Distributed Cooperative Mapping

Sandeep Konam (skonam), Vibhav Ganesh (vnganesh)

Abstract

In this project, we investigate the problem of computing the relative transformation between two quadcopters based on corresponding inter-robot observations developed during autonomous operation in a common unknown environment. Applications that rely on distributed cooperative mapping need to establish a common reference frame for integration of distributed observations and cooperative control decision. Generally they assume existence of a shared environment representation to establish the common reference frame, for which they require a robust strategy to establish relative pose between individual vehicles. Our work primarily involved testing out an EM based approach in simulation to establish the common reference frame.

Introduction

Most multi-robot applications such as coordinated control, surveillance, and rapid exploration that operate in unknown environments require distributed cooperative mapping as an important ability. Applications relying on distributed mapping must assume a shared environment representation in order to establish a common reference frame between vehicles. For this reason, two essential capabilities for a cooperative mapping system are the ability to robustly and efficiently compute data association, and establish relative pose transformations between individual vehicles in an online and distributed manner. We look at two main issues with the implementation of the previously described framework [15]. First we investigate how to obtain possible correspondences in a fast robust manner. Second we study how the data transfer between robots and how it scales to a larger team of robots.

Several solutions have been proposed in robotics community to address challenges encountered in distributed cooperative mapping. Proposed solutions can be broadly divided into Full-SLAM and Pose-SLAM. In multi-robot Full-SLAM approaches, robots have shared views of common landmarks [1], [2], [3], [4], [5]. Several of these Full-SLAM methods have the ability to build robust multi-robot data associations without a prior on initial relative poses [2], [5]. Specifically, Fox et al. consider a similar problem which was solved using a particle filter approach in the Full-SLAM framework [6]. Alternatively, Pose-SLAM approaches achieve greater efficiency and robustness in comparison to Full-SLAM by avoiding explicit estimation of landmark positions. EM-based approach that we adapted is built on Pose-SLAM approach. Many multi-robot Pose-SLAM techniques assume the existence of direct relative pose measurements between robots, and can compute a common reference frame with or without assuming prior information on initial relative poses between robots [7], [8]. However, the above mentioned strategies are built on strong proximal and temporal assumptions regarding interaction between quadcopters, which are inviable for operation in expansive environments. Our adapted approach focuses on missions through large and complex environments which do not permit frequent vehicle interaction, and assumes that vehicles autonomously explore the environment while opportunistically coordinating to build a common reference frame. Because of these constraints, we require robots to localize to one another solely through commonalities observed in collected and shared sensory observations. The problem of building a common reference frame has been solved under the assumption of perfect multi-robot data association [9], [10], [11], [12], but these works do not allow for erroneous data association. To address this issue, methods based on Expectation Maximization (EM) [13], [14] which simultaneously compute a common reference frame and infer data association through imperfect indirect measurements are preferred. Unfortunately, these approaches are unable to identify multi-robot data associations in the presence of drift as they assume perfect trajectory estimates.

In this project, we implement EM based approach discussed in [15], focusing towards a distributed, online, and real-time implementation of multi-robot cooperative mapping with unknown initial poses and indirect data association.

Problem Formulation

In the problem of multi-robot SLAM we seek to find the robot states χ^r for each of the R robots. Given the bayesian network that is commonly used to describe such a problem, finding the robot states becomes a problem of maximizing the likelihood.

$$\hat{\mathcal{X}}^{\mathcal{R}} = \operatorname*{argmax}_{\mathcal{X}^{\mathcal{R}}} p(\mathcal{X}^{\mathcal{R}} | \mathcal{Z}^{\mathcal{R}}),$$

Where Z^R corresponds the observations of each of the robots concatenated together and the multi robot data associations that are described in the next section.

Implementation of EM-based approach requires a fast 2D laser correspondence method to robustly detect correspondences between shared sensory observations. Multi-robot data association and loop closure problem in single robot Pose-SLAM formulation are similar in nature. Both of them involve comparison of a query scan to a set of cached scans. The distinction between the two is that the query scan in the former problem is received from a different robot. Therefore sensory correspondences are computed between quadcopters based on ideas from single robot 2D laser scan loop closure literature. Granstrom et al. [16] introduced a method that combines many global laser scan features, such as Centroid and Close Area features, as to determine loop closures. However, this method is unsuitable for a multi-robot localization task as it is not view-point invariant. Other works describe laser-based feature detection and description methods [17], [18] that can be used to rapidly detect correspondences. We have used FLIRT features [18] introduced by Tipaldi et al. due to their rotational invariance nature.

