Active Exploration Planning for SLAM using Extended Information Filters

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 [3]. However, if the exploration trajectory accumulates new information about the world slowly, then convergence of the map can be slow. By making use of the recent development of constant-time SLAM algorithms, we show how information gain for a single step can be computed in constant time. We describe the concept of an "information surface", which represents at each point in the environment the total potential information gain that results from a complete trajectory to that point. We demonstrate an algorithm for finding this surface that leads to an efficient, global planning algorithm for exploration that is linear in the number of states to be explored in the world and the length of the trajectory.

1 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 [11], and can be a technical challenge because the robot position and the world features must be estimated simultaneously from noisy sensor data. Recent research however, 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.

Although robots can now build good maps of many real environments directly from their own data, there has been little research on how to collect the data autonomously. Typically, a robot is driven around the environment by hand while it records sensor data; this sensor data is either assembled incrementally and online into a partial map, or the sensor data ensemble is assembled offline into a globally consistent map. There are relatively few results concerning how to automate this exploration policy and how best to gather data for building a good map, even though the motion strategy can have a real impact on the quality of the resulting map. When the sensor data is collected from a robot being controlled manually by a novice, the result is often poor maps with internal inconsistencies. A more seasoned controller has an internal model of good motion strategies that gather information quickly, that do so in a way that makes it easier to assemble a consistent map. For example, a good motion strategy can detect when the robot is becoming too uncertain about its pose, and return to known landmarks to re-localize before continuing to explore. A good motion strategy can also detect actions that have resulted in higher error (e.g., point turns that typically result in higher degrees of wheel slippage, etc.) and avoid them, reducing odometry error that must be corrected after the fact.

In this paper, we describe a motion planning algorithm for SLAM that computes the multi-step trajectory that reduces the uncertainty of the map the most. The algorithm uses a variant of the extended Kalman filter called the sparse extended information filter (SEIF). The advantage to the information filter is that a single update step from moving and sensing can be done in constant time, which allows us to compute the change to the map certainty in constant time. By restricting ourselves to the class of trajectories that do not contain cycles, we can compute a 2-D "information surface" that encodes at each point the improvement to the map resulting from travelling to that point. The information surface is not just the information *available* at each point, but the information gained by integrating along the trajectory to the point. We describe an efficient algorithm for computing this surface over a discrete representation of space, using a dynamic-programming approach and the constant-time information gain update. This results in one of the first algorithms that allows exploration for exploiting global information along trajectories, rather than exploiting local information gain via gradient-descent.

2 SLAM

We will assume that we have a quasi-holonomic robot operating in a planar environment. The robot is equipped with a landmark sensor that can sense the range and bearing of any number of landmarks in the plane that are closer than some finite range. The robot issues controls at each time t, which consist of the relative translational and rotational displacement: $U_t = (\Delta d, \Delta\theta)$. The sensor measurements are a sequence of range and bearing measurements: $z_t = (r_1, b_1, r_2, b_2, \ldots r_m, b_m)$ for m landmarks within range of the sensor. The SLAM problem is then how to simultaneously estimate the robot pose (x, y, θ) as well as the positions of all n landmarks $(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$, that is, estimate the full state

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

based only on knowledge of the controls issued by the robot and the sensor measurements received. The state at time t is related to the previous state and control by a function g,

$$\xi_t = g(\xi_{t-1}, U_t, q_t), \tag{2}$$

where q_t is an unobservable noise term. The observation at time t is related to the state by a function h,

$$z_t = h(\xi_{t-1}, v_t), \tag{3}$$

where v_t is an unobservable noise term. Given appropriate probabilistic models of these noise terms, a posterior distribution over the full state ξ can be computed from the full history of controls and observations. Different representations of this posterior trade off computational complexity for representational power and approximation quality.

