Intrinsic Localization and Mapping with 2 Applications: Diffusion Mapping and Marco Polo Localization

Frank Dellaert, Fernando Alegre, and Eric Beowulf Martinson
College of Computing, Georgia Institute of Technology

Abstract

We investigate **Intrinsic Localization and Mapping (ILM)** for teams of mobile robots, a multi-robot variant of SLAM where the robots themselves are used as landmarks. We develop what is essentially a straightforward application of Bayesian estimation to the problem, and present two complimentary views on the associated optimization problem that provide insight into the problem and allows one to devise initialization strategies, indispensable in practice. We also provide a discussion of the degrees of freedom and ambiguities in the solution. Finally, we introduce two applications of ILM that bring out its potential: *Diffusion Mapping* and *Marco Polo localization*.

1 Introduction

In many mobile robot applications it is essential to obtain an accurate metric map of a previously unknown environment, and to be able to accurately localize the robot(s) within it. The process of reconstructing such a map from odometry and sensor measurements collected by one or more robots is known as Simultaneous Localization and Mapping (SLAM) [1]. Sensors that are commonly brought to bear on this task include cameras, sonar and laser range finders, radar, and GPS.

In this paper we investigate **Intrinsic Localization and Mapping (ILM)** for teams of mobile robots, a multi-robot variant of SLAM where the robots themselves are used as landmarks. This idea has been explored before, first by Kurazume [2, 3], and later by Rekleitis et. al. [4]. However, in both cases some robots are kept stationary while only a subset is allowed to move. We do not impose such a restriction here, although we do show that stationary "sentries" improve the global accuracy of the solution. Collaborative localization approaches with extrinsic landmarks were investigated, both Kalman-filter based [5], as sample based [6], using a multi-robot version of Monte Carlo Localization [7].

Our approach is more general than recent work at USC [8], as it handles both bearings-only and range-only scenarios, and does not require that the orientation of other robots can be measured. Both our approach and [8] are essentially straightforward applications of Bayesian estimation to the problem, and hence, though they were independently developed, are similar in many respects. However, in addition to this Bayesian framework, we present two complimentary views on the associated optimization problem that provide insight into the problem and allows one to devise initialization strategies, indispensable in practice. We also provide a discussion of the degrees of freedom and ambiguities in the solution.

Finally, we introduce two applications of ILM that bring out its potential: first, **Diffusion Mapping** is an approach where a highly redundant team of simple robots is used to map out a previously unknown environment, simply by virtue of recording the localization and line-of-sight traces, which provide a detailed picture of the navigable space. Second, **Marco Polo Localization** is a novel localization technique based on sound only, where robots measure range to each other by listening to a sound emitted by each robot in turn. The name is taken after a children's game that operates on a similar principle.

2 Problem Statement

The problem of intrinsic localization and mapping (ILM) is to estimate the *poses* X of all robots at all times given *odometry data* O and additional *intrinsic measurements* Z, i.e., which measure something about the relative pose between two robots. Below we discuss the scenarios where either only bearing or only range measurements to other robots are available, and a third scenario with any arbitrary combination of bearing and range measurements. It is not necessary to be able to measure the orientation of other robots, although this type of measurement is easily integrated in the framework if available. We will also assume that there is no correspondence problem, either because robots can be tracked in a recursive scheme, or because identity is available as part of the measurement.

Without loss of generality we consider a synchronous measurement scheme, where all the robots take a measurement at the same time. This assumption can be relaxed in a straightforward manner. In terms of notation, we will refer to the entire set of sought poses as X, whereas the poses of one robot only are denoted as X_i, with $i \in 1..m$, and the poses of all robots at a specific time t as X^t, with $t \in 1..T$. The pose of robot i at time t is denoted as x_i^t. Similar conventions are used for the odometry O and the intrinsic measurements Z. The set of intrinsic measurements between robots i and j is written as z_{ij}, and can be either empty, a bearing measurement, a range measurement, or both. We define $|e|^2_\Sigma \stackrel{\Delta}{=} e^T \Sigma^{-1} e$ to be the squared Mahalanobis distance with covariance matrix Σ.

Below we provide measurement models for odometry and the intrinsic measurements, respectively for a single robot and at a single time. The next section then considers the entire batch optimization problem over all robots and all times.