Steps involved in the approach followed:

1. Robots are initialized and begin localizing and building individual maps.
2. As they navigate, each robot shares locally acquired laser scans and SLAM pose estimates with other robots over a wireless network.
3. FLIRT features are extracted from laser scans received by each robot and are compared against that robot's local history of laser scan features.
4. RANSAC is used to find correspondences between the feature sets.
5. We then use EM to cluster correspondences and detect inliers amongst those clusters. The set of inlier correspondence clusters is used to build multiple transformation hypotheses for the robot pair.
6. The most probable hypothesis is used to initialize a pose graph transformation constraint between the robots, which is optimized over time using iSAM2 incremental optimizer [19].

Feature-based correspondence

Implementation of a fast 2D laser correspondence method precedes EM and hypothesis formulation. The 2D laser correspondence generation problem can be formulated in the following manner: given a set of cached laser scans $\mathcal{L}^{r} = \{\mathcal{L}^{r}_{i}\}$ from robot r and a query scan $\mathcal{L}^{r'}_{l}$ from robot r', determine the scans $\mathcal{L}^{r}_{k}$ in $\mathcal{L}^{r}$ which share similarity to $\mathcal{L}^{r'}_{i}$, and align them to produce a relative pose measurement $u^{r,r'}_{k,l}$.

Approach can be divided into detection and matching. In the detection step, given a local scan set $\mathcal{L}^{r}$ and a received scan $\mathcal{L}^{r'}_{i}$, we determine all correspondences (r, r', k, l). In the matching step, we align $\mathcal{L}^{r}_{k}$ with $\mathcal{L}^{r'}_{l}$ and return the relative pose measurement $u^{r,r'}_{k,l}$. As to reliably detect correspondences, it's crucial that robust and computationally efficient features need to be extracted. Local image features are used for a wide range of applications in computer vision and range imaging. While there is a great variety of detector-descriptor combinations for image data and 3D point clouds, there is no general method readily available for 2D range data. FLIRT (Fast Laser Interest Region Transform) [18] is widely adopted in Robotics community for this purpose as it combines the best detector with the best descriptor.

FLIRT

We used FLIRTLib which implements FLIRT (Fast Laser Interest Region Transform). The library implements the following detectors: range, normal edge, normal blob and curvature. Scale-space theory is applied to the range data by all the above mentioned detectors. Curvature based detector applies scale-space theory to the continuous geodesic coordinate of the curve where as range, normal edge and normal blob detectors apply scale space theory to the monodimensional signal defined by the laser scanner.

Range detector: It finds interest points in scale-space with a blob detector applied on the raw range information in the laser scan.
Normal Edge detector: It finds interest points in scale-space with an edge detector applied on a local approximation of the normal direction.
Normal Blob detector: It finds interest points in scale-space with a blob detector applied on a local approximation of the normal direction.
Curvature detector: It finds interest points using the scale space theory for curves introduced by Unnikrishnan and Hebert.

FLIRTLib also implements following Descriptors : Shape Context and Beta-Grid
Shape Context: It implements a local and linear version of the Shape Context introduced by Belongie and Malik.
*Beta-Gri*d: It implements a linear-polar occupancy grid. It extends the Shape Context with the notion of free space.

FLIRT combines curvature detector and beta-grid descriptor there-by forming the most powerful detector-descriptor pair. The rationale behind the curvature-based detector is that range data define a curve in Cartesian space and the scale space theory should be applied to this curve and not to the original signal. Curvature-based detector is chosen among other detectors such as range- based and normal-based detectors as it is invariant to different levels of subsampling and oversampling. The reason why the curvature-based detector is more invariant to different levels of subsampling and oversampling is due to the fact that it operates in geodesic coordinates and not on the raw range signal.

An important difference between image and range data is that the latter not only encodes metric distance information but also directed free-space information between the sensor (emitting light or sound) and the measured object. This is relevant extra information, not encoded in the shape context descriptor. Occupancy grids naturally deal with free-space information which is why we adopt this concept for the purpose of the second descriptor considered here. Concretely, for each detected interest point pdet we define a polar tessellation of the space around pdet . Again, this tessellation is linear in polar space, with a radius proportional to the scale of the interest point. For estimating the occupancy probability, we apply Bayesian parameter learning. This approach provides a sound way to initialize cell probabilities and delivers a variance estimation over the occupancy value.

