Global A-Optimal Robot Exploration in SLAM

Robert Sim
Department of Computer Science
University of British Columbia
2366 Main Mall
Vancouver, BC V6T 1Z4
simra@cs.ubc.ca

Nicholas Roy
Computer Science and Artificial Intelligence Lab
The Stata Center, MIT
32 Vassar Street, 32-335
Cambridge, MA 02139
nickroy@mit.edu

***Abstract*— It is well-known that the Kalman filter for simultaneous localization and mapping (SLAM) converges to a fully correlated map in the limit of infinite time and data [1]. However, the rate of convergence of the map has a strong dependence on the order of the observations. We show that conventional exploration algorithms for collecting map data are sub-optimal in both the objective function and choice of optimization procedure. We show that optimizing the *a-optimal* information measure results in a more accurate map than existing approaches, using a greedy, closed-loop strategy. Secondly, we demonstrate that by restricting the planning to an appropriate policy class, we can tractably find non-greedy, global planning trajectories that produce more accurate maps, explicitly planning to close loops even in open-loop scenarios.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is the problem of how to build environmental models or maps from sensor data collected from a moving robot. SLAM is considered to be one of the cornerstones of autonomous mobile robot navigation [2], but is technically challenging because the robot position and the world features must be estimated simultaneously from noisy sensor data. Recent research has resulted in substantial progress in autonomous map-building; there are now a number of systems that can reliably build many kinds of environmental models [1], [2], [3], [4].

There have also been a number of recent attempts to automate the exploration policy for gathering data to be assembled into a map; these algorithms have in particular focused on maximizing expected information gain [5], [6], [7] whenever possible. However, existing approaches are limited in two senses: the exploration trajectory is typically based on purely local optimization, and the traditional measure of information gain does not lead to the most accurate maps.

In this paper, we describe a motion planning algorithm for SLAM that computes the multi-step trajectory that maximally reduces the error of the map. We use the notion of mean uncertainty, or *a-optimality* [8], as our objective function. We then describe a breadth-first search algorithm for computing global exploration trajectories. The information gain is computed not for a given point, but by *integrating* observations along the trajectory to the point. We can prune the search space of trajectories aggressively by restricting ourselves to the class of trajectories that do not contain cycles, leading to a computationally tractable search for a sequence of actions. This results in one of the first algorithms that allows exploration for exploiting global information along trajectories, rather than exploiting local information gain.

It must be emphasized that our use of the *a-optimal* objective function does not necessarily lead to an optimal exploration trajectory as we do not exhaustively search the space of all trajectories. The goal of this paper is to demonstrate that the *a-optimal*, global planning procedure leads to measurably more accurate maps than conventional information measures with local policies. In particular, we show that the *a-optimal* measure generates more accurate maps, even when following a greedy, local search strategy. In addition, we show that our global planning strategy explicitly plans to close loops when necessary, even in open-loop planning scenarios, resulting in faster convergence to the correct map.

II. SLAM

We assume that we have a quasi-holonomic robot operating in a planar environment. Without loss of generality, we assume the robot is equipped with an idealized sensor that can sense the range and bearing of any number of features in the plane[1]. The robot issues controls at each time t, which describe relative translational and rotational displacement of the robot: $U_t = (\Delta d, \Delta\theta)$. The sensor measurements are a sequence of range and bearing measurements: $z_t = (r_1, b_1, r_2, b_2, \ldots r_n, b_n)$ for n features of the environment. The full state space ξ is composed of the robot pose (x, y, θ) and the positions of all n features $(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$, that is,

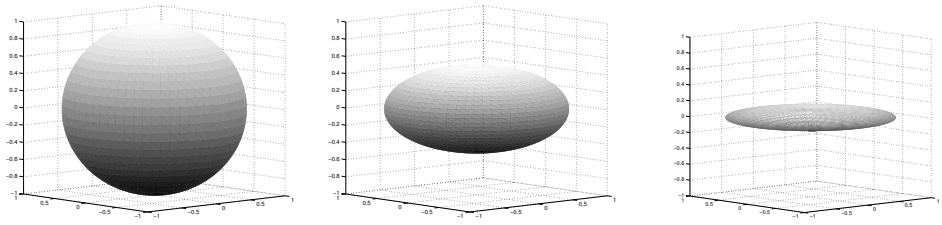
$$\xi = (x, y, \theta, x_1, y_1, x_2, y_2, \ldots, x_n, y_n). \tag{1}$$