2.1 Odometry Measurements

Given no other information, the maximum a posteriori (MAP) trajectory $\hat{X}_i$ of a single robot i given odometry O_i is obtained simply by integrating the odometry over time. If no prior on the initial pose x_i^1 is available, the trajectory can be determined *up to a 2D displacement only*, i.e., an arbitrary translation and rotation in the plane. If a prior is available, there is no remaining ambiguity. In detail, the MAP trajectory is found by maximizing the posterior probability

$$P(X_i|O_i) \propto P(X_i)P(O_i|X_i) = P(x_i^1)\prod_t P(o_i^t|x_i^t, x_i^{t+1}) \qquad (1)$$

where we make the usual conditional independence assumptions, and the only prior knowledge available is a guess $\overline{x}_i^1$ for the initial pose x_i^1. In the case of normally distributed measurement noise, the associated error to be minimized is equal (up to a constant) to the negative log-posterior, given by

$$E_{oi} \stackrel{\Delta}{=} |x_i^1 - \overline{x}_i^1|_Q^2 + \sum_t |o_i^t - g(x_i^t, x_i^{t+1})|_R^2 \qquad (2)$$

where $g(x,y)$ is the odometry measurement function between two poses x and y, and Q and R are the covariances for the prior on x_i^1 and odometry measurements o_i^t, respectively.

2.2 Intrinsic Measurements

Similarly, assuming no prior for now, at each time-step t we can obtain a maximum likelihood estimate $\hat{X}^t$ for the *configuration* of poses X^t given only the intrinsic measurements Z^t at time t, by maximizing

$$P(Z^t|X^t) = \prod_{ij} P(z_{ij}|x_i^t, x_j^t) \qquad (3)$$

or, alternatively, minimizing the associated error:

$$E_z^t = \sum_{ij} |z_{ij} - h(x_i^t, x_j^t)|_S^2 \qquad (4)$$

where $h(x,y)$ is the intrinsic measurement function associated with the ordered pair of poses x and y, S is the noise covariance for each set of measurements, and the summation is over all pairs (i, j) where a bearing and/or range measurement is available. To determine whether the configuration X^t can be determined at all, we need to count the degrees of freedom (DOF). The number of unknown parameters is $3m$, i.e., (x, y, θ) for each robot, and a solution can be obtained only if the number of measurements actually available is more than the DOF. We distinguish three different cases:

- *Bearing measurements only*: the configuration can be determined up to a 2D similarity transformation. Since a similarity has 4 DOF, the system has $3m-4$ DOF, and the maximum number of measurements is $m(m-1)$. Recovering the configuration (up to the stated ambiguity) is possible when at least 3 robots are available, as discussed in [9]. The latter paper also provides a linear method for obtaining an initial estimate. When $m \leq 3$, it is possible that a second flipped solution exists as well [9].

- *Range measurements only*: the orientation of the robots is not observable, and robot positions (2 unknowns per robot) can only be determined up to a 2D displacement and an orientation flip. Hence, the DOF are $2m-3$ whereas $K_{max} = \binom{m}{2}$, possibly yielding a solution (up to the stated ambiguity) when at least 4 robots are available.

- *Mixed measurements*: in general, the configuration can be determined up to a 2D displacement with a minimum of 2 robots.

3 Intrinsic Localization and Mapping

The ILM problem as stated above can now be seen as combining these two estimation problems, i.e.. obtaining the MAP estimate for the poses X for all robots $1..m$ and times $1..T$, given the odometry O and the intrinsic measurements Z:

$$\hat{X} = \underset{X}{\operatorname{argmax}}\; P(X|O,Z) = \underset{X}{\operatorname{argmax}}\; P(X)P(O|X)P(Z|X)$$

or, alternatively, minimizing the following error function:

$$E \stackrel{\Delta}{=} \sum_i E_{oi} + \sum_t E_z^t =$$
$$\sum_i |x_i^1 - \overline{x}_i^1|_Q^2 + \sum_{it} |o_i^t - g(x_i^t, x_i^{t+1})|_R^2 + \sum_{tij} |z_{ij} - h(x_i^t, x_j^t)|_S^2$$

We use a non-linear optimization method, Levenberg-Marquardt with a sparse QR solver, to obtain the MAP estimate in a batch optimization procedure. For on-line applications, it is straightforward to use this method as a subroutine in a fixed-lag smoothing scheme, where one only optimizes for the last n time slices while keeping the other poses constant. To compute the (sparse) Jacobian $\frac{\partial E}{\partial X}$ we have implemented an automatic differentiation (AD) framework. AD is neither symbolic nor numerical differentiation, and calculates the Jacobian at any given value exactly, efficiently, and free of numerical instabilities. See [10] for more details.

