Co-operative SLAM for Robotic Swarms

Keshav Rajasekaran
Mechanical Engineering
University of Washington
Seattle, USA

Qiubao Ye
Mechanical Engineering
University of Washington
Seattle, USA

***Abstract*— This paper addresses the problem of exploration and mapping of an unknown environment by robotic swarms. The paper proposes co-ordination techniques and SLAM strategy to maximize robot utility. We present results of a simulation of three robots working in a coordinated effort to map a simple map.**

Keywords*— *SLAM; Multi-robot; Particle filter; Exploration; Optimal coordination; Navigation

I. INTRODUCTION

Getting a robot to work in an unknown environment is a critical problem in mobile robotics. The uncertainties of any real world scenario necessitates the robot possessing robust strategies to tackle them. One of the key strategies to aid a robot in achieving its goal is for it to be able to map its surroundings and localize itself. This is a recursive problem as any accurate map requires the robot to know where it is and localization involves being aware of the map.

Simultaneous Localization and Mapping(SLAM) play a central role in any robot's effectiveness in an uncontrolled environment. Robotic swarns utilize multiple robots to accomplish a task at higher accuracy or a faster pace. Robots designed to be used in swarms are cheap, small and easily mass producable. Mapping using a swarm with an solid framework should provide vast improvments to single robot mapping. In this paper we will present strategies to effectively coordinate a group of robots while maximizing their battery-life and effectiveness of map construction. We will also test these strategies in simulation and provide comparison of a non-cooperative SLAM.

II. RELATED WORK

There has been extensive work done in the field of robot SLAM. Various techniques exist for single robot SLAM like the Extended Kalman Filter(EKF) SLAM, FastSLAM [1], FastSLAM 2.0 [2], and Graph SLAM. Though there exists a plethora of work in SLAM for single robot system, the problem of using multiple robots has not been studied by researchers at great detail.

Researchers have utilized one robot as an anchor to localize another robot that is performing the mapping [3]. This technique improves the accuracy while not utilizing both the robots very efficiently as one is never performing any exploration.

There is work done on using different sized robots to explore narrow areas and thus build a complete map [4]. There has been research into how to achieve effective coordination over multiple robots [5]. Work done using multiple robots to solve the problem assume either that the robots start within visiblity of one another or are always in communication with each other [6]. This is not a desired characteristic as constantly transimiting data is not only energy expensive and not assured unless the robots are within a certain range of each other(with no other interferences) limiting the posssibilities of effective exploration.

There has been some work done on mergeing maps from multiple robots [7]. There has been extensive work done on feature recognition[8] and optimization of maps after loop closure during the excecution of a SLAM algorithm using scan matching techniques [9].

III. PROBLEM STATEMENT

Mapping as mentioned plays a critical role in robotic performance and is one of the most imporatnt tasks when used in a disaster rescue scenario. Robots are well suited for the job as a clear understanding a drastically altered map can be obtained without risky human lives. In this paper we propose a semi de-centralised strategy to achiveve cooperative SLAM with a robotic swarm. The strategy will not only help conserve robot battery thus extending its time out in the field but also provide accurate and quick updates of the enivronment map. The problem we are set to solve invovles solving two critical problems:-

A. *Coordinated Map Building*

Our robots are assumed to not be in contact with each other unless they are relaticly close to each other. One problem that arises is how to match and merge maps from two different robots. Since they are not always in contact there must be a means by which robots can share their map with others to ensure efficient map building and facilitate faster, more accurate maps.

B. *Coordinated Exploration*

To efficiently explore the unknown areas needs the robots to co-ordinate amongst themselves to minimize overlap of areas. If multiple robots explore the same area it might provide a more accurate result but take more time to generate the entire map. The robots must be able to decide with the sparse information present how to organize the exploration so as to best add to the present map. The challenge of this problem is further increased by not having all robots in constant communication. As robots cannot be expected to know the location of all other robots at any given instant.

Solving these two problems should provide us with a good framework for a cooperative SLAM technique that can be utilized in a real world scenario.