Although there are now a spectrum of competing SLAM algorithms, we will focus on Kalman filter-based approaches for development of our active exploration algorithm. The extended Kalman filter (EKF) computes the posterior distribution by linearizing the functions g and h about the previous state estimate, and making assumptions that the noise terms can be characterized as Gaussian. The exact derivation of the extended Kalman filter has been described elsewhere [7, 18, 8]; it suffices to say that the EKF algorithm consists of iterating two steps: a prediction step that estimates the posterior after each control, and a measurement step that integrates each observation, refining the position estimate. By linearizing and making Gaussian assumptions, the EKF model ensures that the posterior probability distribution over ξ is always Gaussian. While there are some known limitations to this approach, maintaining the explicit covariance matrix Σ in the Kalman filter allows us to reason about how certain any distribution is, and how the distribution is improved by different actions.

The Extended Information Filter

The dual to the extended Kalman filter is the extended information filter (EIF). Instead of maintaining the covariance matrix Σ_t and mean state estimate μ_t, the EIF maintains the inverse of the covariance, called the information or precision matrix Ψ_t, and an information vector b_t. These terms are related to the original Kalman filter terms in the following manner:

$$\mu_t = \Psi_t^{-1} b_t^t \tag{4}$$
$$\Sigma_t = \Psi_t^{-1} \tag{5}$$

Just as the EKF has a two-stage update process, so does the information filter. The full EIF update steps can be found elsewhere [10, 17], and are outside the scope of this paper.

The measurement update of the EIF can easily be shown to be constant time because it simply involves adding a sparse matrix to the information matrix. The prediction update in the standard formulation of the EIF is not, however, since it contains matrix inversions. As the number of landmarks grows, so does the difficulty of these inversions. Thrun et al. [17] prove that by maintaining sparseness in the information matrix Ψ_t and tracking the mean μ_t, the motion update can still be performed in constant time. Tracking μ_t exactly is still not a constant time operation, but there exists an iterative approximate algorithm to recover μ_t if Ψ_t is sparse. Given an upper bound on the number of permitted iterations, the recovery of μ_t is then also constant-time algorithm. The sparseness-preserving algorithm, together with estimation of the information vector b_t is referred to as the Sparse Extended Information Filter (SEIF).

The major issue in the SEIF is maintaining the sparseness of Ψ_t, in that as new landmarks are seen, new non-zero entries are added in the growth of the information matrix. Additionally, as the robot moves around, the mutual information between two visible landmarks is increased. The sparsification procedure of the SEIF makes use of the fact that, although the information matrix tends towards a non-sparse matrix, many of the entries are close to 0. By only keeping the n largest entries of the information matrix, the sparseness of the matrix can be bounded from below, resulting in an upper bound on the update time of the algorithm (the greater the sparseness, the faster the update). Additionally, a constant-time sparsification algorithm exists for maintaining this bound; so long as this bound is appropriate for the environment and robot, the approximation error in the posterior introduced by the sparsification is negligible.

3 Active Exploration

The SEIF algorithm allows us to use collected data to build an accurate map efficiently. However, we still do not have an algorithm for gathering that data efficiently to build the most precise map. The problem of gathering data efficiently is really one of selecting new measurements that are maximally informative about our model. The EKF, and by extension the SEIF, are both generative Bayesian estimators, which means we should select new data that are maximally informative about our probabilistic model of ξ. Decision theory tells us that we can compute the gain in information between any two distributions as the relative change in entropy [9]. We will therefore choose exploration strategies that maximally reduce the entropy of the posterior distribution $p_t(\xi)$.

The entropy of a distribution is defined as

$$H(p(\xi)) = \int_{\Xi} p(\xi) \log p(\xi) \quad (6)$$

which for a Gaussian distribution can be computed directly from the covariance matrix as

$$p(\xi) = (2\pi)^{-\frac{d}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\xi-\mu)^T\Sigma^{-1}(\xi-\mu)\right\} \quad (7)$$

$$\Rightarrow H(p(\xi)) = \frac{d}{2}(1+\log 2\pi) + \frac{1}{2}\log\det(\Sigma) \quad (8)$$

$$\propto \log\det(\Sigma). \quad (9)$$

The maximally informative trajectory must therefore have the smallest covariance matrix Σ. We do not explicitly maintain the covariance matrix in the SEIF but instead the information matrix $\Psi = \Sigma^{-1}$, which means that the gain in information from time t to $t+1$ is

$$\Delta H = \log(\frac{1}{\det \Psi_{t+1}}) - \log(\frac{1}{\det \Psi_t}) \quad (10)$$

If we find the shortest trajectory that minimizes this quantity, we should converge to the most accurate map the fastest.