4 Of Tracks, Slices, and Sentries

In practice non-linear optimization is plagued by local minima, and insight in the structure of the problem is needed to provide a good initial estimate to the solver. This is especially so if no prior $P(X^1)$ for the initial poses is available, in which case the problem of local minima is more severe. Below we present two complimentary views of the problem that enable us to tackle the initialization problem in a two-step approach.

4.1 Tracks View

The first view is to decompose the problem into m robot localization problems, where the individual *tracks* X_i of the robots are related only through the intrinsic measurements Z:

$$P(X|O,Z) \propto P(Z|X)\prod_i \left[P(x_i^1)P(O_i|X_i)\right]$$

If a prior $P(X^1) = \prod_i P(x_i^1)$ is available, then the individual MAP tracks are determined exactly (Section 2.1), and can be seen as *corrected* by additional measurements Z between the tracks, through $P(Z|X)$. Conversely, if $P(X^1)$ is not available, then the tracks are only determined up to a 2D displacement, and the intrinsic measurements Z have the additional effect of *registering* the tracks to each other in the plane. A strategy to avoid local minima in the latter case is then the following:

1. Create the m tracks X_i by integrating the odometry O_i.

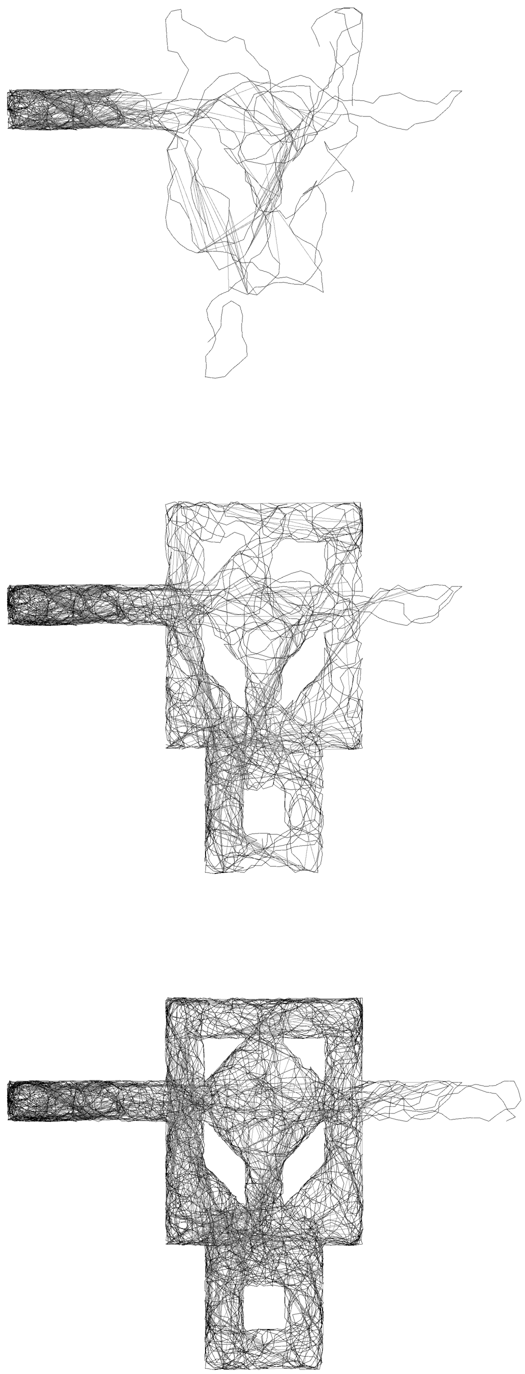

Figure 1: Diffusion Mapping: a simulated example where 15 robots are released on the left and execute a pure random walk control strategy in a large environment, except that they reflect off walls. Shown are the traced trajectories at regular time intervals between 0 and 1000 steps, which collectively constitute a map of the empty space, and hence of the navigable environment. Gray lines indicate recorded lines of sight, which complement the trajectory information.

