A Survey of Implementation of Multi-Robot Simultaneous Localization and Mapping

Jaydeep Kshirsagar
Electrical and Computer Engineering
University of North Carolina at Charlotte
Charlotte, NC
jkshirsa@uncc.edu

Sam Shue
Electrical and Computer Engineering
University of North Carolina at Charlotte
Charlotte, NC
slshue@uncc.edu

James M. Conrad
Electrical and Computer Engineering
University of North Carolina at Charlotte
Charlotte, NC
jmconrad@uncc.edu

***Abstract*— For any robot to effectively traverse its environment, it requires a map and its location along with the orientation within that map. Often a map may not be available if the robot is presented with an unfamiliar or unknown environment. This requires the robot to construct the map as it localizes itself and navigates. This problem of concurrently building a map and localizing a robot in that environment is defined as Simultaneous Localization and Mapping (SLAM). Many SLAM applications already exist for single robot such as navigating the unmanned mines and exploring the sites with natural calamities. With the advent of swarming robots that must interact with each other, extensive research is being conducted for the extension of SLAM problem to multiple robots known as Multi-Robot SLAM (MRSLAM). In the MRSLAM environment, the efficiency is improved, and time constraints are reduced, but it's implementation is restricted due to constraints such as communication bandwidth, memory requirements and problems faced during map merging and co-ordinate transformation. This paper represents many different approaches that are being used for an implementation of MRSLAM.**

***Index Terms*—Multi-Robots, Localization, Mapping, Simultaneous Localization and Mapping, Full SLAM, Online SLAM, Map merging, Rao Blackwellized Particle Filter, Sparse Extended Information Filter.**

I. INTRODUCTION

For a robot to traverse in any environment, it should be well acquainted with the environment features and its own position with respect to those features in that environment. The process of collecting information about the environment features and building a virtual environment in local reference frame is commonly referred to as 'Mapping'. Similarly determining its position relative to its collected features is known as 'Localization'. For efficient navigation of a robot, information regarding the features of the environment is very much essential. An efficient and error free map cannot be built without an information regarding precise location of robot itself and at the same time to determine a precise location of a robot, accurate map is required [5]. To resolve this 'chicken or egg' problem, an innovative approach, called as 'Simultaneous Localization and Mapping' (SLAM), is used. The SLAM can be defined as a process of developing a map of an environment and simultaneously localizing itself in it. SLAM consists of four major tasks: perception, localization, cognition and motion control. Perception can be viewed as the robot's interpretation of its environment, where robot tries to collect information and process information from outside world through the use of various sensors, such as cameras and range-finders. Localization is an estimation of robot pose using the data collected from sensors and previously constructed map. Data collected from the sensors, is used by robot to estimate the current position during motion as well as to generate a map of an environment. Cognition part plays a significant role in analyzing the environment and deciding the response of the robot. Motion control of a robot can be defined as a process of traversing in an already mapped or unmapped environment. Different exploration strategies will be using motion control to define the trajectory of the robot.

SLAM can be implemented for several different topologies such as: SLAM for a system having single robot in a static environment, a system with a robot and dynamic environment and a system with multiple robots in an environment. Extensive research is already done on A 'Single-Robot SLAM' (SRLAM or Classical SLAM). Many techniques including use of recursive estimators such as Extended Kalman Filter (EKF) [28], Rao- Blackwellized particle filter, and Sparse Extended Information Filter (SEIF) have been developed to solve the problem of localization in SRSLAM whereas research regarding mapping has brought into focus the several types of maps like topological maps, metric maps, hybrid maps [25,26].

SLAM problem can be categorized into two main parts as: Online SLAM and Full SLAM. In online SLAM, the posterior is estimated over only the current pose of the robot. Whereas in Full SLAM, posterior is calculated over entire path i.e. all the previous readings are considered for the estimation of the posterior. Following formulae states mathematical expression for the calculation of posterior with both SLAM categories.