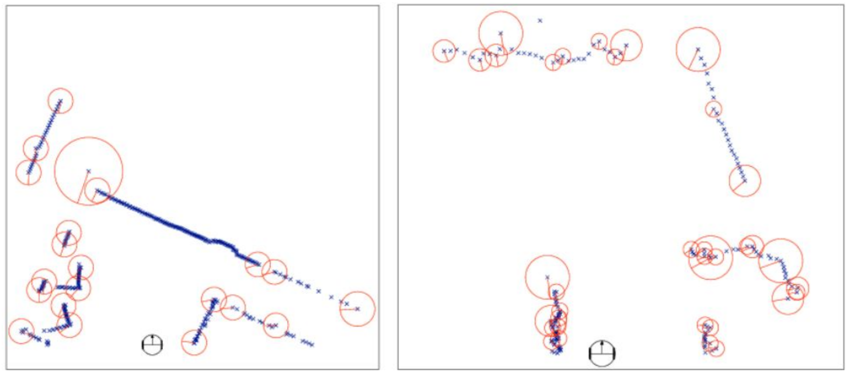


Figure 1. Example FLIRT detection result in indoor (mit-csail, left) and outdoor (fr-clinicum, right) data. The circles show the interest points and their support region (the actual descriptors are not shown).

Correspondence Detection

Correspondence detection includes training and testing phases. During training phase, FLIRT features and 128- dimensional descriptors are extracted for every incoming scan. We use FLANN library [20] to insert the generated descriptors into randomized kd-trees, providing capability to rapidly query for nearest neighbors in the 128-dimensional feature space.

During the testing phase, from the received query scan from another robot, we extract features and descriptors, then perform fast nearest neighbor searching in the local descriptor set for each query descriptor. We then place nearest neighbors of all query descriptors in a histogram, indexed by local scan number. We look for similarity between the two scans by observing the peak value in the histogram which corresponds to local scan with highest number of similar features. Figure 2 (a) shows an example query scan (blue) from another robot being matched against a set of 80 local scans. The scan at index 26 (red), corresponding to the histogram's peak, is indeed well-matched with the query scan.

Correspondence Matching

After detecting correspondences, a RANSAC-based matching strategy is used to find the relative pose measurement for each loop correspondence. We first match FLIRT descriptors in each matched scan pair (query and local matched scan candidate) by finding the nearest neighbour and then use RANSAC - 2 point in this case to reject outliers and build a rigid relative transform. Figure 2(b) depicts feature points plotted as circles, with nearest neighbor correspondences plotted as lines between features, and RANSAC inliers and outliers in black and gray, respectively.