The SLAM problem is to estimate the posterior distribution over ξ from the history of the controls issued by the robot and the sensor measurements received:

$$p(\xi | z_t, u_t, z_{t-1}, u_{t-1}, \ldots, z_0, u_0) \tag{2}$$

Given probabilistic models of the motion and sensors, there are a spectrum of competing SLAM algorithms for estimating the posterior distribution over ξ. Different representations of the map posterior trade off computational complexity for representational power and approximation

[1]This assumption ensures that information gain is strictly monotonic. Allowing previously-unsensed map features to be added to the process is perfectly acceptable, but will require additional book-keeping during the planning process when we add sense new features. We defer this question for future work.



(a) All eigenvalues at 1
$\det \Psi = 1$
$\mathrm{tr}(\Psi) = 3$

(b) One eigenvalue at 0.5
$\det \Psi = 0.5$
$\mathrm{tr}(\Psi) = 2.5$

(c) One eigenvalue at 0.1
$\det \Psi = 0.1$
$\mathrm{tr}(\Psi) = 2.1$

Figure 1. Three example uncertainty ellipses. Notice that we can drive the total volume of the uncertainty estimate close to 0 by shrinking only a single eigenvalue of the matrix covariance. In contrast, the trace preserves more of the overall uncertainty.

quality. In this paper we will focus on Kalman filter-based [9] approaches for development of our active exploration algorithm. The derivation of the extended Kalman filter (EKF) for SLAM has been described elsewhere [10]; it suffices to say that the EKF computes the posterior distribution of map and robot position at each time step through a linearization of the motion prediction and sensor measurement models, assuming that the motion and sensor uncertainty terms can be characterized as Gaussian. The assumptions of linearity and Gaussian noise allow the EKF model to maintain an approximation of the estimated posterior probability distribution over ξ as a Gaussian, that is, $p(\xi) = N(\mu, \Psi)$. While there are some known limitations to this approach, maintaining the explicit covariance matrix Ψ in the Kalman filter gives us principled ways to reason about how uncertain the distribution is about the model, and how the model estimate can be improved by different actions.

III. Active Exploration

Although the EKF allows us to build an accurate map from a stream of sensor data, it does not describe exploration strategies for gathering that data to build the most precise map. The question of how best to gather data is really the problem of selecting new measurements that are maximally informative about our model, which in turn raises the question of how to measure information. One popular measure of information gain of probability distributions is relative entropy [11]. If we define the entropy of a distribution as

$$H(p(\xi)) = \int_{\Xi} p(\xi) \log p(\xi), \tag{3}$$

then the loss in information after action a (as acting without sensing typically introduces uncertainty) is

$$\Delta I(a) = \left(H(p(\xi|a)) - H(p(\xi))\right). \tag{4}$$

After each action we then receive an observation; we cannot predict this observation exactly but we do know the distribution of observations, which allows us to compute the expected information gain. The complete change in information from time t to $t+1$ is then

$$\Delta I_{t+1|t} = \left(E_z\left[H(p(\xi|a,z))\right] - H(p(\xi))\right). \tag{5}$$

We can compute the relative change in entropy for a Gaussian distribution of d variables with mean μ directly from the covariance matrix Ψ, since

$$\begin{aligned} p(\xi) &= k \exp\left\{-\frac{1}{2}(\Delta\xi)^T \Psi^{-1} (\Delta\xi)\right\} && (6)\\ \Rightarrow H(p(\xi)) &= \frac{d}{2}(1+\log 2\pi) + \frac{1}{2}\log\det(\Psi) && (7)\\ \Rightarrow \Delta I_{t+1|t} &\propto E_z\left[\log\det(\Psi_{t+1})\right] - \log\det(\Psi_t) && (8) \end{aligned}$$

where $k = (2\pi)^{-d/2}\det(\Psi)^{-1/2}$ and $\Delta\xi = \xi - \mu$.

Consequently, maximizing information gain under the measure of relative entropy is equivalent to minimizing the determinant of the variance of the posterior distribution $p_t(\xi)$.