Constant-time Information Gain

Although we can compute the posterior information matrix after an action and an observation in constant time using the SEIF, the information gain is not yet a constant-time process, depending on our ability to compute the determinant of an arbitrary size matrix. We can again make use of the fact that the information matrix is sparse to eliminate the information gain's dependence on the size of the information matrix. If we bound the number of non-zero entries in Ψ_t, we can find a constant-time algorithm for computing the information gain in the map submatrix. Using the sparseness of Ψ_t, we swap rows and columns of the Ψ_t to get an block diagonal matrix $\hat{\Psi}_t$ of the form

$$\hat{\Psi}_t \approx \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix}$$

where X contains terms for landmarks that the robot can see and Y contains terms for the remaining landmarks. The effect on the determinant of re-writing the information matrix in this way is minimal: the determinant is affected by at most a sign change. (The magnitude of the determinant of a matrix is unaffected by any number of row and column exchanges.) Once we have the information matrix in this form, the determinant of the block diagonal matrix is the product of the determinants of its blocks:

$$\det \hat{\Psi}_t = (\det X)(\det Y) \quad (11)$$

$$\Rightarrow \Delta H = \log(\det X_{t+1} \det Y)^{-1} - \log(\det X_t \det Y)^{-1} \quad (12)$$

$$= \log(\det X_t) - \log(\det X_{t+1}) \quad (13)$$

Let us suppose that the environment, and therefore the information matrix, is sufficiently sparse that the robot can only see one landmark at a time. This means that the upper-left block of H will always be a 5×5 matrix assuming each landmark has exactly 2 degrees of freedom and the robot has 3. If we want to choose some location in the environment that will maximally reduce the determinant of the covariance matrix, then we can do this by considering only a 5×5 matrix at each point (x, y).

We know, from the sparse extended information filter literature that we can update the posterior, represented by the information vector and matrix, in constant time. We have demonstrated that the information gain between the prior and posterior information matrix can also be computed in constant time, given a lower bound on the sparseness of the information matrix. We now describe an algorithm for computing the global trajectory that maximizes the information gain.

4 The Information Surface

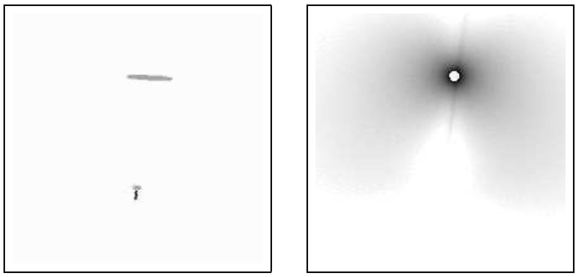
Our approach to exploration planning is to attach to each (x, y) pose in the environment the information gain that results with maximum likelihood from moving from the current (x_r, y_r) pose of the robot to the destination pose (x, y). Computing this information gain over the space of all possible poses gives an "information surface" over the environment; this surface allows us to identify the maximally informative point in the environment, given the current state. By iterating between computing the information surface and then travelling to the global maximum, the map should converge to minimal error as quickly as possible. We approximate the information surface using a discretization; the discretization is simply a restriction to a class of policies over a discretized pose space[1]. Because we can compute updates to the SEIF in constant time, we can compute the information surface efficiently, in time $\mathcal{O}(n)$ in the number of states in the discretized pose space.