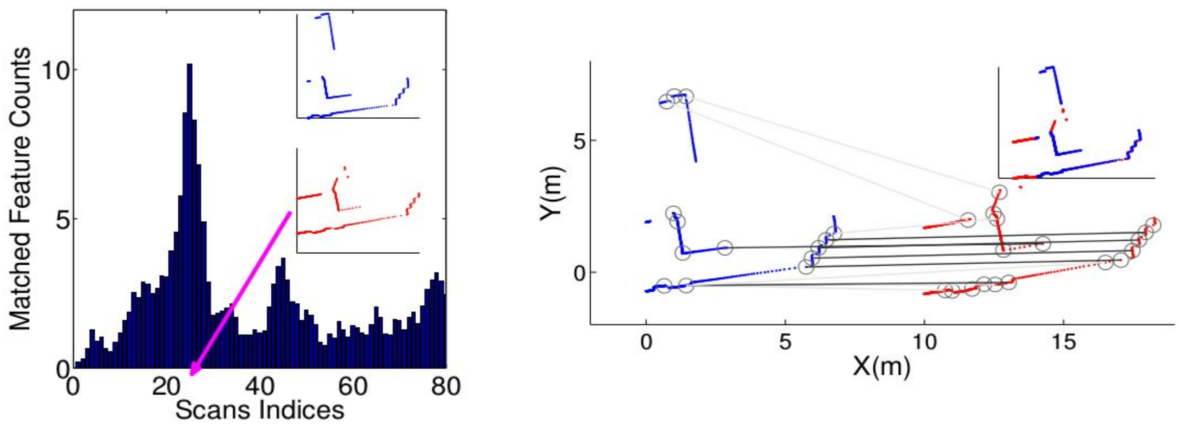


Figure 2. (a) Feature histogram (b) RANSAC between matched features

EM Formulation

With the potential correspondences that we obtain from scan matching, we run an expectation maximization (EM) algorithm to detect which of the scans are true correspondences (inliers) and which are not (outliers). We introduce binary latent variable J where each $j \in J$ corresponds to a multi robot data association that the previous process has provided the robot and

$$\hat{\mathcal{X}}^r = \underset{\mathcal{X}^r}{\operatorname{argmax}} \sum_{\mathcal{J}^r} p(\mathcal{X}^r, \mathcal{J}^r | \mathcal{Z}^r)$$

whether or not it is an inlier. The new problem therefore becomes

We can then formulate the EM as follows. The **E step** consists of calculating the lower bound on the likelihood of a possible relative pose T given odometry and observations marginalized over the latent variables J and creating a weighted average over them. We then attempt to find a relative transpose with the maximum likelihood given the parameters from the expectation step as part of the **M step.**

This setup works due to one key observation. Each data association between two scans leads to a relative transform. True data associations will lead to very similar relative poses whereas outlier associations will lead to scattered relative poses. We can therefore in effect cluster the poses that each data association gives and use the largest cluster us the true relative transform. However doing this leads to two problems. This clustering will not work well when there are very similar parts of the environment at different locations. Due to this **perceptual aliasing**, outlier relative transforms may cluster when they should not. In extreme cases, it will be impossible to combat perceptual aliasing. Furthermore, it is unclear how many clusters there will be and so running a clustering algorithm without having a preset number of clusters, such as mean shift, is intractable and does not work well in real time.

In order to solve these two issues, we use the proposed EM algorithm to create a robust relative transform. However, there are still some caveats to this approach. The EM algorithm still converges to a local optima and is therefore sensitive to initialization. In order to get good initial guesses, we utilize k-nearest neighbors with k = 5 to find the 5 best possible relative transforms. We feed these initial guesses into the algorithm and pick the one with the largest log likelihood. The result of the algorithm is a **hypothesis** which is a partition of the scans to either being inliers or outliers. Note that a partition leads directly to a relative transform since we can use the inlier scans to construct the transform. It is therefore sufficient for our algorithm to simply return the hypothesis**.**

In order to be robust to some perceptual aliasing, we throw away hypothesis that don't have high enough prior; in other words

$$p(j_s = inlier|u_s, \hat{T}^{(t)}, \hat{x}_k^r, \hat{x}_l^{r'}) < 0.8$$

Where s indexes over the scans.

Saliency

In the implementation of the code that we have seen, the rate of communication between a team of six robots is limited to 1Hz. This occurs as each scan is roughly 11Kb and the sending them at a high frequency is intractable for large teams of robots. We first include basic measures such as as time delays on the scans we send and not sending scans unless the robot has moved significantly. However for robots that are operating concurrently and moving within a map, these measures are of little help. Here we introduce the notion of scan saliency in order to determine to determine which scans to send. If we can find a numerical value for the importance of a scan, we can then setup the threshold value to saturate the network with only the most useful scans.

The measure of saliency that we used for our project relates to auto-covariance of a scan [21]. We first explore how to find the scan matching covariance matrix and then how we use it to obtain a numerical score we can use to threshold the data transfer.

Scan Matching Covariance and Measure

We estimate the covariance of a scan by performing the following steps.

1. Obtain N initial relative transformations.
2. For each of the N relative transformations, perturb the scan with relative transformation.

3. Use ICP to match the perturb scan with the original scan
4. Compare the relative transform that ICP obtains with the true relative transform.

Each of the N iterations will result in a Gaussian with the mean corresponding to the most likely relative transpose and the covariance representing the uncertainty measure. We can then combine the N Guassians into one by computing the first and second moments of the mixture:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i}^{N}\mathbf{x}_i$$

$$\mathbf{P} = \frac{1}{N}\sum_{i}^{N}\left(\mathbf{P}_i + (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T\right)$$

Here **P** is the scan matching covariance. We then use the value

$$\frac{1}{trace(P)}$$

to calculate the saliency score for the scan. We can then introduce a threshold to send only the scans with the highest scores.

