Cooperative Multi Robot Path Planning using uncertainty Covariance as a feature
EECS-545 Final Report

Pratik Agarwal
Department of Computer Science
University of Michigan
Ann Arbor, MI
pratikag@umich.edu

Abstract

The project was aimed at applying some of the machine learning tools to the problem of multi-robot path planning. We have designed a Reinforcement Learning(RL) based multi agent planner, which maximizes the information gained as well keep the robots well localized. We have modified a 2D laser scan matcher to recover a multimodel distribution using the Expectation Maximization (EM) algorithm to model the uncertainty more accurately.

1 Introduction

In the last decade a significant amount of research in the robotics community has been performed addressing the problem of Simulataneous localization and Mapping commonly known as SLAM. We have reached a stage in SLAM algorithms in which we have been able to extend from a single agent to multi agent system as shown in the MAGIC competition [1]. A lot of existing literature on multi agent SLAM has humans driving the multiple robots so as to recover good maps. Autonomous exploration for multiple robots still remains an active field of research. Authors have approached this problem from a more engineering point of view, where robots are assigned exploration tasks based on heuristics. Methods such as maximizing information gain and assigning distinct frontier points [2] to a team of robots [3] to find optimal paths is most commonly used. These methods are very efficient at maximizing the information gained (non overlapping paths) but do not consider the uncertainty into factor. More recently planning in the belief space [4] has been performed for single robots, which plans paths with minimum uncertainty. Our approach can be considered as optimizing over both the goals of getting non overlapping paths(maximizing information) while keeping the uncertainty of each robot minimum(aid localization).

Coorperative localization is the technique of using multiple robots to aid localization. The approach used by most authors is the use of geometric constraints to come up with formations which the robots need to maintain [5], [6]. Cooperative localization has always been used seperately from standard planning algorithms. Our argument is that cooperative localization type planning should be only used when the environment has "perceptual aliasing" or the environment is not rich in features in other words the robot is susceptable to get lost. An intuitive way to think about the problem would be that we want to the planner to output plans for multiple robots such that they have minimum ovrerlap in "well localizable" areas and overlapping paths in "ambiguous" areas.

The central contribution of the project are

- RL based framework for multi-robot path planning maximizing the information gained and aid in localization

- use of EM to exactly recover the uncertainty covariance using Real-Time Correlative Scan Matcher [7]

The dual goal of the proposed multi-agent planner is a very desirable property for any team based multi robot system. To our knowledge no existing algorithm handles this problem in a fundamental way.

2 Proposed Method

We use an RL based framework to plan paths for all the robots one at a time using value iteration. We plan for one robot and then use the planned path while planning for other robots. This iterative fashion can be compared to a stochastic gradient method where the cost function converges after multiple iterations.

The rewards used were

- Goal reward for reaching the goal
- Localization reward, for being near a localizable feature
- negative reward for having paths overlapping with other robots
- positive reward if a robot helps localize another robot
- negative reward for being in each cell

The coefficients for each reward was designed by hand. Recovering the rewards in a more fundamental way using Inverse RL [8] or Optimal Reward Problem [9] is of future work.

The next obvious question would be what are good localizable features. We use Kanade-Tomassi corner detector on the Light Detection and Ranging(LIDAR) data as proposed by [10] as stable features. Going a step further we propose that even if an environment is feature rich, similar looking features create "perceptual aliasing" and we can identify such situation by using the Correlative Scan Matcher [7] framework, but using EM to model the distribution exactly, where each mode representing a possible data association. By doing this the uncertainty estimated is more principled as it represents each possible data association.

3 Prior Work

3.1 Reinforcement Learning

RL has been used in the field of robotics for planning and control extensively. RL algorithms have gauranteed convergence properties given the reward function. RL problems are generally posed as a Markov Decision Process (MDP), which provides the formalism for the same. An MDP tuple essentially consists of $(S, A, P_{sa}, \gamma, R)$, where S is a set of states, A a set of action, P_{sa} state transition probability,$\gamma \in [0, 1]$ is a discount and most importantly $R : S \times A$ is the reward function for taking action A in state S. The main benefit of using RL algorithms is the fact that we can specify rewards and recover the optimal policy. To calculate the optimal policy we can use either policy or value iteration algorithms. Policy iteration and value iteration algorithms [11] are Dynamic Programming Algorithms that compute optimal actions to take in a state given a perfect model of the world as a MDP.