Posterior of a robot: $p(x_{1:t}, m|z_{1:t}, \mathrm{u}_{1:t-1})$

Where,

$x_{1:t}$: pose of a robot from time T=0 to T=t

978-1-5386-6133-8/18/$31.00 ©2018 IEEE

m : map
$z_{1:t}$: Observations from time T = 1 to T = t
$u_{1:t-1}$: controls from time T = 1 to T = t-1

For Full SLAM:

$$p(x_t, m|z_{1:t}, u_{1:t-1}) = \int\int\ldots\int p(x_{1:t}, m|z_{1:t}, u_{1:t-1})\, \mathrm{dx}_1\, \mathrm{dx}_3 .. \mathrm{dx}_{t-1}$$

For Online SLAM:

$$p(x_t, m, c_t|z_{1:t}, u_{1:t-1}) = \int\int\ldots\int \sum_{c1}\sum_{c2}..\sum_{ct-1} p(x_{1:t}, m|z_{1:t}, u_{1:t-1})\, \mathrm{dx}_1 \ldots \mathrm{dx}_{t-1}$$

In online SLAM, posterior is obtained from full slam posterior by integrating out past robot poses and summing over all past correspondences.

Though very efficient techniques are developed for classical SLAM, they cannot be immediately extended to the multi-robot SLAM (MRSLAM). In multirobot SLAM, observations of a robot are dependent on every other robots' pose, which makes the posterior of one robot to be dependent on the trajectory or pose of the other. Major challenges faced during implementation of MRSLAM are: posterior estimation from data gathered by different robots, limitations due to unreliable wireless sensing network, coordination between robots and individual frame of reference to shared world representation, complexity and memory requirements and dynamic environment [2,20]. Knowledge of initial positions of robot also plays a key role in underlining techniques. If initial positions of a robot are known then MRSLAM is a simple extension from SRSLAM [10,22] but for unknown initial positions consistent integration over a common map is difficult. For multi-robot SLAM (MRSLAM) with known initial robot position, there must be a third agent monitoring pose of all the robots or they should be initialized to a known position in a common map. In MRSLAM with unknown initial positions, every robot will take sensor measurements and fuse them into common map including the measurements by other robots.

Solutions to MRSLAM can be implemented in two ways as: Centralized MRSLAM [Fig.1a], and Decentralized MRSLAM [28] [Fig.1b]. In centralized system all the robots are connected to a central server. Each robot will gather its local information and transfer this data to a central server. This Central server is responsible for collecting and fusing all the data into a single global map and transfer back this fused data (map) to all the robots. This method imposes an immense amount of overhead due to continuous high exchange of copious amounts of data between server and robots. Its complexity is directly related to the number of robots connected to server because it becomes impractical to connect many robots to single server and it requires continuous monitoring of each deployed robot. One major drawback of this system includes functionality dependency on a single unit like server, because failure in server functionality will cause the entire system to breakdown. On the other hand, decentralized systems consist of many robots implementing its own SLAM and generating local map respective to the local frame. And whenever there is a meeting of two robots, data will be exchanged between them and this data will be used for mapping individual map. This approach implements more flexible and scalable algorithm for MRSLAM.

This paper showcases different techniques used to solve problem of MRSLAM and several types of maps that can be implemented along with it.

II. PROBLEM OF LOCALIZATION AND SLAM

Localization in SLAM is defined as computing position of a robot in known or unknown environment. The main challenge faced in localization is the design of the transformation matrix. Robot percepts the world through its sensors, so environment information is perceived with reference to the robot's perspective, or local frame, and must be transmitted to the map's reference frame, or the global frame. To resolve the differences between these two reference frames and to convert robot's local coordinate system to global coordinate system a transformation matrix is computed. Usually transformation matrix considers the relative pose - delta distance (distance between two robots), theta (angle) and noise in the observations from the sensors. Sensors that are used as an interaction between environment and robot can be categorized into two types as: Exteroceptive sensors and proprioceptive sensors. Exteroceptive sensors like Light Detection and Ranging (LIDAR) and sonars use acoustic or electromagnetic energy to sense and to interact with the environment. Whereas, proprioceptive sensors do not rely on environment for sensing and they use system's internal parameter for measurement. Odometer can be categorized as an example of proprioceptive sensor as it uses wheel encoders to estimate the current pose of a robot. Sensors are susceptible to noise and they may induce some error in data. As this error may affect the precision of localization, many filters are used to eliminate those errors. State of art solution to MR-SLAM implements filters like Kalman filter, Extended Kalman filter, Particle filter to eliminate error and increase the precision of Localization.