IV. Technical Approach

To solve the afore mentioned problems we propose a semi de-centralised approach to the cooperative SLAM.

A. Coordinated Map Building

To solve this problem we design a control framework for the transmission of data. The robots are always listening to signals from a central server and trasmit only under certain conditions. This is energy efficient as listening for a signal costs way lesser that trasnmiting packets of data on a continuous basis. The conditions for robot transmission are:-

- If loop closure occurs it transmits its map to a central server which routes the information to the other robots then.
- If it senses another robot it transmits the data on references and their locations under a certain threshold of uncertainty. The robot also transmits its planned trajectory to avoid colissions.

The central server transmission makes sure that a updated version of the map is available to all robots at relevant times and the robot to robot communication makes sure some information is able to be transmitted in between central server updates. Central server communications expend more energy and so has a harder constraint to fulfill before that communcation can be intiated. Further the central server can run some optimization and merge data from multiple robots to stitch one final map that can be re-transmitted to the robots as they are always listening. Further since we are offloading the computation of the merging and transmitting to a central system, the robots can be made more agile and cheap so they can be mass produced easily.

B. Coordinated Exploration

To solve the coordinated exploration problem we will utilize a startegy called frontier assignment. As described above each robot works individually to construct an individual map of the environment which is aggragated and merged by the central server. The central server than retransimits this data to all robots. So at each instance the robot possesses with it an uptodate map of the environment.

The robot at each instance seprates the map it possesses to three categories openspace, obstacles, and unexplored region. In the unexplored region it creates a frontier at the intersection of the explored and unexplored regions. This frontier is a circle of diameter equal to the detection range of the robot. Fig (1) shows the map differentiated based on catgories and the possible frontiers for a robot. The robot then assigns itself a frontier based on euclidean distnace that it is closest too. The robot uses the frontier as a goal and repeats the process as new information is learnt of the environment.

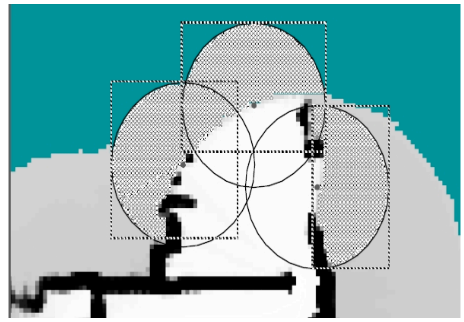

Fig 1. Map breakdown into categories obstacles in black, clear space in white and unknown region in grey[5]

The optimization is run over two factors the information gain with respect to the cost of travel. So the utility of each frontier is caluclated using the formula. In Eq. (1) $\boldsymbol{B_{ji}}$ is the utility for frontier j for robot i, $\boldsymbol{u_j}$ is the information gain, and $\boldsymbol{c_{ji}}$ is the cost for robot i to travel to frontier j.

$$B_{ji} = u_j - c_{ji} \tag{1}$$

All robots repeat this procdure on a local level at all times to make motion transitions possible. During loop closures the coordiantion can be done on a global level. This is accomplished by the central processor as it is more intensivie. Upon receiving data on a loop closure the central server can query all robots about current location and obtain all possible frontiers in the map so as to assign robots to individual frontiers to achieve a golbal optimum.

This optimization over these multiple combinations is obtained by the use of the Hungarian method [10]. There are a few edge cases that need to be adressed.

1) When the number of frontiers are greater than number of robots

In this case the frontiers can be assigned to maximize co-operation which would enable faster loop closures and enable more sharing of information. It can also be assigned to maximize exploration and so covering more of the map quickly and get lucky with certain loop closures. It would be best to find an equilibrium based on scenario. In a search and rescue scenario it would be better to bias towards loop closures so information is available at a faster rate.

2) When the number of grontiers are lesser than number of robots

Rather than assigning just the exact number of robots the other robots can also be assigned to a frontier which is closest to them. This assignment ensures that in case of a branching path in the unexplored region then there are multiple robots present to

- handle that situation. Unless the robot is needed to stay to serve as a beacon to humans it is asked to move.

