic butler HERB, which classifies objects as potential people or chairs (Srinvasa, et al., 2009). These objects representing potential humans and obstacles are stored in a hierarchical tree for easy recall.

Landmarks can also be stored in a tree, allowing the grouping of landmarks to determine relationships and pathways between objects. Using this method is effective for memory storage. The smaller details, which are stored further down the tree, can potentially be removed or reduced in the event that more memory is needed while still maintaining a broad view of the area. This will also us to see a global map

as a compilation of smaller, local maps of each room (Mair, et al., 2014).

These potential solutions, drawn from research involving several branches of robotics, have helped the students form a basis for their solution to the problem of modeling a previously unknown area and storing the information in the most effective ways possible.

2. Analysis

The project simulates a fully autonomous robot design for military use in reconnaissance by traversing hostile structures to generate a map and provide enemy numbers. It primarily consists of three systems: visual scanner (simulated input), coordinate mapping and overarching driver (simulated transversal). After being deployed to the structure, it will begin by attempting to scan through any crevasse of opportunity to evaluate its intended destination for hostile entities. When able, it will enter the new room and repeat this process as it discovers pathways.

The scanning process will record all doorways, windows, objects and entities within the space. These have their locations recorded and are assigned unique IDs as needed, then the information is transcribed through the coordinate system to be stored in the robot’s database in the form of a multi-linked Binary Tree. Since it is designed to avoid engaging an enemy, if discovered it will attempt to retreat through a previously recorded pathway until it is able to resume the mission or return to the operator. If it becomes compromised, there is a fail-safe system in place to erase the software and trigger a self-destruct mechanism.

Upon returning from a successful traversal, the robot's operator is provided an interface to review the information. It can be graphically presented as maps of each room in addition to a text mission log. Each option will also provide the number of living entities encountered and if they appeared to be armed. The maps presented will be comprised of characters with a reference key provided and can be output to a text file.

3. Design

When designing the data structures used, the original idea was to use the maps library to map each room to a search key, but it was quickly discovered that mapping in this manner was going to be too limiting. Only being able to use one search key was not conducive to the connected nature of mapping a building. The obvious solution was to use a tree, which allowed for multiple connections between rooms and easier traversal. It was important to be able to traverse in both directions, so each node had to have a parent pointer in addition to children, i.e., the nodes had to be double linked to ensure that the robot could freely move throughout the building as necessary.

Another important design consideration was the realistic simulation of an actual robot. Without access to a mobile robot with physical sensors, the program had to simulate motion through a state machine decision making concept that controlled a pointer traversing the map. Additionally, in order to simulate the visual sensory information, it was decided that the best option was to use text input files containing the information that would be interpreted by the visual sensors of the robot.

4. Implementation

This simulated robot was coded entirely in C++. The robot itself is represented as a state machine, and the building is stored in a general tree with a variable number of children, each of which is a room containing a vector of objects in the room and a vector of pathways into and out of the room. Each door and stairway have a pathway pointer, pointing to the room on the other side of the door or stairwell. The state machine, acting as the robot, makes the decisions concerning the traversal. The robot starts in an idle state and scans to see if it has been compromised. If it is safe to do so, i.e., no threats have been assessed, the robot scans the next room, enters the traverse state and looks for the next possible untraversed pathway from its current room. It then moves itself into the new room, collects information on the room, and places the gathered information into the tree data structure. If there are multiple untraversed pathways available, a waypoint is saved, noting that the room still has unexplored pathways available. If there is no unexplored pathway available from its current room, i.e., the end of a particular pathway has been reached, the robot returns to the first room it encountered that still has unexplored pathways and explores the next available pathway. The robot will continue traversing unexplored pathways until it reaches either a compromised or a complete state. The compromised state will force the robot to destroy all collected data on the building. If all potential pathways have been traversed, and the robot has not been compromised, it enters a complete state and returns to the operator to deliver the collected information. This information can be accessed by the user via a menu in the driver.

The general tree data structure containing the building is a tree of nodes, and each node consists of a room, a vector of pointers to each door and stairway, and a vector of pointers which contain the destination of each door and stairway in the pathways vector. An instance of the room class contains the room name, the dimensions of the room, a two dimensional vector containing the layout of the room, and a vector of all of the objects in a room.
In addition to storing the information in the tree, a mission log is created which documents everything that the robot does during the mission. This log is output to a file so that the user can see the results of and test all possible scenarios, including situations in which the robot would be compromised. This log is output to a text file and can also be viewed using the user interface menu.