(a) Centralized system (b) Decentralized system

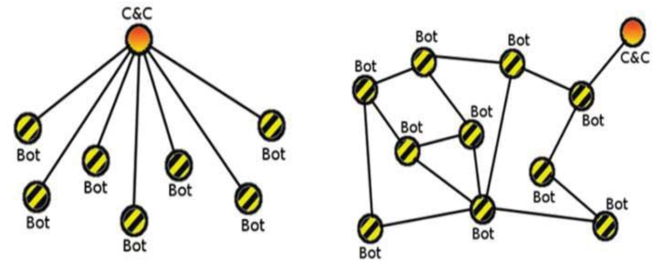

Fig 1. Different types of MRSLAM

Howard [1] put an online approach for MRSLAM using Rao-Blackwellized particle filter with known initial poses as well as with unknown initial poses. His algorithm can be explained using two robots exploring in the environment. Whenever there is an encounter between these two robots, respective poses along with the data are fed to filter. Filter generates two virtual bots and these measurements are fed to these virtual robots in reverse time order.to build a common map. For MRSLAM each particle will be a tuple of < x_{t1}^i, x_{t2}^i, m_t^i, w_t^i >, where x_{t1}^i, x_{t2}^i are robot 1 and robot 2 poses respectively, m_t^i is map and w_t^i represent weight of that particle. At the starting point each robot will implement individual SLAM with occupancy grid mapping and store robot pose measurements in a data structure during continue independent exploration. Now whenever Robot 1 and Robot 2 are in line of sight of each other or able to detect each other, Robot 2 will establish a communication and transfer data to Robot 1. Now robot 1 will update the tuple and occupancy grid. For this algorithm to work, relative pose of robot 2 with respect to robot 1 is considered to define transformation matrix. This algorithm assumes that pose of robots will not interfere with measurements of each other and it is implemented in algorithm to ignore those readings.

For multirobot SLAM with unknown initial poses, encounter will be detected before incorporating the data into the map. In this approach only one robot with arbitrary pose will start mapping and wait for an encounter with each additional robot before taking the data. All the subsequent encounters are neglected. Fig 2. Shows multi robot SLAM with unknown initial poses using particle filter. At some point in time t, delta_2^1 represent relative pose of robot 2 with respect to robot 1 which is used to incorporate data into the map along with coordinate transform. For successful implementation of this filter author makes approximations as follows:

1. Conditional dependencies between robot trajectories is ignored and treated as independent variable.
2. Uncertainty in the relative pose (delta_2^1) is small.

All the data collected by first robot will be considered or incorporated into the map, while data obtained only after

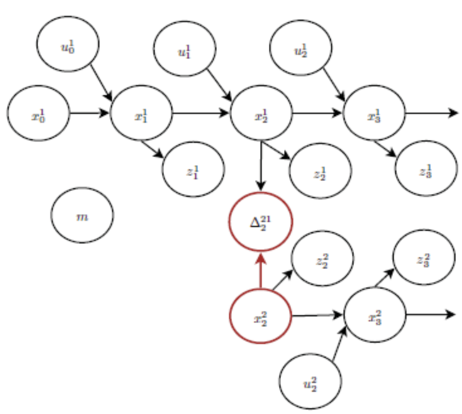

Fig 2. MRSLAM for particle filter [1]

encounter is considered from other robots.

Another approach suggested by Thrun [7,24] to solve MRSLAM is to implement Sparse Extended Kalman Filter (SEIF) with known initial correspondences. SEIF uses Bayesian approach and instead of localizing robots in each other's map, this solution focuses on comparing maps acquired by robots. SEIF is an intermediate algorithm between online SLAM and full SLAM. SEIF, just like online SLAM, implements solution to determine posterior over current pose and a map. At the same time SEIF takes advantages from full SLAM algorithm by keeping all the information. SEIF represents SLAM posterior in the natural parameters of multivariate Gaussians such as 'information matrix' (Ω_t), and 'information state' ($€_t$).

SEIF is implemented in four steps: a motion update step, a measurement update step, a sparsification step, and a state estimation step [30]. A measurement update step updates the local links in information matrix and information vector. It links features to the current robot pose. And this amendment is restricted to only observed features from current pose. A motion update step does not consider measurements from sensors. It is only estimation and updating information matrix as well as information vector based on control signal (u_t). It takes previous information matrix and information vector as an input and incorporates motion control signal to update information vector and information matrix. This step primarily implements filter. It plays key role because it eliminates past pose estimates and differentiate SEIF from full SLAM. Before robot motion, link is established between features and pose. So, after robot motion these links are weakened as robot motion introduces uncertainty in the information state. This uncertainty in the pose of robot relative to map results in the loss of information. This information is not entirely lost because of the fact that we lose information about pose relative to map, but we still have the relative information of features in the map. This causes filter to shift the information from pose-feature to feature-feature pair. This step updates direct links between feature pairs. Sebastian Thrun suggest that this transformation of link is possible only if feature has a active measurement link between itself and pose. This feature of SEIF helps to limit the computational complexity. So, by controlling the number of active feature at any point of time we can control motion update and computational complexity. This transformation of information from motion link (pose to feature) to measurement link (feature to feature) is also called as 'Sparsification'. One important advantage of this algorithm is time required for sparsification is independent of size of the map.