4.1 The Single-Shot Information Surface

One possible model of the information surface is the information gained after integrating a single (typically long) step from the start pose to the end pose and then integrating the single maximum-likelihood measurement for that pose. We will call this approach the "Single-Shot Information Gain". Figure 1(a) depicts an example robot pose (lower cross) and uncertain landmark (elongated ellipse). Figure 1(b) depicts the information surface that results from this problem. The robot has large error in the rotational control, which results in a ridge of high information gain along the poses that require little or no rotation. Notice that the information gain is

[1]This restriction can be relaxed to include continuous-state policies using a policy search algorithm to improve the optimal discrete-state policy [14], but in practice we have not found this to be necessary.

also higher off to the sides of the landmark, in the direction of the principal axis of the covariance of this landmark. In general, the shape of the information surface will depend on both the robot's sensor and motion models, and can vary significantly between noise models with different properties.



(a) Current Robot Pose and Map

(b) Information Surface

Figure 1: (a) An example problem, with the current robot position shown in the middle of the image, and a single landmark with its covariance ellipse shown in the top right. (b) The information surface due to taking a single step to each destination robot pose. Darker areas contain more information. The white circle is the minimum range of the sensor around the landmark.

Unfortunately, the Single-Shot Information Gain may not be the most useful model of information for two reasons. First, unless the trajectory is very short, the robot will receive additional measurements along that trajectory which should be integrated into the estimate of ξ and will add to the certainty of the map[2]. We can extend this algorithm by actually integrating along each straight-line trajectory, but this is inefficient and does not solve the second issue: by computing the information gain along a line to each pose, we have restricted exploration to straight-line trajectories when the most useful trajectories might be curved.

4.2 The Integrated Information Surface

Instead of computing straight-line trajectories, a better approach is to consider trajectories of arbitrary sequences of unique (non-repeated) robot poses. We can do this using the following three ideas:

- The information gain at the start location of the current robot pose (x_r, y_r) is 0. This approximation drives the forward progress of the robot.
- The information gain at any point (x, y) relative to start pose (x_r, y_r) can be computed from the information gain from (x_r, y_r) to some neighbouring pose (x', y') and the information gain from (x', y') to (x, y). If we know the information matrix at each neighbouring pose (x', y') under the optimal trajectory to each neighbour, we can therefore compute the maximally informative trajectory to the pose (x, y).
- Whichever neighbour (x', y') leads to the maximum information gain of (x, y), that neighbour is the parent pose in the optimal trajectory to (x, y).

[2]Note that if the robot elects to take a large step without collecting new measurements along the path, the risk of divergence in the filter increases considerably.

Since we can compute the optimal information gain of a robot pose from the covariance of the best neighbouring pose (assuming we store the covariance of neighbouring poses as those covariances are assembled from *their* neighbours), we can iterate through all poses repeatedly until we converge to a consistent, unchanging information surface. This leads to the following naïve algorithm, very reminiscent of value iteration for Markov decision processes:

- Until the information surface stops changing:

1. For each robot pose (x, y), compute the best posterior information matrix from the neighbours and the information gain
2. If the new information gain estimate (relative to the start pose (x_r, y_r)) is better than the previous estimate for the information gain at (x, y), then set the information gain of (x, y) to the new gain estimate, and store the posterior information matrix associated with this pose.

The information surface computed in this way is related to the value function from Markov Decision Processes. The immediate reward of a state transition is the immediate information gain from the current information matrix. The main difficulty with this algorithm, however, is that it includes all sequences of poses, not just all sequences of non-repeated poses. When we compare some existing estimate of the information gain at a pose (x, y), we need to ensure that the information at this pose has not *already* been included in the information gain estimate. If we do not prevent information from being integrated twice, the algorithm can continue to drive the information gain up by iterating back and forth between neighbouring poses. We want to explicitly exclude such trajectories.

We therefore augment each pose in the information surface with a "parent pointer", $\phi(x, y)$, that indicates which neighbouring pose contributed to the posterior at the current pose. During the iteration over poses, we only update the information gain at the current pose (x, y) if the information gain improves and when we can follow the sequence of parent pointers back to the 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 1.