More recently Inverse RL (IRL) [8] and Apprenticeship learning algorithms[12] have been successfully implemented on practical applications. In IRL we recover the rewards given the behaviour of an expert (optimal policy) on the problem. In apprenticeship learning there is no notion of maximizing the reward but instead we are trying to match the feature expectation of the expert (or optimal policy if available). These algorithms have been used for control and planning problems in robotics such as quadruped robot control, performing extreme aerial manuevers, outdoor navigation in a rough, parking Lot navigation and extreme parking manuevers.

3.2 Kanade-Tomassi features for localization

It was shown in [10] that stable features can be extracted from Light Detection and Ranging (LIDAR) data. Authors of [10] showed that multi-scale Kanade-Tomasi (KT) corner detector can be used of identifying highly stable and repeatable features at a variety of spatial scales without knowledge of environment, and produce principled uncertainty estimates and corner descriptors at same time. They show that the extracted features are stable for running SLAM algorithms on both indoor and outdoor datasets. Our idea is that if a robot is near one of these features then they are more likely to be well localized. The more the the number of features, the more the distinguishable the environment.

The KT corner detector defines corners as pixel patches whose self-similarity is sharply peaked. It simply checks the weighted sum of the square difference between an image patch $I(u, v)$ and the counterpart patch shifted by (x, y). After the approximation of the shifted patch with Taylor expansion, the problem is simplified as:

$$S(x, y) \approx (x \quad y)\, A \begin{pmatrix} x \\ y \end{pmatrix}$$

$$A = \sum_u \sum_v w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix} \tag{1}$$

Probabilistic planning algorithms such as [13], [4] rely heavily on sampling algorithms. Our results motivate the fact that these extracted features may use to bias the sampling process in addition to the proposed methods in [13], [4].

3.3 Correlative Scan Matcher

The Correlation based Scan-Matching [7] approach results in both a more robust maximum likelihood estimate and a principled estimation of uncertainty. The problem is formulated such that the robot is moving from x_{i-1} to x_i , according to some motion u. The observation z is dependent on the environment model m and the robots position. Our goal is to find the posterior distribution over the robots position, $p(x_i|x_{i-1}, u, m, z)$.

By applying Bayes rule and removing irrelevant conditionals, we get

$$p(x_i|x_{i-1}, u, m, z) \propto p(z|x_i, m)p(x_i|x_{x-1}, u) \tag{2}$$

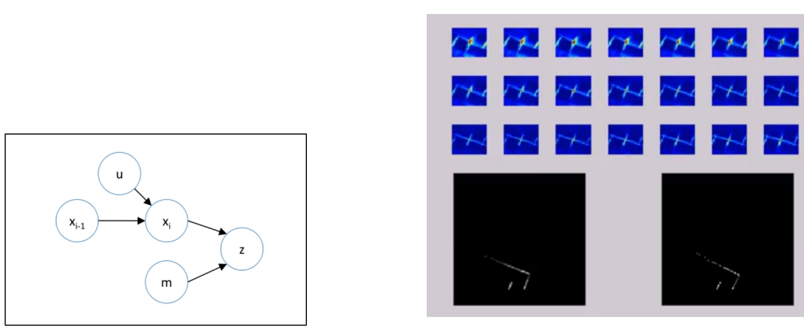

Figure 1: Graphical model for the probabilistic scan matching. x_{i-1} represents the previous position of the robot, u the motion of the robot, x_i the new robot position, m the world model, and z the laser scan observation. The right image shows the 3D cost function for matching LIDAR scans shown in black tiles. Each blue tiles shows the cost for all values of x, y for a particular value of θ.

The first term, $p(z|x_i, m)$ is the observation model: how likely is a particular observation, if the environment and the robots position are known? The second term, $p(x_i|x_{i-1}, u)$ is the motion model of the robot, as obtained (for example) from control inputs or odometry. Each individual lidar return z_j is independent giving Eq.3:

$$p(z|x_i, m) = \prod_j p(z_j|x_i, m) \tag{3}$$

In principle, we need to evaluate $p(z|x_i, m)$ over a three-dimensional volume of points; the three dimensions corresponding to the unknown parameters of the rigid body transformation, T : Δx,Δy, and $\Delta\theta$. Once the value of the cost function has been evaluated over a range of values of x_i , a multivariate Gaussian distribution can be fit to the data as per Eqn.4.