Last step in SEIF deals with state estimation (μ). State can be estimated using equation $\mu = \Omega_t^{-1} * €_t$. where Ω_t is 'information matrix' and $€_t$ is 'information state'. This step introduces high time complexity in the algorithm and so to avoid this SEIF exploits 'relaxation algorithm'. In relaxation algorithm each step is updated based on best estimates of nearby elements in the information graph. Due to sparse information in the SEIF, each update will require a constant time and the updates state vector can be defines as approximation in all updating step instead of using correct mean estimate. Accuracy of result of SEIF and computational efficiency are decided by the degree of sparseness in SEIF.

Extending this approach to MRSLAM can be viewed as an implementation of SRSLAM in a local reference frame with respect to each robot and then exchanging of data between robots. Additivity and locality of update step makes this algorithm amenable for MRSLAM. Thrun in [7] explains the fusion of maps faces mainly two problems such as: each robot maintains its own co-ordinate system so transformation from one robot reference frame to another is nonlinear and along with it data association problem must be solved while fusing the maps. Suppose we have two robots j and k each with information as $(\Omega_t^j, €_t^j)$ and $(\Omega_t^k, €_t^k)$ respectively. To fuse maps, we require transformation vector with linear displacement matrix 'd' rotational difference element 'a'. Using this transformation matrix, we can map position of robot j and feature in its map to k robot's co-ordinate system with a rotation by 'a' and displacement by 'd'. Equations for map fusing can be written as:

$$\Omega^{j \rightarrow k} = A * \Omega^j * A^T$$

$$€^{j \rightarrow k} = A * (€^j - \Omega^{j \rightarrow k} d);$$

Where,

$$A = \begin{bmatrix} \cos a & \sin a & 0 \\ -\sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$d = [dx \quad dy \quad 0]^T.$$

Fused maps are now having a common coordinate system. Information state and information vector both can be added to find the joint information state. For a correspondence list with identical feature 2 and 4 joint map is computes by collapsing as given below:

$$\begin{bmatrix} \Omega 11 & \Omega 12 & \Omega 13 & \Omega 14 \\ \Omega 21 & \Omega 22 & \Omega 23 & \Omega 24 \\ \Omega 31 & \Omega 32 & \Omega 33 & \Omega 34 \\ \Omega 41 & \Omega 42 & \Omega 34 & \Omega 44 \end{bmatrix} =$$

$$\begin{bmatrix} \Omega 11 & \Omega 12 + \Omega 14 & \Omega 13 \\ \Omega 21 + \Omega 41 & \Omega 22 + \Omega 42 + \Omega 24 + \Omega 44 & \Omega 23 + \Omega 43 \\ \Omega 31 & \Omega 32 + \Omega 34 & \Omega 33 \end{bmatrix}$$

$$\begin{bmatrix} €1 \\ €2 \\ €3 \\ €4 \end{bmatrix} \rightarrow \begin{bmatrix} €1 \\ €2 + €4 \\ €3 \end{bmatrix}$$

The results of above algorithm are shown in Fig 3 and Fig 4.

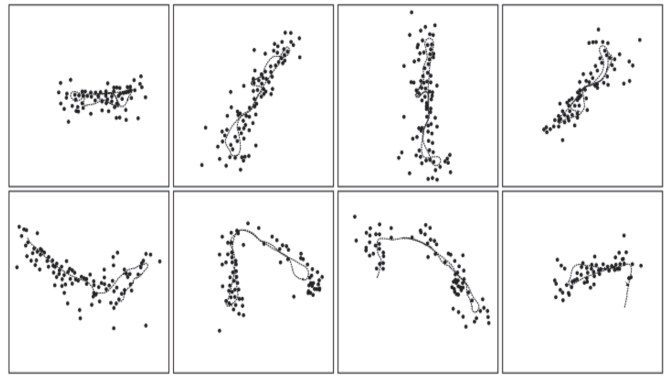

Fig 3. Eight local maps obtained by splitting map [7,23]

S. Thrun divided data collected from single robot to eight local subsets (Fig 3) and runs this algorithm on them to produce a common map (Fig 4).

