Landmark based Localization utilizing Monte Carlo

Localization

Anthony Calandra   
Electrical And Computer Engineering  
Oakland UniversityRochester Hills, Michigan  
acalandra@oakland.edu

*Abstract*— Human generated maps are effective tools at capturing a robots starting position and providing a partially reliable starting map to aid in effective localization and mapping for an autonomous robot. Since concise starting maps are not always accessible to mobile robots, this paper investigates the use of a human created starting map containing landmarks, as well as, information on the robot’s current pose to aid in localization. This paper focuses on the utilization of image processing to extract features and robot data from provided maps and generate map data that can be consumed by our navigation stack for effective localization. The utility of this method is investigated through simulation-based experiments.

# Introduction

Localization and mapping is a heavily studied field in mobile robotics[2]. Localization being the problem of estimating the robots pose and mapping is the challenge of being able to construct and use a map of environment data, such as landmarks, to localize itself and navigate. [4] The large base of research in this field has yielded highly effective methods of localization[4], however these methods are often dependent on having preexisting maps provided [2,3]. Methods that do address the reliance on provided maps by means of utilizing human generated hand drawn maps, do so in indoor environments and show to be highly dependent on environment for effective localization, in the sense that robot ends to either preform with high success or be completely ineffective in localizing based on the environment it is placed in. This shows a dependence on environment layout for effective localization[2], our technique would ideally address this issue by use of utilizing the map for starting robot position and by helping the user create better maps rather than just free handing drawings.

In many cases robots will not have the luxury of producing or being provided a perfect map to start, such as in the case of search and rescue in natural disaster areas, as the environment would be heavily damaged and any existing maps of the area would no longer be accurate, at which point methods utilizing such would struggle to operate effectively[2].

This paper address those issues, by looking for an effective method of receiving a user generated map in a way that isn’t encumbering to the user, such that the user doesn’t need to have strong artistic ability, such as the ability to draw with proportion or perspective, as well, they shouldn’t need any considerable training. But still be able to draw a map that can provide the robot much needed environmental context when it isn’t possible for itself to acquire it from the start. We assume the users map will be inaccurate , in that it will be non-proportional, may have landmarks drawn out of place positionally, as well as, possibly contain landmarks that are overlapping or drawn twice. We try to minimize this by creating an intuitive human machine interface to allow for the person to create as accurate of a map as possible, as well as, by leveraging image processing to generate as dependable of a map as possible for the robot and to correlate the pixel distances to metical distance in the robots actual environment. To then localize we use the Monte Carlo Localization(MCL) algorithm. We had to augment MCL to fit our application, as we utilize pixel based coordinates in our system to represent landmarks and our robots pose.

# Related Work

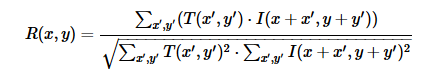
With the large body of research in mobile robot localization a  number of different methodologies for solving the localization problem have been developed[2]. In regards to my paper I’m going to split localization methods into a couple categories, localization algorithms that require prior maps and static environments. These methods include Markov Localization[10] Monte Carlo Localization(MCL)[3] and Multiple Hypothesis Localization[11][2]. One of the most popular being MCL. As described in Dellaert et al MCL is a localization algorithm that operates in two phases,  prediction and update. The prediction phase that starts with a set of particles Sk-1 computed in the previous iteration of the algorithm, then have the robots motion model applied to each particle, which represent a robots pose. Second is the update phase which utilizes measurements from the measurement model and the weights of the particles, which represent the probability of a particle prediction given a measurement. In this phase we perform resampling, where we select the highest probability particles to then be set into the new set of particles Sk. A limitation of this work are sample impoverishment, which is the selection a high weight particle multiple times, creating a lack of particle diversity.

Continuing we have methods of localization that utilize “hand drawn” maps the idea of using a hand drawn map instead is that it is a dynamic and quick solution to providing a robot a map when a complete map like a floor plan isn’t available or no longer accurate, such as if the building was damaged in a natural disaster. The area of hand drawn maps hasn't been as thoroughly researched, as stated in Behzadian et al. most works that use hand drawn maps are not using them as a tool for enhancing a localization algorithm but as a means of communication between human and robot.[2] One work that does use hand drawn maps for localization, Behzadian et al. does so by pairing a hand drawn map with Monte Carlo Localization. I this paper a augmented version of MCL(described earlier) is utilized. This algorithm is modified in two ways the robots state space is modified to have another variable to represent local deformation in the drawn map. As well, the robot is localized in a pixel coordinate frame instead of a world coordinate frame. This implementation however, should dependence on its operating environment for successful localization, meaning it is not effective solution to work in heavily varying situations. This is shown in the experimentation results, as the robots success rate was over 80% in 2 of the 9 room layouts and a success rate of under 40% in 5 of the room layouts. Showing a trend for either strong or poor performance based on its environment.

This paper looks to build on this work, by expanding on hand drawn maps to capture more information, including an initial robot pose, and by generating more consistent map data based on the provided drawn map.