Let $x_i^{(j)}$ be j^{th} the evaluation of x_i :

$$\begin{aligned}
K &= \sum_j x_i^{(j)} x_i^{(j)^T} p(x_i^{(j)}|x_{i-1}, u, m, z) \\
u &= \sum_j x_i^{(j)} p(x_i^{(j)}|x_{i-1}, u, m, z) \\
s &= \sum_j p(x_i^{(j)}|x_{i-1}, u, m, z) \\
\sum_{x_i} &= \frac{1}{s}K - \frac{1}{s^2}uu^T
\end{aligned} \tag{4}$$

x_i is a 3-vector in x, y, θ and j is search space over x, y, θ. Later we show in the results section how we modified our approach to fit a multi-variate GMM to Eqn.4 using EM.

3.4 EM Algorithm

EM is an iterative optimization method to estimate some unknown parameters θ, given measurement data $X = (x_1, \ldots, x_N)$. Our job is to recover a family $\mathcal{F}$ of gaussian density functions, i.e. find the gaussians $f(x) \in \mathcal{F}$ that is most likely to have generated the given measurement data X.

$$f(x; \theta) = \sum_{k=1}^{K} p_k g(x; \mu_k, \Sigma_k) \tag{5}$$

where

$$g(x; \mu_k, \Sigma_k) = \frac{1}{|2\pi\Sigma_k|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\right]$$

and

$$\theta = (\theta_1, \ldots, \theta_k) = ((p_1, \mu_1, \Sigma_1), \ldots, (p_k, \mu_k, \Sigma_k))$$

and k is the number of mixtures, $\mu = \{\mu_1, \mu_2 \ldots \mu_k\}$ are the mean, and $\Sigma = \{\Sigma_1, \Sigma_2 \ldots \Sigma_k\}$ are the corresponding covariances for each gaussian. Here Σ_k is a full multi variate covariance matrix. In a hard clustering method such as k-means each sampled point belongs to only one cluster, but in a soft clustering method such as EM each point has a probability of belonging to each of the clusters, i.e. is the mixing probability. The likelihood function can be defiend as:

$$\Lambda(X; \theta) = \prod_{n=1}^{N} f(x_n; \theta)$$

and for the mixture of gaussians we have

$$\Lambda(X;\theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} p_k g(x; \mu_k, \Sigma_k) \tag{6}$$

where we need to compute

$$\hat{\theta} = \arg\max_{\theta} \Lambda(X;\theta)$$

The EM algorithm has 2 main steps - *E step* and *M step*. The *E step* tries to "guess" the values of the parameters and *M step* updates it. For more details on EM algorithms, refer [14],[15].

4 Results

First we show paths planned with beacons serving as localization points. The closer a robot is to a beacon the more it is assumed to be localized. Next we run our algorithm on some simulated maps using the April Robotics Toolkit [16]. We also show results of the implemented EM algorithm to recover the multi model uncertainty covariance on the scan matcher. All the code for the planner as well as for the EM estimation was written on Java.

4.1 Paths Recovered

For the first experiment (Fig.2) we have a grid world of 128 $\times$ 128 cells. Four robots begin at the bottom left and need to reach top right of the world. The agents action are non deterministic, such that it moves with a probility of 0.8 in the intended direction and a probability of 0.1 either left or right. Range beacons are randomly placed in the environment. The number of robots, start and destination location for each robot is given to the planner. The planner plans for one robot at a time. At each instance it uses the planned paths of the other robots. For each robot the planner gives the cost surface(gradient) of the map, shown in each tile of Fig.2. Once the cost is found we can simply choose the best action to take in each cell starting at the start position to reach the goal. We have used Manhattan distances for all computations. The last tile shows the planned paths for all robots. Visually the paths are non overlapping while keeing it near beacons as much as possible. It can also be seen that the planner doesnot try to over fit the localization or the exploration criteria.