III. PROBLEM OF MAPPING IN SLAM

SLAM is defined as concurrently building map of the environment from sensor measurement and using this map to obtain estimate of sensor measurement and using this map to obtain estimate of position of a robot. For a robot to traverse in any given environment mapping is the most important process, where a robot will integrate data from sensors and process that to build a map. In SR-SLAM, mapping is limited to a robot itself but extension of the same techniques in MR-SLAM is challenged by the factors such as: constraints on memory requirements, coordination between robots, Data association problem [31] - where robots have to decide if the current observation belong to the previously observed same landmark or different landmark of similar properties, Close loop problem and transformation of reference frames Consider an environment for two robots with unknown initial locations.

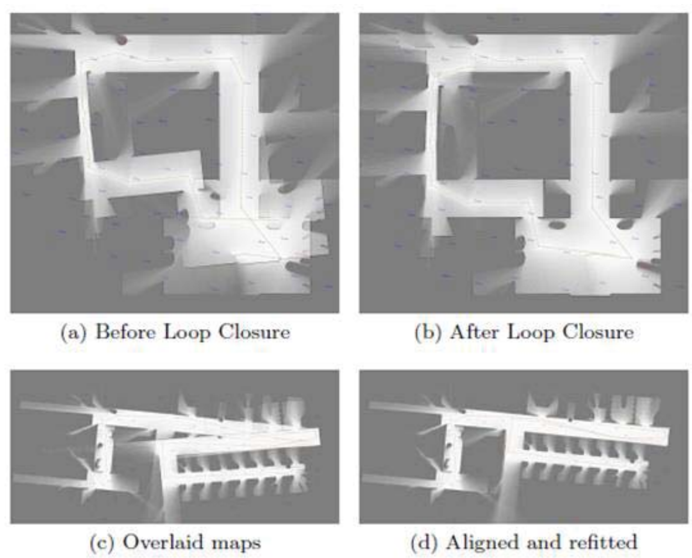

Fig 5. Loop Closing and Loop Merging [3]

Generally, each robot will start exploring and use their sensor data from sensors like laser scan, Pan-Tilt camera [2 20] to build a local map. Each robot maintains its own local map with respective local reference frame. Once communication is established between two robots, each robot will transfer previous readings to fuse into a common map (global map).

Andrew Howard put a similar approach in [1]. In any multi robot environment, a starting point for a solution can be an implementation of single robot SLAM, unless and until there is an encounter with another robot. At a starting point, no robot will know the initial locations of any other robot, and so each robot will start individual SLAM. Once it encounters another robot it will localize other robot with relative to the self-position. Each robot will maintain a data structure, either a queue or a listing, to store the observations by laser and odometry sensors (sometimes odometry along with laser also known as 'Lodometry' is also used for precision). These reading can be divided into casual and acasual readings (before and after an encounter with another robot). This data is then fused into common map avoiding the similar observations and incorporating the associated data that was recorded by the robot before an encounter. Each particle used to develop a map and localize through filter will have a tuple <x, m, w>, where x is a robot pose at time t, m is a map generated at time 't' and 'w' is the particle weight which eventually decides the validity of a respective particle. Particle having large weights are incorporated in the map and those with the low weights are discarded [1].

Several types of maps can be implemented while solving a problem of SLAM, amongst which topological maps (graph) are highly scalable due to their compact description and on the other hand grid-based maps have higher resolution but at the cost of larger amount of memory [3]. Pfingsthorn et. al. put hybrid approach [3], where sensor data without any processing will be stored at nodes and edges will represent the transformation between these nodes. In MR-SLAM each robot will have an individual part of a map (either same or different), this method helps to merge different section of maps by correctly connecting the proper nodes.

Two important concepts to be considered during merging maps are: loop closing and overlap detection. `Loop closing` is a process of connecting two separate regions of a map, which also helps to reduce the cumulative error [29]. Seemingly two different observations in a map may belong to the same location in the environment. When displacement between two nodes is computed, inconsistency introduced by the this is resolved by incorporating the additional information regarding common landmark into the global map [3, 11, 8]. Map merging is a process of connecting two different regions into the common or global map. It is very difficult to decide how two different maps from two robots are connected to each other. For this reason, a common point of reference is decided, and map overlap is checked.