D-Optimality

Using relative entropy, and hence variance minimization, as an objective function is equivalent to the *d-optimality* criterion from experimental design theory [8]. Geometrically, the determinant is a measure of (although not equal to) the volume of the hyperellipsoid that bounds the uncertainty of the parameter estimate provided by the EKF. By minimizing the determinant, we are minimizing the volume of the model uncertainty. What a *d-optimal* exploration strategy fails to account for is an obvious property of the determinant; recall that the determinant of a square matrix is the product of its eigenvalues:

$$\det(\Psi_t) = \prod_{i=1}^{n} \lambda_i \tag{9}$$

where the λ_i are the eigenvalues of Ψ_t. We can drive the determinant of the entire covariance matrix to 0 by reducing a single eigenvalue to 0, making the matrix singular; a trivial way to do so is to reduce the variance of a single feature. The real problem here is that while the probability distribution of our data can shrink rapidly in some dimensions, in other dimensions we gain no information; using entropy as an information measure does not capture this phenomenon. Figure 1 illustrates this potential failure of the *d-optimal* objective function. By reducing the magnitude of a single eigenvalue, we see in this figure that the volume of the ellipse shrinks rapidly.

A-Optimality

A better measure of map quality can be found by viewing the EKF as a procedure to maximize the log-likelihood of the data under the assumption of Gaussian error, which is equivalent to minimizing the squared error between the

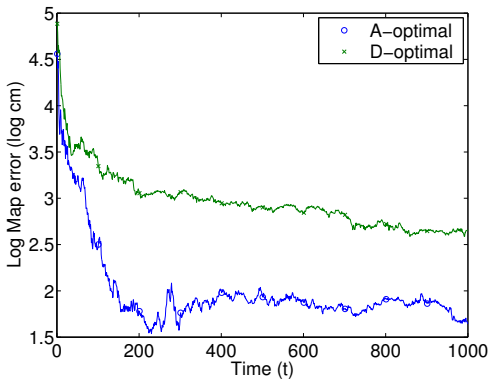


Figure 2. Map accuracy as a function of time for strategies optimizing an *a-optimal* and a *d-optimal* measure of uncertainty using a closed-loop, greedy strategy..

data and the model parameters. Rather than using the *d-optimal* measure to minimize overall variance, we can use the *a-optimal* information measure to minimize the mean squared error [8] of the model. Whereas the *d-optimal* information measure uses the product of the eigenvalues, the *a-optimal* information measure uses the sum (which is proportional to the mean for a fixed number of landmarks) of the eigenvalues, given by trace of the matrix:

$$I(\Psi_t) = \sum_{i=1}^{n} \lambda_i = \mathrm{tr}(\Psi_t) \quad (10)$$

$$\Rightarrow \Delta I_{t+1|t} = E_z\left[\mathrm{tr}(\Psi_{t+1})\right] - \mathrm{tr}(\Psi_t) \quad (11)$$

Minimizing the mean of the eigenvalues allows us to minimize the "average" uncertainty of the model, which we will see below gives a more accurate map than the variance-based approach. The trace is sometimes considered to be a poor estimate of information of a multi-variate Gaussian, because unlike the determinant, the minimum of the trace is not invariant to different scaling of the parameters. Also, unlike the determinant, the trace may not be physically meaningful when the quantities are in different units; it is not clear how to compute the mean of different uncertainties such as location, velocity and acceleration. However, for the SLAM problem, all variables represent spatial locations except for the orientation of the robot, and so long as we take care to use the same units for representing all map features and vehicle parameters, we are not really subject to scaling problems. One approach to handling the different units of orientation may be to marginalize out the orientation; in practice the approximation introduced by including the orientation in the trace is minimal so long as the orientation is represented in reasonable units (i.e., radians rather than arc-seconds). The key point is that despite the vulnerability to scaling that could result from poor representations of the covariance, the *a-optimal* measure still does a better job of capturing the overall uncertainty of the model, resulting in more accurate maps.

Figure 2 illustrates the improvement from an *a-optimal* measure compared to a *d-optimal* measure for greedy local search. The plot depicts the mean error between twenty map feature estimates and their ground truth positions as a function of time. For both curves, the same map and optimization strategy are employed (described in the next section); only the measures that are optimized differ. While both measures generally demonstrate improvements over time, the strategy employing an *a-optimal* measure produces a significantly more accurate map.

It is worth noting that many exploration algorithms attempt to deal with the problems inherent in conventional variance minimization differently, by ignoring correlations between most parameters of the EKF (in other words, assuming an *ad hoc* independence between map features). This approximation allows the determinant of Ψ to be computed as

$$\det \Psi \approx \det \Psi_v + \sum_{i=1}^{n} \Psi_i, \quad (12)$$

where Ψ_v is the vehicle covariance, a sub-block on the diagonal of Ψ, and Ψ_i are the map feature covariances, also sub-blocks on the diagonal of Ψ. This approximation will tend to under-estimate the uncertainty of covariance matrices by ignoring cross-correlations between map features and between features and the vehicle position.