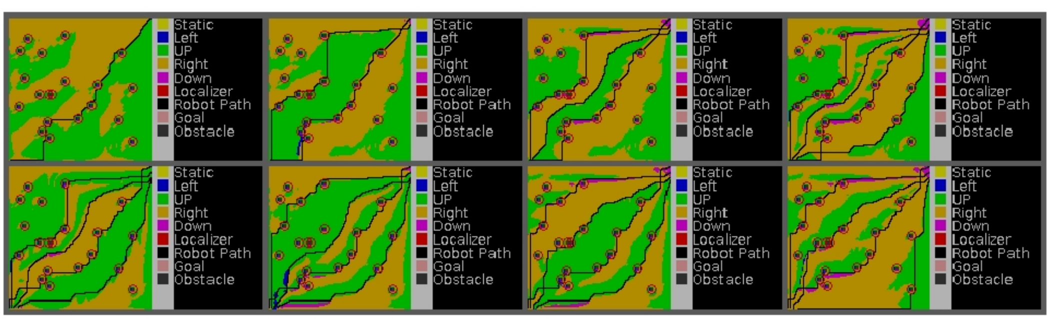

Figure 2: The figure shows the path planned for each robot at each iteration. The i^{th} column shows the evaluated plan for the i^{th} robot. The j^{th} row shows the result at the j^{th} iteration. The red circles are randomly placed range beacons. The start position for all the robots was bottom left and goal was to reach top right. The colors on the map show the optimal policy for each cell for each robot and paths are shown in black.

For the second experiment(Fig.3,Fig.4) we computed plans in a more realistic world with real world objects serving as features. Fig.3 shows a simple maze. To reach the goal the obvious choice would be for both the robots to take the two seperate paths. The first step is to use the KT corner detector to preprocess the map to find the features. These extracted features serve as good localization points and are shown by red circles in the images. To enhance computation we have 2D look up tables for the rewards. Seperate lookup tables were used for different rewards. The rewards for localization using KT features remains fixed in any of the iterations for all the robots, but the rewards to minimize overlap change with each evaluated robot. The second row in Fig.3 also shows the convergence of plans. Fig.4 shows another map and the planned paths for 4 robots. The plans do shuffle around before converging.

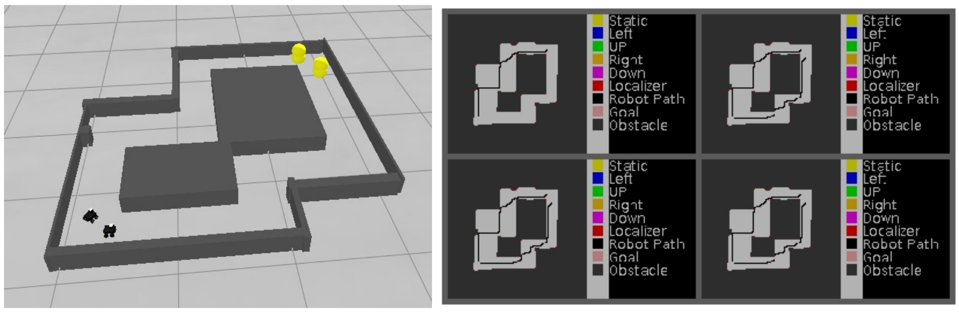

Figure 3: The image shows the path recovered for a simulated map for two robots. The gradient was not shown so that the paths can be seen clearly. The plans in the bottom row do not change showing that the planned paths converged and no more iterations are required.