While ideally the robot will use visual sensors to gather information, the sensors must currently be simulated for testing purposes. Text input files indicating objects and their locations in each room were used for this purpose. Examples from the test building are shown in figures 1, 2, and 3. Each room has a unique text file which includes the information necessary to create an instance of the room class, including the roomID (room name), the dimensions of the room, and the locations and types of objects and entities located within the room.

Figure 1: Layout of floor 1

For our testing purposes, the test input files (figure 1, 2 and 3) are used to indicate for the robot simulator the objects and locations in each room.

5. Results

The robot was able to successfully traverse the building in the test scenario and store the pertinent information. The running results of a test traversal of the example building are shown in figures 4, 5 and 6. The activity log of the robot is the output text file, while the user interface, including the menu, room displays, and mission log are depicted in the provided screenshots.

Figure 2: Layout of Floor 2

For our testing purposes, the test input files (figure 1, 2 and 3) are used to indicate for the robot simulator the objects and locations in each room.
Figure 3: Layout of Basement

The activity log of the robot is next:

Scanned Room: f1r110
Number of Entities in Room: 2
Robot continued traversal
Appended data

Scanned Room: f1r210
Number of Entities in Room: 1
Robot continued traversal
Appended data

Scanned Room: f1r209
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f1r309
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f2r410
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f2r310
Number of Entities in Room: 1
Robot continued traversal
Appended data

Scanned Room: f2r210
Number of Entities in Room: 2
Robot continued traversal
Appended data

Scanned Room: f2r309
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: b1r410
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: b1r210
Number of Entities in Room: 2
Robot continued traversal
Appended data

Scanned Room: b1r209
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f2r311
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: b1r211
Room contains no Entities
Robot continued traversal
6. Conclusion

The students were successful in completing the project. Their program was able to traverse an unknown building and store data related to the layout of the environment. Sorting and storing the information in a general tree was the key to making the information easily accessible. The next logical step with this project would be to implement the project with an actual robot, relying on visual sensors instead of simulating input with text files. Additional user tools, including an option to view the entire building at once and by individual floor would also be a worthy addition to the project, as it would facilitate easy analysis of the entire structure at a glance. A robot of this type would have many real-world applications within the military and intelligence communities, from scouting hostage situations to providing inside information concerning enemy research. This sort of information could save military lives and make missions more cost-effective by enabling the user to strategize optimally before deploying actual personnel into a hostile environment.

This has been a positive experience and had several benefits. First, it developed the intellectual mentality of the students as they were required to perform research and look at other similar published attempts and analyze them. The project required them to explore the concept of using a general tree to solve the problem of simultaneous localization and mapping. They choose to apply the requirements to ‘military’. They then came up with a design and kept on enhancing their design as they discovered better ways to implement the system. They tested their system and provided rational. Second, it enhanced their teamwork skills as they had to learn how to divide the activities among themselves. They learnt how to manage their time and how to re-plan and re-schedule activities due to other classes’ needs and personal issues. Third, although they might not be thinking about graduate school at this point but working on this project will give them insight about research and graduate studies.

Figure 4: snapshot 1 of testing program
View another room? (y/n) Y
What Room would you like to see?
f2r210
# # W # # W W W # W W # #
# . . . . . . . . . #
# . . . # E . . E . #
# . . . . . . . . . #
# # | # # # | # # # | # #
Entities: 2
View another room? (y/n) y
What Room would you like to see?
b1r410
# # | # #
# U . . #
# . . #
# # # #
Entities: 0
View another room? (y/n) y
What Room would you like to see?
f2r310
# # | # #
# . . #
| . E . |
# . . #
# . . #
# # | # #
Entities: 1
View another room? (y/n) y
What Room would you like to see?
f1r110
# # W # | # W # #
# . . . . . . . #
# . . . . . . . #
# . . . . . . . #
# . . . . . . . #
# # # # | # # # #
Entities: 2
View another room? (y/n) y
What Room would you like to see?
b1r209
# # # # #
# O O . |
# O . . #
# # # # #
Entities: 0
View another room? (y/n) -

Figure 5: snapshot 2 of testing program

***STARTING MENU***
1.) Set probabilities (not currently functional)
2.) Deploy Robot (run program)
3.) View Mission Log
4.) Review Reconnaissance Info
5.) Exit

Selection: 3
Scanned Room: f1r110
Number of Entities in Room: 2
Robot continued traversal
Appended data

Scanned Room: f1r210
Number of Entities in Room: 1
Robot continued traversal
Appended data

Scanned Room: f1r209
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f1r207
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f1r211
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f1r311
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f1r210
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f2r210
Room contains no Entities
Robot continued traversal
Appended data

Scanned Room: f2r310
Number of Entities in Room: 1
Robot continued traversal
Appended data
References