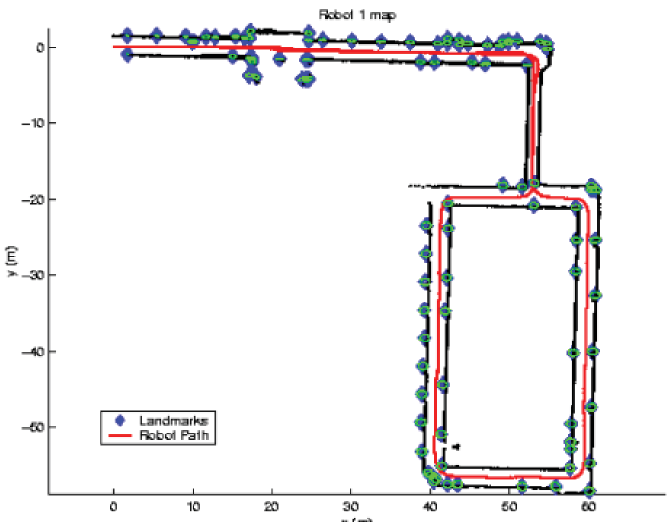

Fig 6. Map by robot 1 before map merging [4].

Every robot has its own local reference frame and therefore to merge maps in MRSLAM, a transform between their coordinate frames must be determined either by searching for common landmarks in two maps or by robot to robot measurement when robot meets [4]. Zhou et. al. proposes another approach in which, when two robots are in communication range but with unknown relative locations, one robot will receive data from another robot and try to estimate its location by matching the received scan patch with respect to its own map. These robots then try to meet at a location on this assumed common map and if and only if they succeed, maps will be combined permanently, otherwise maps will be discarded and exploration will continue. For efficient merging

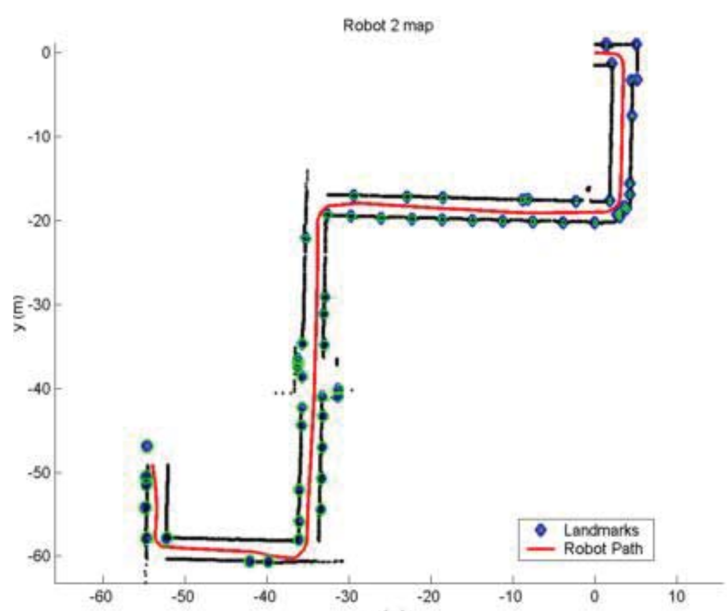

Fig 7. Map by robot 2 before map merging [4].

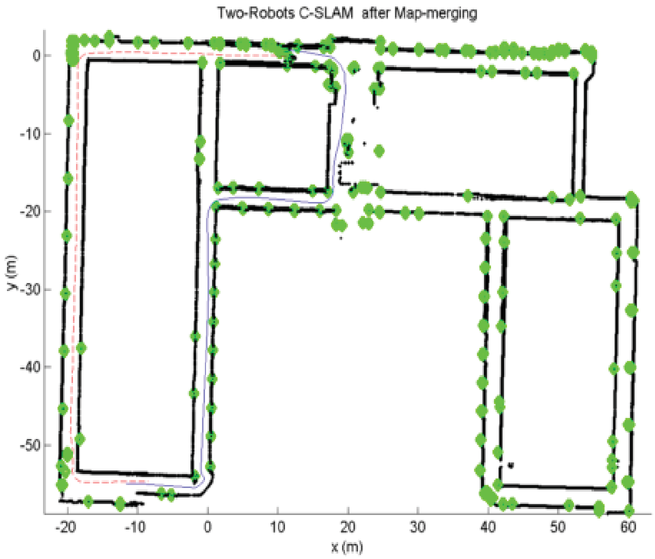

Fig 8. Common map after map merging [4].