This frontier assement model would help achieve the necessary co-ordination with limited communcaition. The semi de-centralised system has other advantages. The system is not completely reliant on full scale communication but the efficiency depends on it. Though all robots need to be able to communicate with the central server to achieve global optimum even if a robot was to fail to communicate the remaining system still works in a global optimumn for the sub sytem. So if a robot does loose its communication or doesn't work for whatever reason the system can compensate for its loss and still perform at a local optimum and adapt again if communication is re-established. The system hits a sweet balance between flexibility and optimallity.

C. SLAM Strategy

For our co-operative SLAM we propose to use a FastSLAM or a particle filter SLAM. The equations for the same can be found in extensive detail in [1]. FastSLAM has a few advantages.

- Due to the use of particle filters mutiple data associations for landmark data is possible.
- Each particle keeps only N 2x2 EKFs where N is the number of landmarks so data stored is smaller.
- The filter can be vectorised so computation time is highly reduced.
- The algorithm runs in O(M log N) time where M is the number of particles which is better than $O(N^2)$ time of a vanila EKF while being more robust as mentioned in point 1 of the list.

So we use a particle filter for every robot that keeps track of each landmark it comes across. During loop closure a variety of techniques can be used to reconstruct the map based on the particle history. During resampling the uncertainty in the observation decides how peaked or spread out the sample is. If a landmark's location is certain the resampling would result in a peaked distribution.

More information on the implementation of our strategy will be presented in the following section on our simulation environement and setup.

V. Simulation Environment

To view our SLAM strategy in practice we tested the strategy in a simulation setup. The SLAM technique described above was tested on a three robot system in a closed map. The simulation environment we designed for this purpose is described in this section to better explain the results.

A. World

The simulation happens in a 2D grid world of resolution grid size 0.1 units. A grid is referenced by the coordinates of the point at the bottom left corner. The world has an occupancy map described by color. White indicates open space and black indicates obstacles. Other colors are used to indicate references. Each reference is identified by a unique color. Though this may seem unreasonable techniques like SIFT can identify features to a great deal of accuracy and since we implemented a FastSLAM approach multiple data hypothesis can be tackled in the real world. So a unique reference identifier isn't that much of a leap of faith in the simulation environment.

B. Robots

The robots being simulated are circular robots with a 0.1 units diameter. The robot can traverse along X and Y axes and rotate in the plane. Each robot is described by its unique color. This is similar to how a highly fluorescent cylinder on a robot is used to identify it in some of the references [6][7]. The robot has noise in its dynamics. The noise is not coupled i.e. if a robot is moving in a particular direction the noise is added only in the direction of motion. For the simulation we used a Gaussian noise of mean 0 and variance 0.1. This ensures that the noise caused doesn't change the robot pose dramatically. There are three similar robots in the simulation.

C. Sensor

For our sensor we simulate a modified LASER rangefinder or LiDAR. For our simulation we assume that the sensor is not only able to tell us the orientation of the scan and the range in that direction but also the color of the cell it is reading. This is to enable picking up features and other robots. In the real world other techniques can be used to identify features as described before and this sets up an easy simulation setting. LiDARs usually also come equipped with an intensity measure that can be used to track fluorescent objects in the real world. The sensor emits rays at every 5^o from the robot heading.

For the sensor model we modeled it as a Gaussian with mean 0 and variance 0.01. This is coupled with a max reading probability. The sensor model is shown in Fig (2).

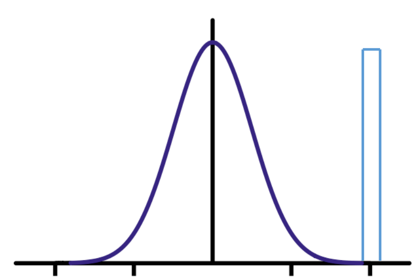

Fig 2. Sensor model

VI. Results

A link to the full video to the simulation can be found in the appendix. In this section we will discuss a few critical parts of the simulation.