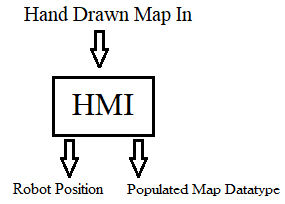
# Human Machine Interface

In this section we explain our system design and methodologies. Our system is split into two major components, the human machine interface(HMI) and our Localization system. To begin we will discuss our HMI, this is user facing feature which receives a drawn an image from a digital design tool, this image is then sent to the system for processing. The first thing we do is perform template matching algorithm using the normalized cross-correlation technique. [5] (eq 1)



Equation 1

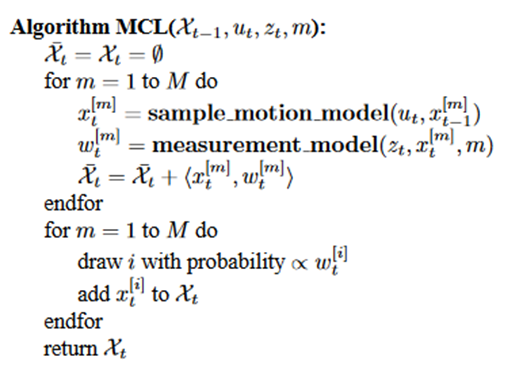
to find the drawn obstacles. We then draw bounding boxes around these obstacles to capture location data. We then perform a second round of template matching to find the robot and again draw its own bounding box, utilizing the same template matching algorithm (eq 1). this data is then used to populate a map for the robot. The map consists of the detected positions of the objects from the template matching and bounding algorithms. we determine the size by doing simple arithmetic on the boxes subtracting the x and y values in the top left and bottom right corners to get the object size, and then track the corners of the box to get it’s positions. as well we perform the same steps to capture the robots starting position. This map and robot data is then passed along.



# Localization

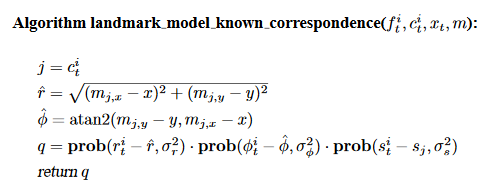
The way we designed our system is broken down into multiple parts, based on the implementation in of the following algorithm MCL[6] with adjustments made to interface with our HMI. The core portions are the main Monte Carlo Localization algorithm, our motion model, and our measurement model.

Beginning we have our MCL algorithm. This is implemented to use a feature-based map designed to be effective at operating with outdoor environments, This map contains landmarks with range, bearing, and signature attributes [7] which aid in the robots localization. The following algorithm was the base for our implantation



Equation 2

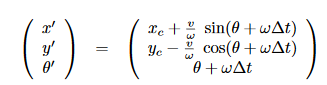
This algorithm works in two major parts the first part were we create particle predictions, pose and weight, with motion and measurement models. The second phase being the resampling phase were we draw particles from the existing particle set to populate a new refined set of particles. Resampling is done based on the particles weight, which is calculated by the measurement model.



Equation 3

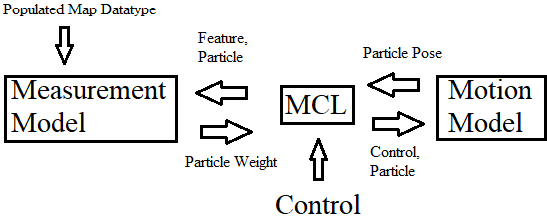
Here we calculate the predicted range bearing and signature of a landmark, can calculate a numerical probability based on the feature given a correspondence value the map and a position. In our implementation we mapped the difference between the actual and calculated landmark range, bearing, and signature to a probity, such that a small delta would carry a high probably. We also implement a random value to represent sensor noise.

Following that is the Motion Model. Our model is a simple velocity motion model, represented by the following equations, calculating the predicted X,Y, and theta positions after the movement is complete. In this equation movement is based on giving the robot a translational and rotational movement control.



Equation 4

Below you can see a system design for our localization system and some of the data flow between modules.



# Experimentation

We evaluated our system in simulation by testing the system’s ability to generate accurate maps from the human created drawings as well as the robots ability to localize accurately. Testing was done utilizing the Gazebo simulator. Our testing procedure consisted of creating 3 separate worlds for the robot to operate in, each world had 3 sets of testing where a human created a map provided it to the robot and then the robot attempts to localize.

# Results

# Conclusion

1. Conference on Intelligent Robots and Systems (IROS), 2015.
2. **1.** B. Behzadian, P. Agarwal, W. Burgard, G. D. Tipaldi, "Monte Carlo localization in hand-drawn maps", Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.
3. Frank Dellaert, Dieter Fox, Wolfram Burgard, Sebastian Thrun, "Monte carlo localization for mobile robots", Proceedings of the IEEE International Conference on Robotics and Automation, 1999.
4. Sebastian Thrun , Wolfram Burgard , Dieter Fox, Probabilistic Robotics, The MIT Press, 2006, 191-237
5. Open CV "Object Detection. Internet: https://docs.opencv.org/3.2.0/df/dfb/group\_\_imgproc\_\_object.html#ga586ebfb0a7fb604b35a23d85391329be, Dec. 23, 2016[Nov. 21,2018]
6. Sebastian Thrun , Wolfram Burgard , Dieter Fox, Probabilistic Robotics, The MIT Press, 2006, 250-263
7. Sebastian Thrun , Wolfram Burgard , Dieter Fox, Probabilistic Robotics, The MIT Press, 2006, 176-177
8. Sebastian Thrun , Wolfram Burgard , Dieter Fox, Probabilistic Robotics, The MIT Press, 2006, 127
9. Sebastian Thrun , Wolfram Burgard , Dieter Fox, Probabilistic Robotics, The MIT Press, 2006, 237
10. Sebastian Thrun , Wolfram Burgard , Dieter Fox, Probabilistic Robotics, The MIT Press, 2006, 197-205