Convergence of the algorithm is guaranteed by the fact that no state can be repeated on any given trajectory, and the fact

1. Initialize all $I(x,y) = \infty$
2. Push current $\{x, y, b, \Psi\}$ onto Q, with priority $p = -\log\det\Psi$
3. While Q not empty
   (a) Pop $\{x, y, b, \Psi\}$
   (b) For each neighbour (x', y') of (x, y):
      i. Compute posterior $\{b', \Psi'\}$ as a result of moving from (x, y) to (x', y')
      ii. Compute ΔH from equation (13)
      iii. If $I(x,y) + \Delta H < I(x', y')$ and (x', y') is not on the path $\phi(x, y)$ to (x_r, y_r) then
         A. $I(x', y') = I(x, y) + \Delta H$
         B. Push $\{x', y', b', \Psi'\}$ onto Q with priority $p = I(x', y')$
         C. Set $\phi(x', y') = (x, y)$
4. $(x, y) = \text{argmin}_{(x,y)} I(x, y)$
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 1: The complete algorithm for finding the trajectory to the global maximum in information gain.

that the state entropy at point (x, y) decreases monotonically as more informative trajectories are expanded. The complexity of this algorithm is a result of $O(n)$ state updates, and each state update involves a constant-time information gain operation, and m iterations to follow the parent pointers, checking the trajectory for loops. The complexity is therefore $(O)(mn)$ for n discrete states in the environment and the maximum trajectory of length m, hence linear in the number of state and the length of the longest trajectory.

5 Experimental results

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

- Random: At each time step, the algorithm takes a random control action $\Delta t, \Delta\theta$

- Most Uncertain Landmark: The robot drives successively to each landmark that it is most uncertain about (that is, the landmark with the largest covariance). This is a heuristic technique that assumes the most information can be gained by sensing near the most uncertain landmark. In order to prevent oscillation, the next landmark to visit is not considered until the robot reaches its current goal landmark.

- Gradient Descent: The robot moves in the direction of maximum-likelihood information gain. This is the most common form of active exploration found in the literature [19, 4, 1].

- Single Shot: This is the algorithm described in section 4.1, where the information surface is computed from a single (large) step of motion to each grid cell and the single maximum-likelihood measurement at that pose.

- Integrated Trajectory: This is the algorithm summarized in Table 1. The information surface is computed recursively by integrating successive one-step prediction and measurement steps along the trajectory to each grid cell. The robot then follows the trajectory to the pose with the highest information gain.

In all cases, the algorithm replans (e.g., computes a new information surface) whenever it reaches the intended target destination.

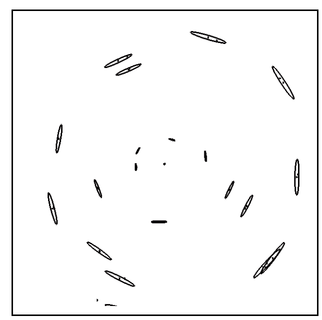
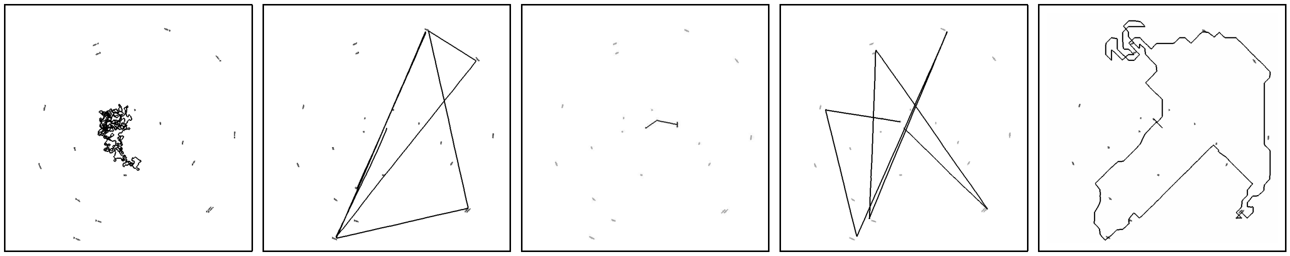


Figure 2: The initial map for the exploration tests.