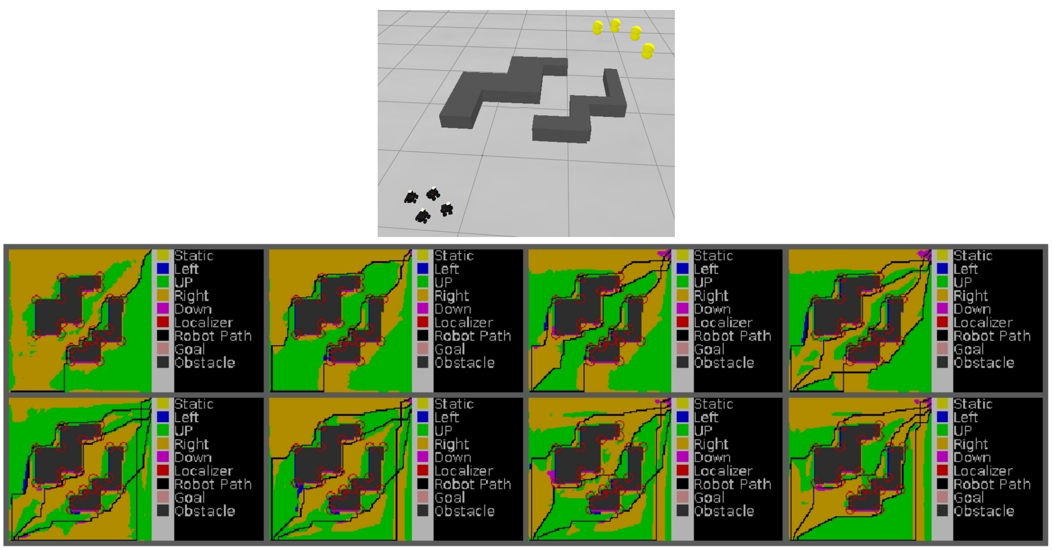

Figure 4: Paths recovered for a simulated world (top), with robots at the start point and goals shown by the yellow objects. The red circles are locations where the KT corner detector fired. The image (bottom) shows the optimal policies for each of the robots one at a time. The last map shows the optimal plan for all the robots.

Next we evaluate the situation where the robots help localize each other and show that they achieve the dual optimization (Fig.5). The image on top shown the plan which maximizes the information gained only, the 4 paths on the last tile of Fig.5(top) are at maximum seperation. The bottom image shows the planned paths with the world containing beacons and robots help each other to localize also.

Currently the way the algorithm is implemented it is slow for implementing on a real time system. It takes roughly $2s$ for each robot in a rasterized world of 128×128 cells. Each discreetized cell for the simulated world is considered to be $50cms$ (size of the robot).