Results

When implementing this framework, we were able to successfully have two robots find a relative transform between themselves in a simulated tunnel mapping scenario. This was done using a quadrotor and laser simulator written by Nathan Michael. We have one robot start of at location (10,0) and another robot at (0,2) and after 170 seconds of operation, they are able to find the relative transform although with a slight error (10.037, -2.076). The following figure shows the two robots and the joint map that they have made. Robot 1 is presented in red and robot 2 is in blue. The individual maps that the robots are also in the same color. The joint map shown is in the perspective of the first robot. Since we have developed a distributed framework, each robot is developing an independent joint map. The total run of this experiment was 300 seconds.

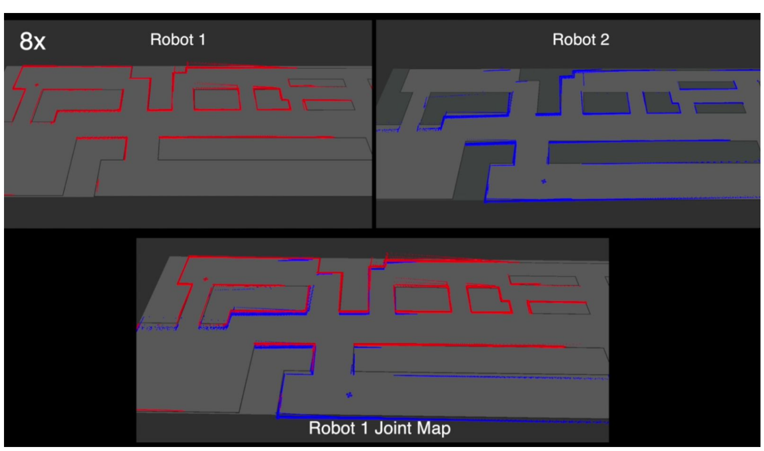
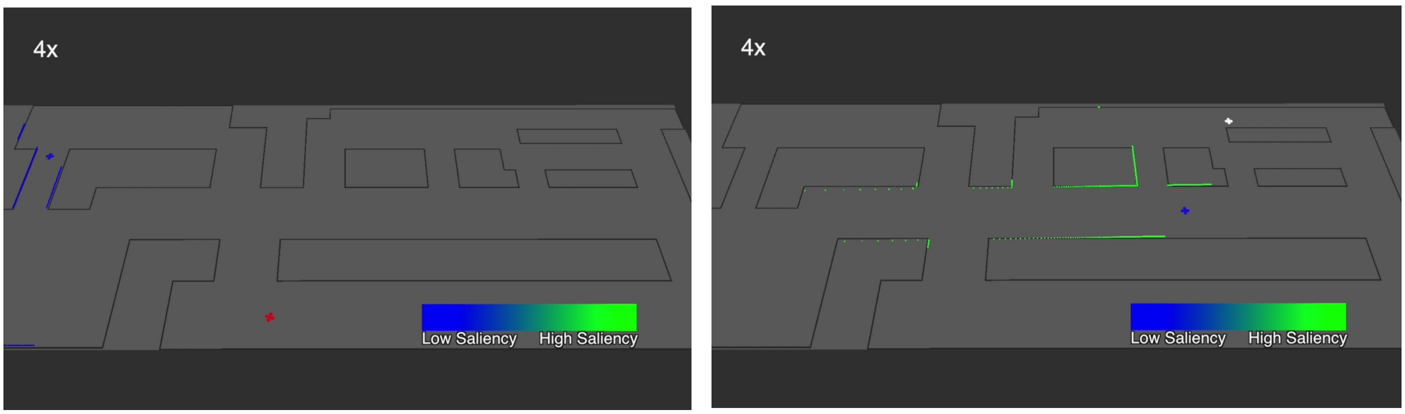


Figure 3. The top two local maps are merged in the lower map.

And we were able to visualize the saliency of the scans by running the experiment with a single robot and developing a color scheme for the measure. In figure 4(b) we see that there are lot of corners in the scan and so the scan is highly salient. However in figure 4(a), the scans only show walls and do not appear to be as salient



Figrue 4 (a) Low saliency (b) High saliency

As we did not put this ourselves on robots and perform the network communication, we were not able to test on a real system. However, in simulation, we monitored the network communication and found that it was below the bandwidth.