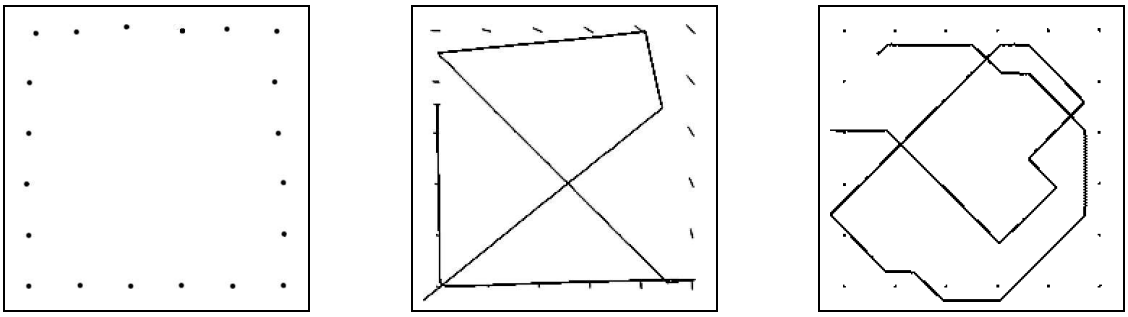
IV. Global Planning

Having addressed the problem of selecting the correct objective function for exploration, we turn to the problem of global exploration. We will assume that we have discretized the environment into a grid, where each grid cell represents a specific robot position; the actions in the planning problem are motions from grid cell to grid cell. The planning problem is how to choose the sequence of discrete positions through the environment that, taken together, form the maximally informative trajectory. Without loss of generality, we will assume a quasi-holonomic robot in order to avoid explicitly planning the orientation of the robot, being mindful that tracking the robot's orientation is essential for the actual mapping problem. Again without loss of generality we will ignore the problem of obstacles in the environment, since obstacles can be easily integrated into the planning problem as grid states that cannot be included in any trajectory.

Current exploration approaches [5], [6], [7] have used purely local, greedy approaches for choosing the single best next action that maximizes the expected information gain, as in

$$\pi(\xi, \Psi) = \underset{a}{\mathrm{argmin}}\, E[\mathrm{tr}(\Psi)]. \quad (13)$$

However, figure 3 shows the disadvantage to the standard greedy approach. The true map is shown in figure 3(a), and the greedy trajectory is shown in figure 3(b). We see the robot criss-crosses from corner to corner, resulting in poor estimation of the map features between the corners. Figure 3(c) shows an example trajectory from our global planning algorithm, that loops around all map features and results in a more accurate map. In particular, the global planning algorithm appears to deliberately "close the loop", that is, revisit already-explored areas in order to improve the overall map estimate. The map feature uncertainty ellipses are shown in both figures, and the greedy trajectory clearly results in larger ellipses and therefore higher uncertainty.



(a) Initial Map (b) Greedy Trajectory (c) Global Trajectory

Figure 3. (a) An exploration problem, with a set of features arranged in a ring around the edge of the environment. (b) The trajectory from a greedy exploration algorithm, computing the single position in the environment with maximum expected information gain, and moving to that position. (c) The trajectory from our global planning algorithm; notice the deliberate loop-closing.

The difficulty with planning a sequence of actions $\{a_0, a_1, \ldots a_t\}$, so that

$$\pi(\xi, \Psi) = \underset{a_0, a_1, \ldots a_t}{\operatorname{argmin}} \operatorname{tr}(\Psi_t), \tag{14}$$

is that the computational cost incurred can be large. In trying to choose the optimal sequence of actions, we are in reality trying to compute the optimal trajectory through the space of expected EKF estimates

$$\pi(\xi, \Psi) = \underset{(\xi_0, \Psi_0), (\xi_1, \Psi_1), \ldots (\xi_t, \Psi_t)}{\operatorname{argmin}} \operatorname{tr}(\Psi_t). \tag{15}$$

Each of these EKF estimates is a belief over map feature locations and robot positions. Solving for the optimal sequence of actions exactly is therefore an instance of a Partially Observable Markov Decision Process (POMDP), which typically involves enormous computational cost to solve any but the smallest toy problems. Good approximation methods have been described recently to solve many POMDPs approximately [12], [13], but the exploration problem has structure that allows even greater computational efficiency.