2. Solve the $3m$ dimensional optimization problem of registering the tracks by maximizing $P(Z|X^1)$, while keeping the tracks rigid.

3. Use that as the starting point for the global optimization.

There is still no guarantee that step 2 will not get stuck in a local minimum. However, since the related optimization problem is relatively small, it can be restarted several times at a low computational overhead.

4.2 Slice View

The second view is to decompose the problem into T individual time slices X^t, with each *slice* X^t coupled through the *aggregate odometry* O^t at time t to the next slice X^{t+1}:

$$P(X|O,Z) \propto \left[P(X^1) \prod_t P(Z^t|X^t) \right] \prod_t P(O^t|X^t, X^{t+1})$$

In case enough measurements Z^t are available at time t, a maximum likelihood (ML) estimate for the slice configuration can be obtained, up to a 2D displacement and possibly up to a scale and orientation flip (Section 2.2). The corresponding staged optimization strategy is:

1. Find the ML estimate for each slice X^t, *if possible*.

2. Solve the $3T$-dimensional (or 4T, if range-only) problem of the slice poses by maximizing $\prod_t P(O^t|X^t, X^{t+1})$, while keeping the slices rigid.

3. Use that as the starting point for the global optimization.

The advantage of the optimization problem in step 2 is that the Hessian is bandwidth limited, and hence can be solved in linear time. However, in the range case there is the additional problem that the orientation of each slice needs to be determined. The disadvantage of this approach is that step 1 might not be possible, if not enough measurements are available in some slices.

4.3 Drift and Sentries

Note that neither of the strategies outlined above is a recipe for success: it might still be the case that the global optimization process in step 3 gets stuck in a local minimum. The slice view also provides another insight, which is that the global optimization problem is very similar to the single robot localization process, when one views the slices as an evolving articulated robot system. That makes it clear that, much like the single robot case, error will accumulate over time and the entire system will drift away from the ground truth situation.

A possible solution to the drift problem is to use sentry robots that remain stationary over time, and provide a series of landmarks that can help “close the loop”, when part of the team wanders back into a previously visited area. The number of ways in which this can be done is endless and will depend on the application. One way is to leave a “coordinate frame” team of two or three robots at the starting point of the robot-team. Another approach, possible with highly redundant robot swarms, is to occasionally “drop” sentries along the trajectory, according to some distance or line of sight criterion.