Figure 2 shows the initial configuration for this problem. There are 20 landmarks 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 landmark being measured, and the variance of a bearing measurement is a constant five degrees. As such, measurements to distant landmarks are noisier than measurements to nearby landmarks. We assume that at the outset the robot knows how many landmarks there are, but of course not where they are. Figure 2 is after initializing the exploration with a random policy of 3 steps, in order to have a reasonable initialization of the information matrix before computing the information surface.

Figure 3 shows an example trajectory for each of the five algorithms. Notice that the gradient descent algorithm did very poorly, becoming trapped in a local maximum of information gain and fixating on a single landmark. The uncertain landmark and single-shot heuristics both cover space rapidly, but the straight-line nature of these two algorithms



(a) Random (b) Most Uncertain Landmark (c) Gradient Descent (d) Single Shot (e) Integrated Trajectory

Figure 3: Example trajectories using the four control strategies for exploration. Notice that even with infinite range, the Integrated Trajectory controller is the only one that aims to cover the entire space and close the loop.

means that neither is able to model the effect of small detours to reduce uncertainty on their way to some large information gain. Figure 3(e) shows our algorithm, and notice that it still explores the map rapidly, but periodically loops (sometimes making large loops, sometimes making very small loops). The intuition for this looping behaviour is that the robot periodically needs to re-localize by moving closer to well-localized landmarks, and register new measurements with the existing map. "Closing the loop", registering new map data with data acquired sometime ago, is one of the canonical difficult problems in SLAM. An active exploration algorithm that can model the effects of loops, and can close loops when appropriate will almost certainly build better maps than an heuristic approach.

Figure 4 shows quantitative comparisons of the performance of the five algorithms, over trajectories of length 1000 steps. On the left is the posterior entropy of the map covariance matrix, recovered from the information matrix. Oddly, the Most Uncertain Landmark heuristic does not perform significantly worse than the Integrated Information algorithm in terms of the entropy of the distribution. However, when we compare the accuracy of the map with ground truth in terms of the average L_2 norm between the estimated landmark positions and their true positions, we see that the Integrated Information algorithm had significantly higher accuracy than any other approach[3]. Notice that the Most Uncertain Landmark heuristic initially has a higher accuracy. However, the deliberate loop-closing of the Integrated Information algorithm means that it quickly recovers to a higher accuracy. 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 Integrated Information algorithm should be more accurate sooner.

[3] 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.

Figure 5 shows a final map built by the Integrated Information algorithm; we can see that the landmark covariances are all small and approximately equal. There are also few elongated ellipses, as the algorithm is able to appropriately model the information gain from different viewpoints of the landmarks, keeping the ellipses small in general.

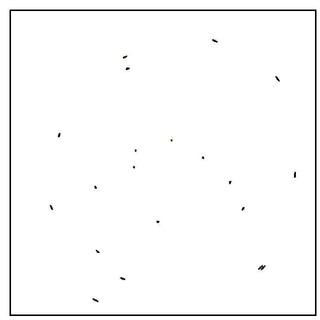
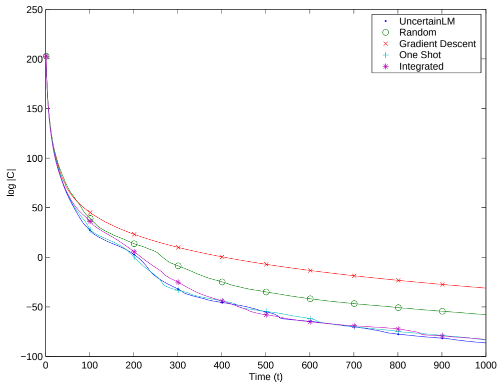