of two maps, transformation between their respective frames must be determined. To determine the transformation, vector estimate of both robot poses and robot to robot distance and bearing measurements are essential. Both the robot implement Kalman Filters for individual mapping before rendezvous and combined map will be computed from the above information. As these individual maps can contain common regions, map overlapping, and matching is also performed to reduce alignment errors and state vector matrix. There are two main techniques put forward for the identification of ideal landmark 1) Nearest neighbor (NN) [4] and 2) Joint Compatibility Branch and Bound (JCBB) algorithms [23]. NN simply search for the closest two landmarks from two maps where as JCBB find the largest number of many compatible pairing. NN is easier to implement whereas JCBB is robust but with high computational cost. Fig 6, Fig 7, and Fig 8 shows map generated during MRSLAM. Local map generated by robot 1 is shown Fig 6 while local map generated by robot 2 is shown in Fig 7. Then these two maps are compared for shared areas and merged together to form a global map shown in Fig 8.

IV. CONCLUSION

This paper tries to cover most of available solutions for MRSLAM architecture with respect to localization techniques and mapping algorithms. It can be deduced that the major difference and tradeoff is between computational complexity and memory requirements of the solution. Some maps require large memory space but are more efficient and precise. Topological maps require much less memory, but map topologies like 2D geometric maps or occupancy grip maps will be stored with less details. Similarly, some filter may have higher time complexity or same time complexity and higher computational complexity but can be stated as a state of art estimators for robot localization.

Thrun approach in [7] can be implemented with less computational complexity because robot's initial pose is not required to build a precise map. This algorithm fuses different maps by identifying shared areas and correct alignments. Extending EKF for MRSLAM is more complicated because its computational complexity quadratically increases with number of landmarks and EKF considers gaussian noise.

For MRSLAM, RBPF can be a good alternative because it produces a precise information map with a little higher computational cost depending on the number of particles.

V. REFERENCES

[1] Howard, Andrew. "Multi-robot simultaneous localization and mapping using particle filters." The International Journal of Robotics Research 25, no. 12 (2006): 1243-1256.

[2] Carlone, Luca, Miguel Kaouk Ng, Jingjing Du, Basilio Bona, and Marina Indri. "Rao-Blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication." In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 243-249. IEEE, 2010.

[3] Pfingsthorn, Max, Bayu Slamet, and Arnoud Visser. "A scalable hybrid multi-robot SLAM method for highly setailed maps." In Robot Soccer World Cup, pp. 457-464. Springer, Berlin, Heidelberg, 2007.

[4] Zhou, Xun S., and Stergios I. Roumeliotis. "Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case." In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pp. 1785-1792. IEEE, 2006.

[5] Grisetti, Giorgio, Gian Diego Tipaldi, and Cyrill Stachniss, Wolfram Burgard, and Daniele Nardi. "Fast and accurate SLAM with Rao–Blackwellized particle filters." Robotics and Autonomous Systems 55, no. 1 (2007): 30-38.

[6] Chang, H. Jacky, CS George Lee, Y. Charlie Hu, and Yung-Hsiang Lu. "Multi-robot SLAM with topological/metric maps." In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, pp. 1467-1472. IEEE, 2007.

[7] Thrun, Sebastian, and Yufeng Liu. "Multi-robot SLAM with sparse extended information filers." Robotics Research (2005): 254-266.

[8] Fox, Dieter, Jonathan Ko, Kurt Konolige, and Benjamin Stewart. "A hierarchical bayesian approach to the revisiting problem in mobile robot map building." Robotics Research (2005): 60-69.

[9] Eliazar, Austin, and Ronald Parr. "DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks." In IJCAI, vol. 3, pp. 1135-1142. 2003.

[10] Williams, Stefan B., Gamini Dissanayake, and Hugh Durrant-Whyte. "Towards multi-vehicle simultaneous localisation and mapping." In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, vol. 3, pp. 2743-2748. IEEE, 2002.

[11] Howard, Andrew, Gaurav S. Sukhatme, and Maja J. Mataric, "Multirobot simultaneous localization and mapping using manifold representations", Proceedings of the IEEE 94, no. 7 (2006): 1360-1369.

[12] Ko, Jonathan, Benjamin Stewart, Dieter Fox, Kurt Konolige, and Benson Limketkai. "A practical, decision-theoretic approach to multi-robot mapping and exploration." In Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003

IEEE/RSJ International Conference on, vol. 4, pp. 3232-3238. IEEE, 2003.