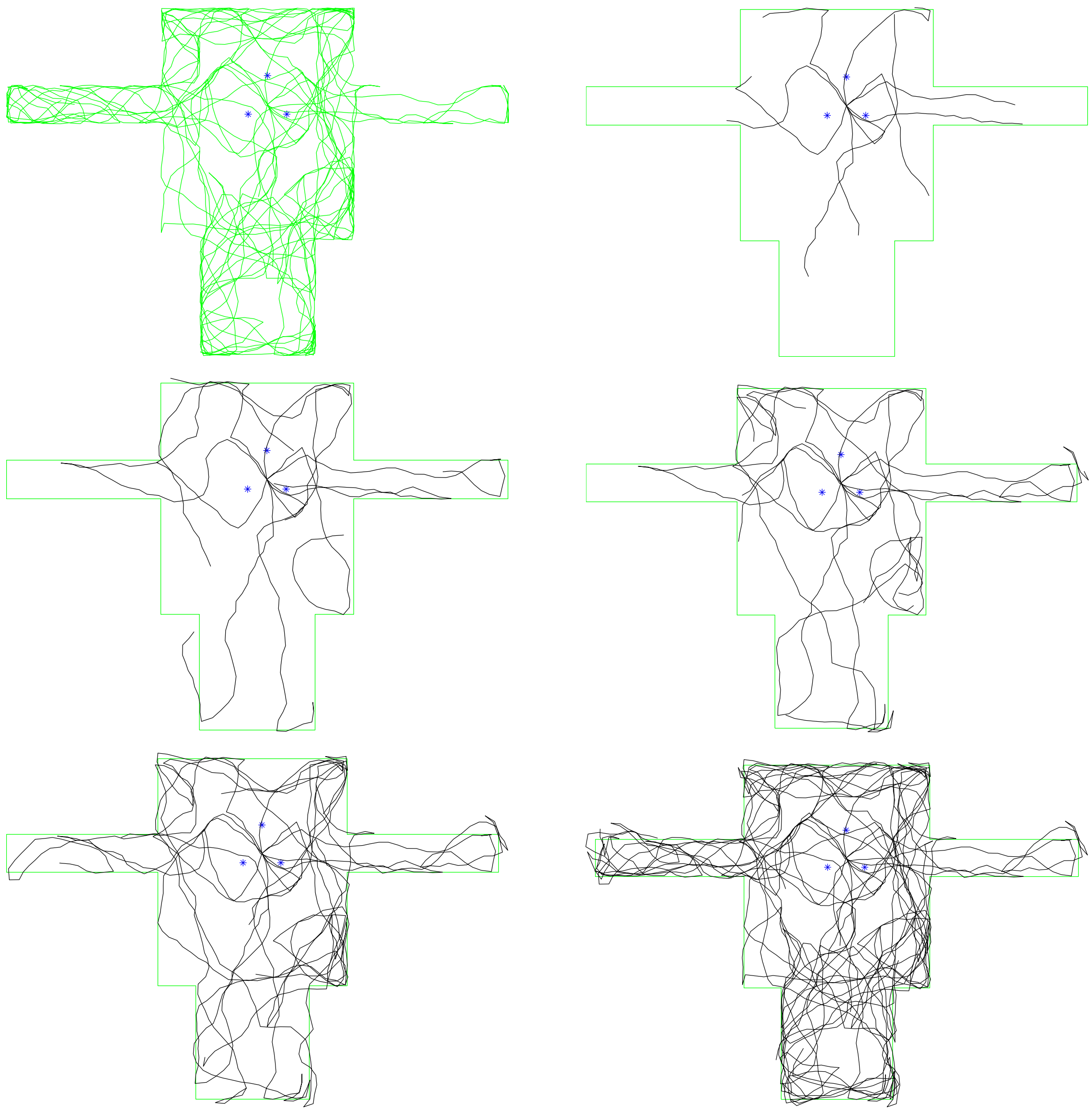

Figure 2: Simulation with 10 robots (and 3 sentries) over 200 timesteps. The first panel shows the actual simulated tracks. The simulated environment outline (modeled after an existing museum building) is shown in gray in all other panels, which show the MAP estimate $\hat{X}$ at different time steps, respectively 20, 40, 60, 100, and 200.

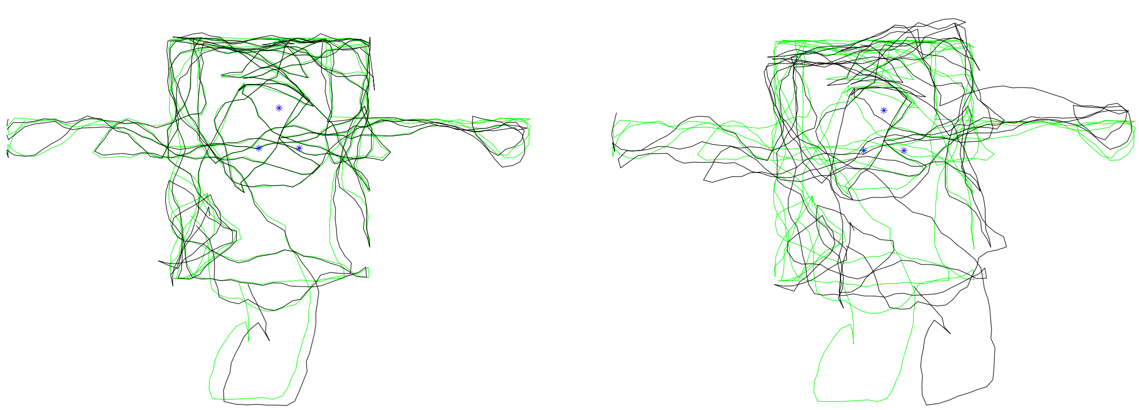

Figure 3: Simulation that illustrates the global correction by sentry robots. The ground truth tracks of 5 robots running for 100 timesteps are shown in gray. The 3 stars in the middle are sentries that establish a coordinate frame. Left: the estimated tracks using the measurements from the sentries. Bottom: the estimated tracks without using the sentries.