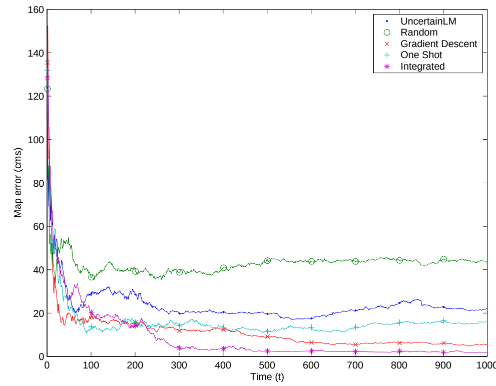
Figure 5: The final map produced by the Integrated Information algorithm.

6 Related Work

This is not the first work to consider the problem of active exploration for building the most accurate, lowest-uncertainty map. One popular approach in the literature, described in several places [19, 4, 1], is the gradient descent method discussed above. This has substantial computational efficiency, in that the information gain need only be computed for the 8-connected neighbouring states. The major disadvantage to the gradient-descent approach, as we saw in the experimental results, is that it is subject to local minima. While it has been argued that repeated observations taken from a local minimum will eventually flatten out the information surface at that pose, it may take a considerable amount of time for this process to occur, as is evidenced by the slow convergence of this approach in the experimental results. The gradient-descent approach also has no notion of "closing the loop" and cannot model long paths with large payoffs in map certainty at the end of the path.



(a) Posterior Entropy



(b) Map Accuracy

Figure 4: Performance results for the 5 algorithms: on the left is the total posterior entropy of a function of time, and on the right is the average L_2 error between the true and estimated landmark poses. Although the Integrated Information Gain algorithm does not converge to minimum entropy significantly faster than the Most Uncertain Landmark heuristic, the Information Information Gain consistently builds the most accurate map the fastest.

Stachniss et al. [16] describe one of the few global exploration algorithms, in that they explicitly compute the information available at all locations in the environment. Their algorithm is closest in spirit to our "Single-Shot" algorithm, although they integrate an additional notion of travel costs to gather data at different locations in the environment. The disadvantage to their approach is the same as the "Single-Shot" algorithm in that they are search for the maximally informative location, rather than the maximally informative trajectory.

The problem of active exploration is related to a number of other sequential decision making problems. The active localization problem is one where the robot's position is unknown but the map *is* known. Fox et al. [5] use the same criterion of entropy minimization to find a trajectory that localizes the robot as quickly as possible. Once again, the authors choose a purely local gradient-descent approach, although they augment the information surface with a notion of relative costs of different actions.

Active exploration and localization are both specific instances of a more general framework, known as the Partially Observable Markov Decision Process [15]. Whereas the Markov Decision Process computes the optimal action to take for any state, the POMDP policy provides the optimal action for any *distribution* over states. Conventional POMDP approaches are wildly computationally intractable, but good approximation algorithms have been emerged recently. The Augmented MDP (AMDP, also known as "Coastal Navigation") algorithm [13] is one that is particularly relevant, in that it provides an efficient approximation to the POMDP if the distributions are Gaussian, such as provided by a Kalman filter. The AMDP approach could be applied to the active exploration problem, if the information surface were replaced by an "information volume"; each voxel in the information volume would represent the current robot pose and map entropy. Given a transition function relating a voxel and action to some posterior voxel, the planning problem would be just be a shortest-path problem to the "lowest" reachable voxel in the information volume. The disadvantage to this approach is that computing the expected transitions in the information volume is infeasible for reasonable-sized maps.

Finally, active learning and statistical experiment design solve very related problems. The framework asks the question, given a space of possible data (or queries, or experiments), what is the one additional measurement that would maximally improve the existing model? [2, 6] The active learning approaches typically maximize the information gain criterion as we do in this work, but the disadvantage to these approaches is that they are not physically motivated; that is, there is no notion of "travelling" to acquire more data, there is no notion of a sequential decision making process that gathers a stream of data, and there is no notion of the relative costs of different queries. The active learning approaches are therefore closest to our "single shot" information surface that models the effect of a single motion and a single measurement.