A. Pruned Breadth-First Search

Our approach to the planning problem is to use breadth-first search over all robot positions to search for the expected sequence of EKF estimates that lead to the maximum information gain (A^* search would be even better, but the choice of a good heuristic is an open question). We will take as the space of poses the discretized grid of the environment, and the space of actions as motions to the eight-connected neighborhood of each grid cell; in general, this search would be exponential in both the number of states and search depth, which for a reasonably sized environment will be unacceptable. However, by pruning the breadth-first search carefully, we can avoid some of the exponential cost.

The two key insights of our planning algorithm are as follows.

- For the purposes of planning, we can express the state of the EKF as the mean state of the robot and the trace:

$$(\xi, \Psi) \approx (x, y, \operatorname{tr}(\Psi)). \tag{16}$$

  This is an approximation, but by discretizing the EKF estimate of the robot position onto the grid of our action space, we can represent the state in the search as a vector of length 3, rather than a complete mean and covariance. Note that this planning approximation is very reminiscent of the Coastal Navigation heuristic used for POMDP-style navigation [14]. The major differences are that we are using the *a-optimal* heuristic representation, and we are using search rather than dynamic programming.
- We will limit the search to trajectories that do not contain repeated states; that is, trajectories that do not self-intersect in the discretization. This may seem like an overly restrictive policy class, but the advantage is that it allows us to prune the search tree substantially. Consider a search tree that has already expanded a node with state (x, y, I), and is about to expand a node with state (x, y, I'); these two nodes correspond to two trajectories t and t' that will arrive at the same robot location but with different expected information gain along the trajectories. If $E[\operatorname{tr}(I)] < E[\operatorname{tr}(I')]$, then the only way the trajectory t' can do better than trajectory t, gaining more information starting from the current state of (x, y, I'), is to return to (x, y) with an information measure I'' such that $E[\operatorname{tr}(I'')] < E[\operatorname{tr}(I)]$. By explicitly disallowing such trajectories, we can prune away much of the search space.

The restricted policy class effectively converts the search from a 3-D process into a 2-D process. Rather than representing, for each (x, y), all achievable information states $\operatorname{tr}(\Psi)$, we need only represent the *best* information state $\operatorname{tr}(\Psi)^*$. We can think of the state of the search process as an "information surface" over the space of robot positions, where the height of the surface is given by $\operatorname{tr}(\Psi)^*$, the best information state found for a trajectory passing through this state.

We embed the search tree in the information surface by adding to each position (x, y) in the grid a "parent pointer", $\phi(x, y) = (x', y')$, that indicates which neighboring position (x', y') in the grid contributed to the current best estimate $\operatorname{tr}(\Psi)^*$ at (x, y). During our search, we only update the information gain at the current pose (x, y) if the information gain estimate is improved and we can

follow the sequence of parent pointers back to the robot's start pose (x_r, y_r) without encountering pose (x, y) again. Finally, we use a priority queue keyed on the current entropy of the distribution to minimize repeated iteration over all robot poses. The full algorithm is given in Table I.

1) Initialize all $I(x, y) = \infty$
2) Push current $\{x, y, \xi, \Psi\}$ onto Q, with priority $p = \mathrm{tr}(\Psi)$
3) While Q not empty
   a) Pop $\{x, y, \xi, \Psi\}$
   b) For each neighbor (x', y') of (x, y):
      i) Compute ΔI from equation (11)
      ii) If $I(x, y) + \Delta I < I(x', y')$ and $(x', y') \notin \{\phi(x, y), \phi(\phi(x, y)), \ldots, (x_r, y_r)\}$ then
         A) $I(x', y') = I(x, y) + \Delta I$
         B) Push $\{x', y', \xi', \Psi'\}$ onto Q with priority $p = I(x', y')$
         C) Set $\phi(x', y') = (x, y)$
4) $(x, y) = \mathrm{argmin}_{(x,y)} I(x, y)$
5) while $(x, y) \neq (x_r, y_r)$
   a) $(x'y') = (x, y)$
   b) $(x, y) = \phi(x, y)$
6) Move to (x', y')

Table I

THE COMPLETE ALGORITHM FOR FINDING THE TRAJECTORY TO THE GLOBAL MAXIMUM IN INFORMATION GAIN.