5 Application I: Diffusion Mapping

Mapping navigable space is important for mobile robots and can also be a product in its own right, e.g., in the case of reconnaissance. In this paper we introduce one way of tackling this problem, *diffusion mapping*, which is based on the ILM framework discussed above and is illustrated in Figure 1. The idea is to use teams of many small robots as a dynamically deploying, richly connected net of sensors, in order to map out navigable space. We call this diffusion mapping, as very simple random walk or diffusion control strategies are used for the individual team members.

Diffusion mapping is a straightforward implementation of the ILM framework for the case of bearing-only measurements. We maintain the global position and configuration of the evolving sensor net, using robot odometry and relative bearing measurements between the robots themselves. The traces of the robots through space as well as lines of sight between the robots at successive times are used to carve out the free space.

We are currently building a large number of small robots in the BORG lab at Georgia Tech (http://borg.cc.gatech.edu) in order to, among other goals, validate diffusion mapping experimentally. However, the system already runs in simulation and will be used to test the algorithms and help determine the accuracy specifications of the bearing sensors that we will put on the robots. Under consideration are both IR and RF-based bearing and range sensors.

Figure 2 shows the results of a simulation with 13 robots, of which three were used as sentry robots as discussed in Section 4.3. The simulation was run for 200 time-steps, and the figure shows the ground truth tracks of the robots, as well as the MAP estimate $\hat{X}$ at different times in the simulation, superimposed on the ground truth. In this case we used simulated odometry with a positional accuracy of 10cm, orientation accuracy of 5 degrees, and bearing sensors with an accuracy of 5 degrees. All the accuracy figures correspond to the standard deviation of normally distributed noise that was generated during the simulation of the measurement values. In the last timestep, the optimization is over 7800 unknowns, using information from 5683 bearing measurements and 7761 odometry measurements. Using the sparse QR solver, each Levenberg-Marquardt iteration for that size takes a couple of seconds.

Figure 3 more clearly illustrates the beneficial effect of dedicating some of the robots as sentry robots, in order to establish a global frame of reference.

6 Application II: Marco Polo Localization

In Marco Polo Localization, we apply sound as a tool for gathering the range measurements between robots, and solve those as a range-only Simultaneous Localization and Mapping (SLAM) [1] problem. By applying sound as a sensor for gathering range estimates, we hope to demonstrate the feasibility of using sound for localization and to increase the generality of the SLAM problem. Marco Polo Localization is described in more detail in [11], but below we present some experimental results to illustrate the approach.

Figure 4: Robots used to test Marco Polo Localization.

The scenario is as follows: a group of robots are scattered about the environment in unknown starting positions. They each generate a sound, and time differences are gathered between each of the robots, allowing us to obtain range estimates. This is called a slice, as it reveals the position of the robots at one point in time. After sounding off, the robots move about the room, recording their change in odometry. At some time later, the robots are stopped, and another slice is gathered. This procedure of moving, stopping, and sounding off is repeated regularly throughout the experiment.

Experiments were run on four Nomad 150 robots, see Figure 4, equipped with laptops and a wireless connection. Each Nomad had a speaker mounted on one side of the laptop, and a microphone mounted on the other. In order to synchronize the recordings between channels, microphones were plugged into a single desktop computer with a 16bit sound card. The digital sampling quality of all recording was performed at 16bit quality, and 22050 Hz. While the microphones used were wired, the system can in principle be implemented wirelessly without difficulty.

Distance estimates were gathered from the robots in pairs. One robot played a sound while recording data, while a second robot was also recording data. These two files were then saved. Then the robot started recording again and repeated the sound while another robot was listening. This process repeated until every pair of robots had recorded a time-delay between them. We were limited to two robots at a time, because the standard sound we used was limited to recording two channels at a time. In principle, one sound could be used to generate $n-1$ readings using the appropriate hardware to synchronize n channels.