However, some active learning and experiment design problems, such as geological surveys, have a physical component to them. In future work, we hope to be able to extend the active exploration algorithm described in this paper to problems where sequential decision making has implications in the active learning domain.

7 Conclusion

We have described an approach for active exploration on mobile robots that allows us to find the most accurate map in the least amount of time. The algorithm is not based on local features or distributions over local landmarks, but by computing a global information surface over the entire space and searching for the trajectory that results in the lowest entropy distribution. We emphasize that almost all existing approaches are not based on sequential decision making, but at best single-step look-aheads; we are *only* able to compute the information surface for arbitrary sized maps because of recent developments in constant-time SLAM that allow us to compute the information gains efficiently. In particular, we made use of the sparse extended information filter, however, other constant-time algorithms such as CTS [12] would be equally appropriate.

There are three specific disadvantages to our algorithm which we plan to address in future work. Firstly, we are computing the information surface based on the maximum-likelihood information gain, compared with the *expected* information gain. One possible way to do this is to use Monte Carlo methods to compute the expected information gain at each location, sampling from the sensor model distribution, but there may be more appropriate approaches as well.

Secondly, in all cases, the algorithm replans (e.g., computes a new information surface) whenever it reaches the intended target destination. However, it is clearly the case that as new information is acquired, the optimal trajectory may change. This raises the question then of when and how to recompute the plan. It is usually infeasible to recreate the information surface at the same rate that data arrives, but it is also unlikely that the entire information surface will change with the addition of new information; if we can devise a way to integrate motion and new measurements into the information surface directly, we should be able to make only local modifications to the surface and generate a new trajectory quickly, rather than integrating the measurements into the map and then creating a new information surface from scratch.

Finally, the Kalman filter-based SLAM algorithms have given us a principled way to reason about the quality of the current map, and predict the quality of future maps based on actions and predicted measurements. However, many techniques that build good metric maps are not based on landmarks but rather on scans of range data, images, etc.. These algorithms are still probabilistic in nature; if it is possible to compute some quantity over these maps that is similar to entropy of the map distribution, we should be able to extend our technique to more general map applications.

References

[1] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. Information based adaptive robotic exploration. In *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, EPFL, Lausanne, September 30 - October 4 2002.

[2] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[3] 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*, 17(3), 2001.

[4] 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*, 18(7):650–668, July 1999.

[5] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems*, 25(3-4):195–207, 1998.

[6] Y. Freund, H. Seung, E. Shamir, and N. Tishby. Selective sampling using the Query By Committee algorith m. *Machine Learning*, 28:133–168, 1997.

[7] Emil Kalman, Rudolph. A new approach to linear fi ltering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[8] John Leonard and Hugh Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

[9] D. J. C. MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, 1991.

[10] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Math. in Science and Engineering*. 1979.

[11] 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.

[12] Paul M. Newman and John J. Leonard. Consistent convergent constant time slam. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco Mexico, August 2003.

[13] Nicholas Roy and Sebastian Thrun. Coastal navigation with mobile robots. In *Advances in Neural Processing Systems 12*, volume 12, pages 1043–1049, 1999.

[14] Nicholas Roy and Sebastian Thrun. Motion planning through policy search. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, pages 2419–2424, Lausanne, Switzerland, September 2002.

[15] Edward Sondik. *The Optimal Control of Partially Observable Markov Decision Processes*. PhD thesis, Stanford University, Stanford, California, 1971.

[16] 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.

[17] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information fi lters. Submitted for journal publication, April 2003.

[18] Greg Welch and Gary Bishop. An introduction to the Kalman fi lter. Technical Report TR 95-041, University of North Carolina at Chapel Hill, Chapel Hill,NC, 2001.

[19] Peter Whaite and Frank Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Patten Analysis and Machine Intelligence*, 19(3):193–205, March 1997.