A. *Central Map Update*

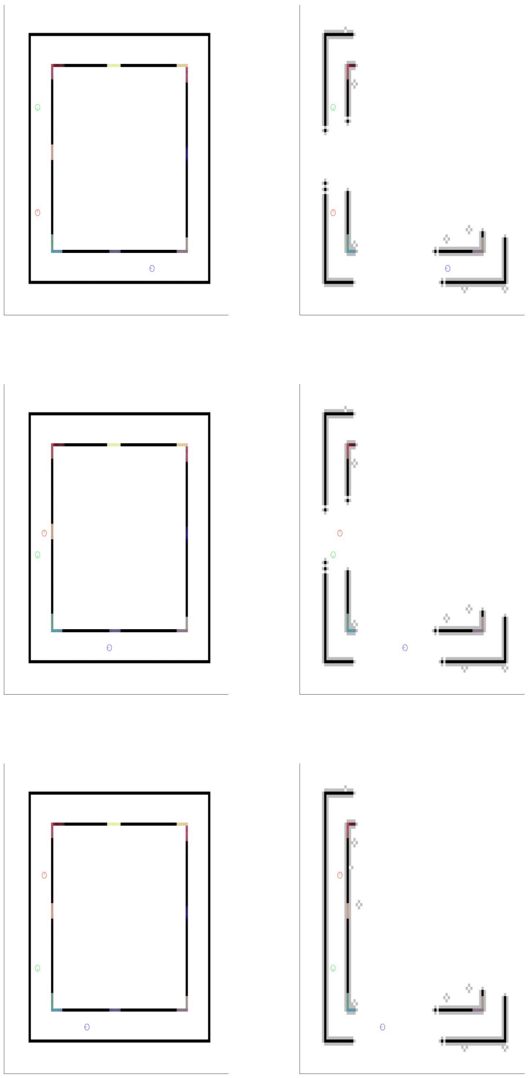

Fig 3A(top) Robots losing contact with their anchor landmarks
3B(middle) No map updates to central server as the landmark is uncertain
3C(bottom) Loop closure and central map update

As seen by Fig 3A-3C there are no central map updates until a reference of uncertainty below a certain threshold is observed by the robot. So there is no communication initiated until a loop closure. In our simulation our part is recreated by the best particle at the point of loop closure. More sophisticated techniques would provide accurate results in the real world. Though the motion is noisy the resolution of the sensor provides us with a complete map. It should also be noted that loop closures occurred at landmarks different from their initial anchors due to the cooperative nature of the SLAM framework developed.

B. *Avoiding each other*

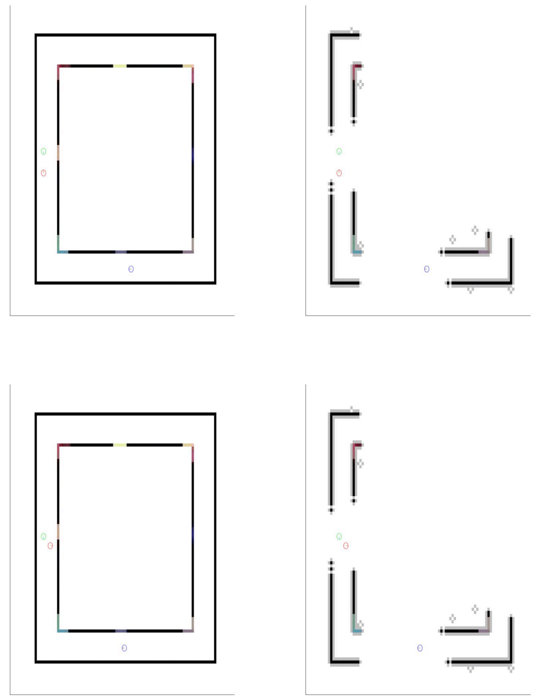

Fig 4A(top) Robots are within range of communication
4B Only one robot executes path change to avoid crashing

Due to exchange of information only one robot needs to excecute the evasive maneuver. This is picked at random. If the new paths don’t intersect then they continue on their merry way else new paths are re-calculated.