It should be noted that our planning algorithm requires an initial state estimate for all of the landmarks. In practice we have the robot take a small number of initial observations before computing the global plan. One outstanding issue is the question of how often the global plan should be recomputed to close the loop around sensor data. Given the computational cost, it is desirable to re-plan as infrequently as possible. The drawback of not re-planning is that as the actual state estimate drifts from the initial estimate, the computed plan becomes sub-optimal. For this work, we compute an initial global plan and recompute only when the robot has visited all of the states in the plan once. The plan is then recomputed based on the new state estimate.

Convergence of the algorithm is guaranteed by the fact that no state can be repeated on any given trajectory, and the fact that the information gain at each point (x, y) increases monotonically as more informative trajectories are expanded; trajectories with less information are pruned immediately. The complexity of this algorithm for s states is a result of $\mathcal{O}(s^2)$ state updates, and $\mathcal{O}(m)$ iterations for each state to follow the parent pointers, checking each trajectory of maximum-length m for loops. If we disallow trajectories that re-visit states, m is bounded above by s, and the complexity is therefore $\mathcal{O}(s^3)$, hence cubic in the number states. In practice the running time is much less than cubic, and the bound can be further reduced by shortening the planning horizon m.

V. EXPERIMENTAL COMPARISON

We tested three open-loop exploration algorithms in a simulated environment, looking at example trajectories and a quantitative comparison of each algorithm's performance. The three algorithms are as follows:

- Random exploration
- Greedy exploration: The robot drives to the single location that is maximally informative about the environment as developed by Feder et al. [5], [6].
- Breadth-first Search: This is the algorithm summarized in Table I, to compute the trajectory with the highest terminal information gain.

There are 20 map features distributed randomly about a square environment of size $200m \times 200m$ and the robot generally moves in steps of $1m$. For these simulated experiments we allow the sensor to have infinite range; we take this approach because none of the algorithms under consideration emphasize coverage explicitly, a topic for future investigation. We employ a sensor model in which the variance of a range measurement is proportional to the distance to the feature being measured, and the standard deviation of a bearing measurement is a constant five degrees. As such, measurements to distant features are noisier than measurements to nearby features. We assume that at the outset the robot knows how many features there are, but of course not where they are. In all cases, the algorithm replans (e.g., computes a new information surface) whenever it reaches the intended target destination.

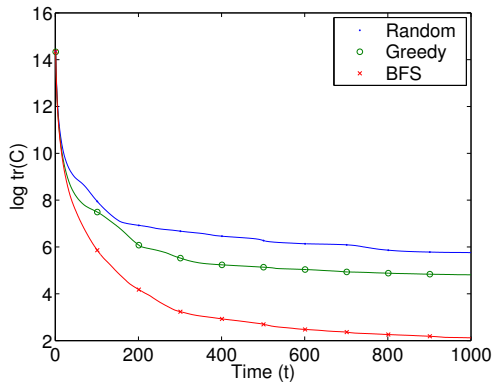
Figure 4 shows quantitative comparisons of the performance of the three algorithms, over trajectories of length 1000 steps. At the top is the trace of the covariance matrix. Notice that the Breadth-First Search algorithm converges the fastest of all the algorithms. Comparing the accuracy of the map with ground truth in terms of the log average L_2 norm between the estimated feature positions and their true positions, we see that the Breadth-First Search algorithm had significantly higher accuracy than any other approach[2]. It is worth noting that if these experiments were continued out to an infinite number of time steps, the entropy of each algorithm would converge to the same global minimum, as would the map accuracies. The idea is that the Breadth-First algorithm should be accurate sooner.

VI. DISCUSSION

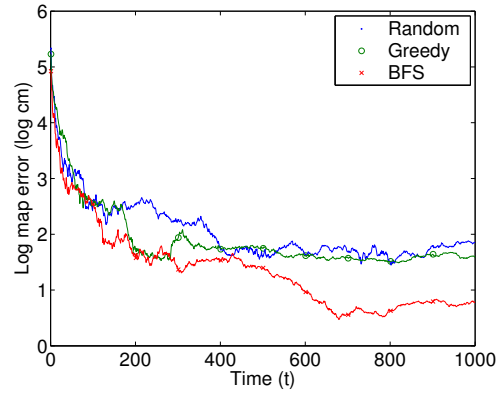
We have described an approach for active exploration on mobile robots that uses the *a-optimal* measure of information gain and a global optimization strategy. The *a-optimal* measure allowed a local, closed-loop motion strategy to outperform the existing *d-optimal* information measure, learning a more accurate map in the same amount of time. Additionally, our global optimization strategy was able to build a more accurate map using an open-loop exploration strategy.