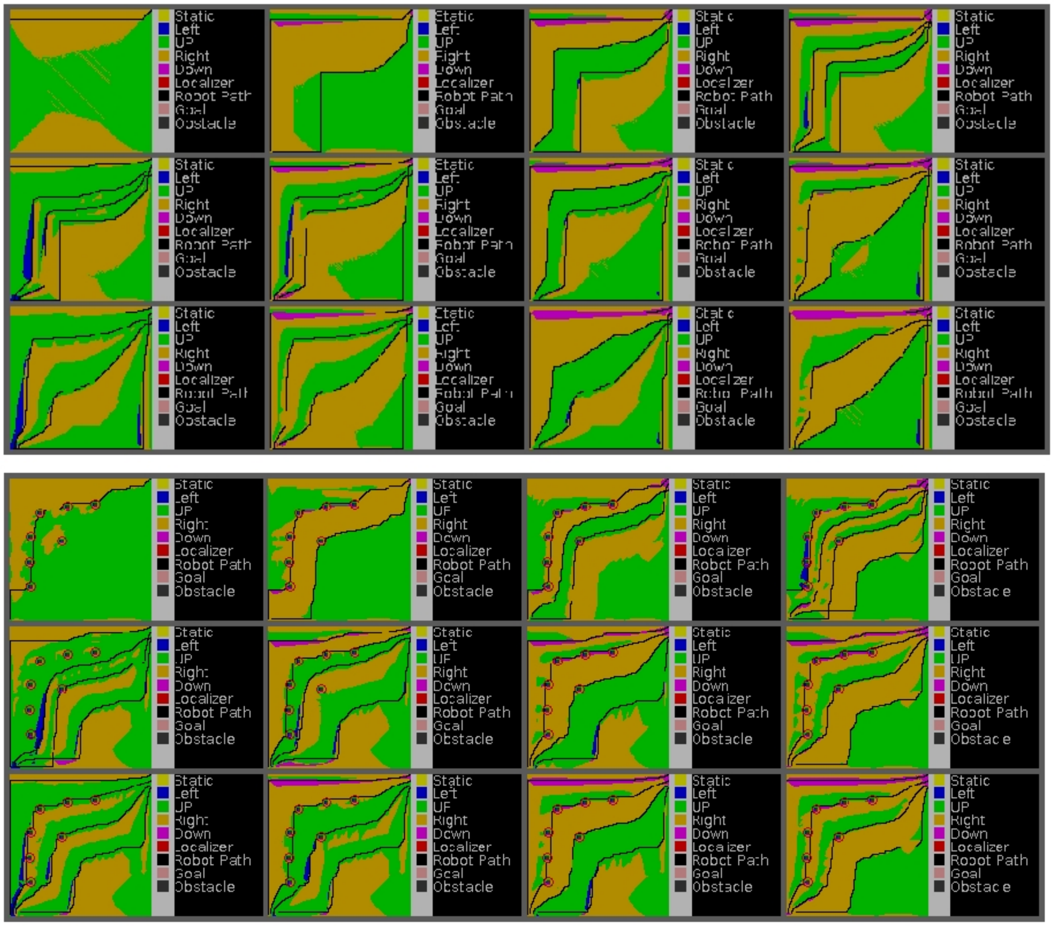

Figure 5: Here we compare the plans for maximizing the information with the proposed dual optimization. The top image shows the recovered plans with the goal of only maximizing information. We can see that all the paths are at maximum seperation. The distance are computed using Norm1, manhattan distances. The bottom image shows the paths for same start and goals for 4 robots given a few beacons. The first couple of robots are able to reach the goal while being very near the the beacons and well seperated, but with more robots this becomes a problem. The final tile shows the paths which are well localized and well seperated.

4.2 Estimated uncertainty as a Mixture of Gaussian

We now present the results to recover the multi modal data association using EM. In our case the fist step is to sample the (3D) cost Function where the values of the (3D) cost function gives $X = x_1, \ldots, x_N \in \mathcal{R}^3$. For us $\mu = \{\mu_1, \ldots, \mu_k\}$ represent each possible data association, and it is represented by a local maxima in $\mathcal{F}$. We performed a hill climbing algorithm in the 3D Cost Function after performing a hard clsutering to recover the possible data associations. These are essentially the peaks in $\mathcal{F}$. Finally the outcome of the EM also gives a prior on each gaussian representing the likehood of each gaussian, i.e. the probability of each match. In well constrained environments we will recover a single mode meaning that only 1 data association is possible. Care was taken not to over fit and model the noise itself. By hill climbing we also eliminated the problem of getting multiple estimates of the same μ_i. We also had a threshold to choose a gaussian for the multi modal distribution. If the probability for a particular gaussian was less than a chosen factor times the best match we rejected it, considering it to be noise.

Fig.6 shows a simulated world of the "perceptual aliasing" problem. It also shows the LIDAR data as seen by the robot. The environment is feature rich and will actually contain a lot of features, but not help in localization. Fig.7(top) shows the multi modal cost function recovered if we try to match scans for the above environment. In well localizable areas the distribution will have a single peak, but due to perceptual aliasing we have multiple peaks. The image shows the distribution in $\mathcal{R}^2$ but the distribution is actually in $\mathcal{R}^3$. The bottom images show the estimated covariance from the original implementation versus our GMM implementation. Each of the recovered gaussians in our EM implementation has a probability of being the correct match. In the original implementation of [7] the best estimate was chosen and a single covariance was computed for all data points.

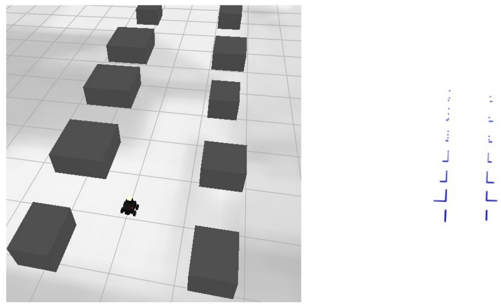

Figure 6: The left image shows a simulated world containing "perceptual aliasing". The right image shows the LIDAR data as observed by the robot.