[13] Vincent, Regis, Dieter Fox, Jonathan Ko, Kurt Konolige, Benson Limketkai, Benoit Morisset, Charles Ortiz, Dirk Schulz, and Benjamin Stewart. 2008. “Distributed Multirobot Exploration, Mapping, and Task Allocation.” Annals of Mathematics and Artificial Intelligence 52 (2–4): 229–55. doi:10.1007/s10472-009-9124-y.

[14] Fenwick, John W., Paul M. Newman, and John J. Leonard. "Cooperative concurrent mapping and localization." In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on, vol. 2, pp. 1810-1817. IEEE, 2002.

[15] Roumeliotis, Stergios I., and George A. Bekey. "Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization." In Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on, vol. 3, pp. 2985-2992. IEEE, 2000.

[16] Roumeliotis, Stergios I., and George A. Bekey. "Distributed multirobot localization." IEEE Transactions on Robotics and Automation 18, no. 5 (2002): 781-795.

[17] Durrant-Whyte, Hugh, and Tim Bailey. "Simultaneous localization and mapping: part I." IEEE robotics & automation magazine 13, no. 2 (2006): 99-110.

[18] Bailey, Tim, and Hugh Durrant-Whyte. "Simultaneous localization and mapping (SLAM): Part II." IEEE Robotics & Automation Magazine 13, no. 3 (2006): 108-117.

[19] Simmons, Reid, David Apfelbaum, Wolfram Burgard, Dieter Fox, Mark Moors, Sebastian Thrun, and Håkan Younes. "Coordination for multi-robot exploration and mapping." In AAAI/IAAI, pp. 852-858. 2000.

[20] Carlone, Luca, Miguel Kaouk Ng, Jingjing Du, Basilio Bona, and Marina Indri. "Simultaneous localization and mapping using rao-blackwellized particle filters in multi robot systems." Journal of Intelligent & Robotic Systems 63, no. 2 (2011): 283-307.

[21] Andersson, Lars AA, and Jonas Nygards. "C-SAM: Multi-robot SLAM using square root information smoothing." In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, pp. 2798-2805. IEEE, 2008.

[22] Thrun, Sebastian. "A probabilistic on-line mapping algorithm for teams of mobile robots." The International Journal of Robotics Research 20, no. 5 (2001): 335-363.

[23] Neira, José, and Juan D. Tardós. "Data association in stochastic mapping using the joint compatibility test." IEEE Transactions on robotics and automation 17, no. 6 (2001): 890-897.

[24] Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. MIT press, 2005.

[25] Alitappeh, Reza Javanmard, Guilherme AS Pereira, Arthur R. Araújo, and Luciano CA Pimenta. "Multi-robot deployment using topological maps." Journal of Intelligent & Robotic Systems 86, no. 3-4 (2017): 641-661.

[26] Saeedi, Sajad, Liam Paull, Michael Trentini, and Howard Li. "Occupancy grid map merging for multiple robot simultaneous localization and mapping." International Journal of Robotics and Automation 30, no. 2 (2015): 149-157.

[27] Jessup, James, Sidney N. Givigi, and Alain Beaulieu. "Robust and efficient multirobot 3-d mapping merging with octree-based occupancy grids." IEEE Systems Journal 11, no. 3 (2017): 1723-1732.

[28] Kia, Solmaz S., Stephen F. Rounds, and Sonia Martínez. "A centralized-equivalent decentralized implementation of extended Kalman filters for cooperative localization." In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, pp. 3761-3766. IEEE, 2014.

[29] Chen, Xieyuanli, Huimin Lu, Junhao Xiao, Hui Zhang, and Pan Wang. "Robust relocalization based on active loop closure for real-time monocular SLAM." In International Conference on Computer Vision Systems, pp. 131-143. Springer, Cham, 2017.

[30] Torres-González, Arturo, Jose Ramiro Martinez-de Dios, and Anibal Ollero. "Range-only SLAM for robot-sensor network cooperation." Autonomous Robots 42, no. 3 (2018): 649-663.

[31] Bowman, Sean L., Nikolay Atanasov, Kostas Daniilidis, and George J. Pappas. "Probabilistic data association for semantic slam." In Robotics and Automation (ICRA), 2017 IEEE International Conference on, pp. 1722-1729. IEEE, 2017.