Once all the data was recorded, each pair of sound files was

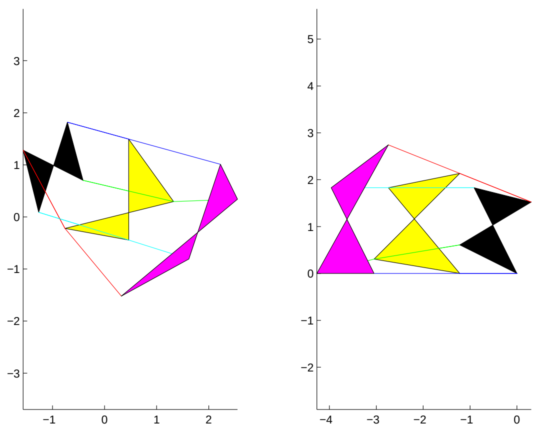

Figure 5: (left) Real Robot results for 4 robots, and 3 slices. (right) True positions. In this experimental run, 14 range measurements were used to align the tracks.

compared using a cross-correlation algorithm to find the time-delay between the two channels. The sound used was a clicking noise that was experimentally determined to provide the best time-delay estimates using cross-correlation. The software used for the actual cross-correlation was Ishmael 1.0, developed by the Office of Naval Research [12]. If the cross-correlation algorithm returned an estimated time-delay greater than 15ms, then that measurement was discarded. This threshold value was experimentally determined to be the physical limit of the microphones/amplifier used.

In practice, a large number of the actual time-delay measurements will not be available to help align the tracks. Especially as robots start to move around in the environment, it occurs more often that they are located in positions where a good time-delay estimate is difficult to obtain. On the experiments with 4 robots and 3 stops, three tests recorded 14, 10, and 10 useful time-delay estimates out of a possible 18. The rest were removed by thresholding. Actual bad measurements which were not removed by thresholding were rare, and did not influence the data much. The error here is mostly due to the off centered position of the microphones on each robot. Microphones could not be placed exactly center because of existing equipment on the robots.

Figure 5 displays the reconstructed results from test 1, using the track view method. Two changes to procedure would help correct for this error. First, the microphones should be located as close to the center as possible, or their position on the robot needs to be incorporated into the model. Second, the more range estimates obtained, the better.

7 Conclusion

The Intrinsic Mapping and Localization framework we propose can be efficient, as demonstrated using the large scale Diffusion Mapping simulations. It has been validated in practice in the context of Marco Polo Localization. And it can be implemented easily, using automatic differentiation to take care of the most tedious aspect of optimization. We have shown results for both bearing only and range-only scenarios, and the method handles any combination thereof. In future work we will attempt to validate the concept of diffusion mapping in practice using a highly redundant team of small mobile robots, using between 15 and 100 robots. We are also planning to further explore the use of sound localization, using both an intrinsic measurement scheme as discussed here, as well as extrinsic sound measurements from the environment.

References

[1] J. Leonard and H. Durrant-Whyte, “Simultaneous map building and localization for an autonomous mobile robot,” in *IEEE Int. Workshop on Intelligent Robots and Systems*, pp. 1442–1447, 1991.

[2] R. Kurazume, S. Nagata, and S. Hirose, “Cooperative positioning with multiple robots,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 2, pp. 1250–1257, 1994.

[3] R. Kurazume and S. Hirose, “An experimental study of a cooperative positioning system,” *Autonomous Robots*, vol. 8, pp. 43–52, January 2000.

[4] I. M. Rekleitis, G. Dudek, and E. Milios, “Multi-robot collaboration for robust exploration,” *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 7–40, 2001.

[5] S. Roumeliotis and G. Bekey, “Distributed multi-robot localization,” *IEEE Trans. on Robotics and Automation*, August 2002. To Appear.

[6] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, “A probabilistic approach to collaborative multi-robot localization,” *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, 2000.

[7] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte Carlo Localization for mobile robots,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1999.

[8] A. Howard, M. J. Mataric, and G. S. Sukhatme, “Localization for mobile robot teams using maximum likelihood estimation,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, September 2002.

[9] F. Dellaert and A. Stroupe, “Linear 2D localization and mapping for single and multiple robots,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, May 2002.

[10] A. Griewank, “On Automatic Differentiation,” in *Mathematical Programming: Recent Developments and Applications* (M. Iri and K. Tanabe, eds.), pp. 83–108, Kluwer Academic Publishers, 1989.

[11] E. Martinson and F. Dellaert, “Marco Polo Localization,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2003.

[12] D. Mellinger, “Ishmael 1.0 user’s guide. ishmael: Integrated system for holistic multi-channel acoustic exploration and localization,” Tech. Rep. OAR PMEL-120, NOAA Technical Memorandum, 2002.