Again, the goal of this paper was not to demonstrate a provably optimal algorithm for exploration, but show a more appropriate choice of objective and to show that global exploration strategies can be found tractably. The computational cost of global exploration strategies is still admittedly high, and re-planning after every observation is difficult (but not impossible). Our experimental results

[2]It is possible to introduce arbitrary error in the L_2 norm between a perfect map and ground truth by rotating the map about the start pose before comparison. The error reported here for all algorithms follows a correction procedure to rotate the map back to the best orientation possible for minimizing error.



(a) Trace of Posterior Covariance



(b) Map Accuracy

Figure 4. Performance results for the 3 algorithms: at the top is the convergence of the posterior trace with time, and on the bottom is the log average L_2 error between the true and estimated feature positions.

demonstrate that re-planning after every observation is clearly not necessary, as our planner performed well even in the unreasonably pessimistic case of an open-loop plan. We will address the problem of *when* to replan in future work.

One advantage to our approach we have not explored is that measuring information gain using the *a-optimal* measure does not depend on the number of map features in the world when a finite number of map features are visible; although the trace obviously does depend on the size of the covariance, computing changes to the trace, and therefore the information *gain* does not. Given the recent development of a number of constant-time SLAM algorithms, such as CTS [3] and the Sparse Extended Information Filter [4], an exploration algorithm that also avoids any computational complexity dependence on the number of map features will be highly applicable.

There are two additional limitations to the algorithm which we plan to address in future work. Firstly, our use of properties of the covariance matrix has a strong dependence on the Kalman filter-based SLAM model. However, many techniques that build good metric maps are not based on point features but rather on scans of range data, images, etc. These algorithms are still probabilistic in nature; if we can capture an *a-optimal* information measure over these representations, we should be able to extend our technique easily.

Secondly, our approach is an approximation in that we are using a restricted class of policies, that is, discretized policies that do not self-intersect. This approximation is for purely computational reasons; we could easily allow self-intersecting policies by no longer pruning trajectories that have repeated states. Some of the most useful exploration trajectories, however, are likely to be very loopy. Further analysis is required to assess the computational penalty this will incur. The restriction to discretized policies can be relaxed to include continuous-state policies by using a policy search algorithm to improve the discrete-state policy. We are also currently investigating alternative discretizations based on local maxima in the greedy search space.

Acknowledgments: The authors gratefully acknowledge the suggestions and insight of Sebastian Thrun, Geoff Gordon and John Leonard.

REFERENCES

[1] M. W. M. G. Dissanayake, P. M. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotic and Automation*, vol. 17, no. 3, 2001.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit., "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

[3] P. M. Newman and J. J. Leonard, "Consistent convergent constant time SLAM," in *Proc. International Joint Conference on Artificial Intelligence (IJCAI*, Acapulco Mexico, August 2003.

[4] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," April 2003, in submission.

[5] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," *International Journal of Robotics Research, Special Issue on Field and Service Robotics*, vol. 18, no. 7, pp. 650–668, July 1999.

[6] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, "Information based adaptive robotic exploration," in *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[7] C. Stachniss and W. Burgard, "Exploring unknown environments with mobile robots using coverage maps," in *Proc. of the International Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

[8] V. Fedorov, *Theory of optimal experiments*. New York: Academic press, 1972.

[9] E. Kalman, Rudolph, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[10] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, June 1991.

[11] D. J. C. MacKay, "Bayesian methods for adaptive models," Ph.D. dissertation, California Institute of Technology, 1991.

[12] N. Roy and G. Gordon, "Exponential family PCA for belief compression in POMDPs," in *Advances in Neural Information Processing Systems 15 (NIPS)*, S. Becker, S. Thrun, and K. Obermayer, Eds. Vancouver, BC: MIT Press, 2003, pp. 1043–1049.

[13] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," in *Proc. IJCAI*, Acapulco, Mexico, August 2003.

[14] N. Roy and S. Thrun, "Coastal navigation with mobile robots," in *Advances in Neural Processing Systems 12*, vol. 12, 1999, pp. 1043–1049.