C. *Frontier Assignment*

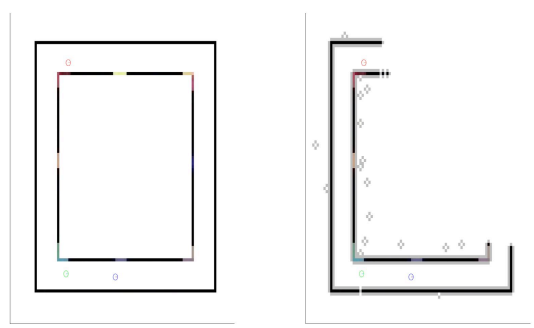

Fig 5. Two robots heading to one frontier

Two robots are sent to the same frontier to tackle any unexpected branching in the map.

The simulation shows that the SLAM framework provided performs well and that all the critical issues that were discussed were solved.

VII. Conclusions

The framework set up in this paper performs well to solve the problems of co-operation and coordination. As seen loop closures occur quicker than with a single robot. We tested both cases in the simulation and the single robot needs almost four times the amount of time to have its first loop closure. In situations like search and rescue this could be critical. In case of the cooperative SLAM emergency services can start acting in the area that has been mapped while other areas are being mapped this means the cooperative SLAM facilitates better reaction time. Since the robots aren't constantly transmitting data they can spend more time in the field thus improving performance overall. Since individual robots are expendable in our framework robots can hold their position at points of interest say an injured person for example. These robots can then act as a beacon for rescue teams in zones where visibility is compromised.

If this framework is coupled with other work like using visual sensors to identify people [11]. The maps can also indicate possible victims still trapped in the disaster zone and requiring immediate attention. Integrating the two would help tremendously in bettering rescue efforts in disaster prone areas.

Our strategy works well in achieving optimal use of the robotic swarm without every robot in the swarm being a necessity. The map is generated in parts but quickly and in sections large enough to be useful by themselves. Our framework has been tested in simulation but needs to be tested on actual hardware to characterize and tune its performance in a real world scenario. In the simulation we did not perform any optimization on the path being traced after loop closure. Adding an optimizer would increase performance on hardware. Further the central processor in our simulation built a map using measurements of walls that were closest to the robot in direction of measurement. Basically a wall behind a wall is as good as the second wall not existing. So an actual merge algorithm as mentioned in the references would make a good addition when testing on hardware.

References

[1] FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem

[2] FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges

[3] I. Rekleitis, G. Dudek, E. Milios. "Accurate Mapping of an Unknown World and Online Landmark Positioning." In *Proc. of Vision Interface 1998*, pp. 455-461, Nagoya Japan, 1997

[4] K. Singh and K. Fujimura. "Map Making by Cooperating Mobile Robots". In *Proc. Intl. Conf. on Robotics and Automa- tion*, 1993.

[5] Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., & Younes, H. (2000, July). Coordination for multi-robot exploration and mapping. In *AAAI/IAAI* (pp. 852-858).

[6] Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., & Stewart, B. (2006). Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, *94*(7), 1325-1339.

[7] Zhou, X. S., & Roumeliotis, S. I. (2006, October). Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1785-1792). IEEE.

[8] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). Ieee.

[9] Diosi, A., & Kleeman, L. (2005, August). Laser scan matching in polar coordinates with application to SLAM. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3317-3322). IEEE.

[10] Bruff, D. (2005). The assignment problem and the hungarian method. *Notes for Math*, *20*.

[11] Treptow, A., Cielniak, G., & Duckett, T. (2005, August). Active people recognition using thermal and grey images on a mobile security robot. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2103-2108). IEEE.

Appendix

Link to simulation video - https://drive.google.com/file/d/0B0DhO5Zfos14QkhFZ3dHX3hlWTg/view?usp=sharing