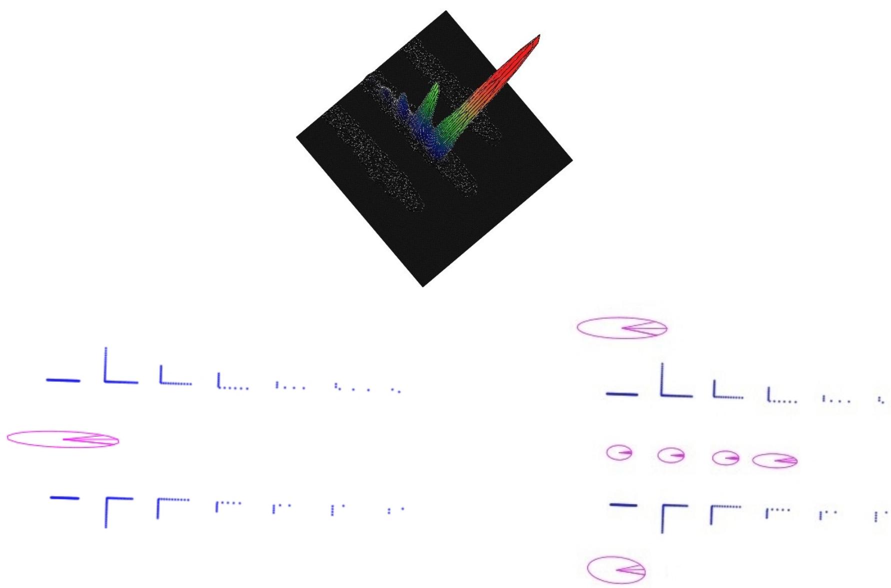

Figure 7: The top image shows the cost function of the matched scan. The "peaks" represent possible data association, while the "spread" of each peak represents the covariance for each. It shows the actual multi modal distribution in $\mathcal{R}^2$ - values for x, y for the best value of θ. The cost function is actually in $\mathcal{R}^3$ but for visual aid it shown for $\mathcal{R}^2$. Bottom left image shows the single covariance recovered using the original implementation. The bottom right image shows the estimated GMM using EM. The recovered covariances for both cases are in $\mathcal{R}^3$.

5 Conclusion

We proposed a framework for multi-robot planning using RL for the dual optimaization of maximizing information gained and aiding localization. We also showed results to extract multi modal distribution for multiple data association for the "perceptual aliasing" problem. Both these problems are individually subject to future work. We would like to make the proposed planner presented here into a much faster algorithm, so that in can be used in a real time system. The multi modal gaussian extraction in itself can be used to perform SLAM more effectively and more correctly.

References

[1] P. Ranganathan, R. Morton, A. Richardson, J. Strom, R. Goeddel, M. Bulic, and E. Olson, "Coordinating a team of robots for urban reconnaisance," in *Proceedings of the Land Warfare Conference (LWC)*, November 2010.

[2] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings of the IEEE International Symposium on Computational Intelligence (CIRA)*, 1997, p. 146.

[3] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2000, pp. 476 – 481.

[4] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *International Journal of Robotics Research*, vol. 8, no. 11-12, pp. 1448–1465, December 2009.

[5] S. Tully, G. Kantor, and H. Choset, "Leap-frog path design for multi-robot cooperative localization," in *Springer Tracts in Advanced Robotics*, vol. 62, 2010, pp. 307–317.

[6] I. M. Rekleitis, G. Dudek, and E. E. Milios, "Multi-robot collaboration for robust exploration." in *ICRA'00*, 2000, pp. 3164–3169.

[7] E. Olson, "Real-time correlative scan matching," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 4387–4393.

[8] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *in Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 663–670.

[9] J. Sorg, S. Singh, and R. Lewis., "Internal rewards mitigate agent boundedness," in *27th International Conference on Machine Learning*, 2010.

[10] Y. Li and E. Olson, "A general purpose feature extractor for light detection and ranging data," *Sensors*, vol. 10, no. 11, pp. 10 356–10 375, November 2010. [Online]. Available: http://www.mdpi.com/1424-8220/10/11/10356/

[11] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[12] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *In Proceedings of the Twenty-first International Conference on Machine Learning*. ACM Press, 2004.

[13] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, pp. 566 – 580, August 1996.

[14] C. Tomaski, *Estimating Gaussian Mixture Densities with EM - A tutorial*, Duke University.

[15] F. Delleart, "Expectation maximization algorithm," College of Computing, Georgia Institute of Technology, Tech. Rep. GIT-GVU-02-20.

[16] E. Olson, "April library," 2007. [Online]. Available: http://april.eecs.umich.edu/