Future Work

The next steps of this work will be to test the system on a real robot and scale the system to more robots. As we scale the number of robots, we will need more strategies to deal with the network limitations. One possible solutions is to make the score for saliency better by incorporating information theoretic approaches. By looking at the local information gain from each scan, we might have a better idea of how relevant a scan is to send. Furthermore, scans are being transmitted and in cases of very high load, it might make sense to send the important features of the scans rather than the scans themselves.

References

[1] A. Howard, “Multi-robot simultaneous localization and mapping using particle filters,” Intl. J. of Robotics Research, vol. 25, no. 12, pp. 1243– 1256, 2006.

[2] L. Andersson and J. Nygards, “C-SAM : Multi-robot SLAM using square root information smoothing,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), 2008.

[3] K. Y. K. Leung, T. D. Barfoot, and H. H. T. Liu, “Decentralized cooperative slam for sparsely-communicating robot networks: A centralized-equivalent approach,” Journal of Intelligent & Robotic Systems, vol. 66, no. 3, pp. 321–342, 2012.

[4] I. V. Melnyk, J. A. Hesch, and S. I. Roumeliotis, “Cooperative vision-aided inertial navigation using overlapping views,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), pp. 936–943, 2012.

[5] A. Cunningham, K. Wurm, W. Burgard, and F. Dellaert, “Fully distributed scalable smoothing and mapping with robust multi-robot data association,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), (St. Paul, MN), 2012.

[6] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, “Distributed multi-robot exploration and mapping,” Proceedings of the IEEE - Special Issue on Multi-robot Systems, vol. 94, Jul 2006.

[7] X. Zhou and S. Roumeliotis, “Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case,” in IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), pp. 1785–1792, IEEE, 2006.

[8] L. Carlone, M. K. Ng, J. Du, B. Bona, and M. Indri, “Rao-Blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), pp. 243–249, 2010.

[9] L. Merino, J. Wiklund, F. Caballero, A. Moe, J. M. De Dios, P.-E. Forssen, K. Nordberg, and A. Ollero, “Vision-based multi-uav position estimation,” Robotics & Automation Magazine, vol. 13, no. 3, pp. 53– 62, 2006.

[10] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, “Multiple relative pose graphs for robust cooperative mapping,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), (Anchorage, Alaska), pp. 3185–3192, May 2010.

[11] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein, “Graph-based distributed cooperative navigation for a general multi-robot measurement model,” Intl. J. of Robotics Research, vol. 31, August 2012.

[12] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein, “Distributed vision-aided cooperative localization and navigation based on three-view geometry,” Robotics and Autonomous Systems, vol. 60, pp. 822–840, June 2012.

[13] V. Indelman, E. Nelson, N. Michael, and F. Dellaert, “Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), 2014.

[14] V. Indelman, N. Michael, and F. Dellaert, “Incremental distributed robust inference from arbitrary robot poses via em and model selection,” in RSS Workshop on Distributed Control and Estimation for Robotic Vehicle Networks, 2014.

[15] J.Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert, “Distributed Real-time Cooperative Localization and Mapping using an Uncertainty-Aware Expectation Maximization Approach,” in IEEE International Conference on Robotics and Automation (ICRA) 2015, At Washington, USA.

[16] K. Granstrom, J. Callmer, F. Ramos, and J. Nieto, “Learning to detect loop closure from range data,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), pp. 15–22, 2009.

[17] R. Zlot and M. Bosse, “Place recognition using keypoint similarities in 2d lidar maps,” in Experimental Robotics, pp. 363–372, Springer, 2009.

[18] G. D. Tipaldi and K. O. Arras, “Flirt-interest regions for 2d range data,” in IEEE Intl. Conf. on Robotics and Automation (ICRA), pp. 3616–3622, 2010.

[19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping using the Bayes tree,” Intl. J. of Robotics Research, vol. 31, pp. 217–236, Feb 2012.

[20] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in International Conference on Computer Vision Theory and Application VISSAPP), pp. 331–340, 2009.

[21] J. Nieto, T. Bailey, and E. Nebot, "Recursive scan-matching slam, " Jrnl. of Robotics and Autonomous Systems, vol. 55, no. 1, pp. 39-49, 2007.