Probabilistic Resurvey Mission Planning

Cheng Fang and **Stefan B. Williams**
Australian Centre for Field Robotics
The University of Sydney, Australia
cfan6677@uni.sydney.edu.au, stefanw@acfr.usyd.edu.au

Abstract

Problems in which autonomous agents perform simultaneous localisation and mapping (SLAM) are well explored in the literature. A related problem is rediscovery, in which the agent attempts to refine estimates of feature locations derived from previous SLAM surveys. However, most works in rediscovery assume high density of features. The major contribution of this work is a formulation of the sparse rediscovery problem. The applicability of current techniques in planning under uncertainty are subsequently considered with respect to the identified characteristics, specifically the high dimensionality of the problem in general and the low density of features in the surveys in the sparse problem. A solution to the sparse rediscovery problem is proposed and empirical results are presented and discussed.

1 Introduction

Simultaneous localisation and mapping (SLAM) is a well studied method for an autonomous vehicle to construct a map of the environment and incrementally refine pose estimates [Dissanayake *et al.*, 2001]. In SLAM, estimates of the state of the system describing the vehicle pose and the locations of features are represented by probability distributions. The distributions are updated to account for the uncertain state transitions due to imperfect control of the vehicle. However, updates to the estimates are also made to incorporate information from observations of the system. Through these explict representations of uncertainty and correlation SLAM allows refinements of the estimates.

A related problem is that of rediscovery, in which an agent attempts to refine estimates obtained from a previous SLAM survey. Of particular interest are autonomous underwater vehicle (AUV) mine hunting missions with high resolution sonars [Mandelert *et al.*, 2010], where the density of features (ie. mine-like objects) is low. In such problems, the scarcity of observations of features leads to large uncertainty associated with the estimates. Uncertainty in navigation and observations propagate, potentially resulting in highly inaccurate estimates for map feature locations. While more observations can be made by revisiting each feature, predicting the resulting changes in the estimates, in particular the uncertainty of the map, is non-trivial, due to the imperfect control and observations. Rather than simple revisits, the problem thus requires the agent to execute actions which have a higher probability of producing observations leading to the greatest reduction in the uncertainty of the map.

Given the uncertain changes in the estimates of the state, probablistic planning methods are well suited to the rediscovery problem. A popular approach to planning under uncertainty is the Markov Decision Process (MDP) [Sutton and Barto, 1998]. In a MDP framework the actions available to the autonomous agent lead to non-deterministic transitions in state. Given some model for how the transitions occur the agent attempts to act in fully-observed states to minimise some defined penalty or maximise some defined reward. The Partially Observable MDP (POMDP), an extension of the MDP with uncertainty in the states, similarly plans over a belief distribution for the current state of the system [Kaelbling *et al.*, 1998].

This work will show that, despite the maintenance in SLAM of a belief distribution for the true state of the system, the rediscovery problem may be approximated as an MDP working over the representation of estimates. Furthermore the analysis provided in this paper will show that exact solutions to the problem are still intractable. A Monte Carlo solution method is presented and compared against a round robin algorithm.

The remainder of this paper is organized as follows. Section 2 presents background information relating to the formulation of the SLAM problem, and provides an overview of MDP and POMDP methods. Section 3 outlines the formulation of the rediscovery problem as

a MDP and presents a Monte Carlo method for planning resurveys. Section 4 presents experimental results comparing the presented method against deterministic methods for map refinement. Finally Section 5 provides concluding remarks and directions for future work.

2 Background

2.1 SLAM

Simultaneous localisation and mapping (SLAM) is a method for accurate estimation of vehicle state (localisation) and the location of the map features (mapping). The pose of the vehicle, defined by its position and attitude, as well as the locations of the features are represented as probability distributions. A good introduction to basic SLAM techniques is given by [Durrant-Whyte and Bailey, 2006].

Although SLAM with moving map features has been considered by [Wang *et al.*, 2003], the following restricts itself to the better understood SLAM with stationary features. In general, the tuple $\langle \mathbf{v}_{v,k}, \mathbf{x}_m, \mathbf{a}_{k-1}, \mathbf{z}_k \rangle$ is defined at any time k where:

- $\mathbf{v}_{v,k}$ is the pose of the vehicle, including position and attitude;
- $\mathbf{x}_m = \left\{ \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{|\mathbf{x}_m|} \right\}$ is set of positions of landmarks;
- $\mathbf{a}_{k-1}$ is the action taken at time $k-1$; and
- $\mathbf{z}_k$ is the set of observations obtained.

Further, a history is maintained, consisting of $\mathbf{V}_{0:k}$, $\mathbf{Z}_{0:k}$, and $\mathbf{U}_{0:k-1}$, where:

- $\mathbf{V}_{0:k} = \{\mathbf{v}_{v,1}, \mathbf{v}_{v,2}, ..., \mathbf{v}_{v,k}\}$ is the set of vehicle pose estimates at every time step up to the present time k;
- $\mathbf{Z}_{0:k} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$ is the set of observations at every time step up to the present time k; and
- $\mathbf{U}_{0:k-1} = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{k-1}\}$ is the set of vehicle actions at every time step to the previous time $k-1$.

Solutions to the SLAM problem calculate the probability distribution

$$p\left(\mathbf{v}_{v,k}, \mathbf{x}_m \,|\mathbf{Z}_{0:k}, \mathbf{U}_{0:k-1}, \mathbf{v}_{v,0}\right)$$

representing the probability of the vehicle pose and map features being in specific states, conditioned on the history maintained. Updates to the belief distribution are made through the following equations:

Time Update:

$$p\left(\mathbf{v}_{v,k}, \mathbf{x}_m \,|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{v}_{v,0}\right) = \int p\left(\mathbf{v}_{v,k} \,|\mathbf{v}_{v,k-1}, \mathbf{a}_{k-1}\right) \times p\left(\mathbf{v}_{v,k-1}, \mathbf{x}_m \,|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-2}, \mathbf{v}_{v,0}\right) d\mathbf{v}_{v,k-1} \quad (1)$$

Observation Update:

$$p\left(\mathbf{v}_{v,k}, \mathbf{x}_m \,|\mathbf{Z}_{0:k}, \mathbf{U}_{0:k-1}, \mathbf{v}_{v,0}\right) = \frac{p\left(\mathbf{z}_k \,|\mathbf{v}_{v,k}, \mathbf{x}_m\right) p\left(\mathbf{v}_{v,k}, \mathbf{x}_m \,|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{v}_{v,0}\right)}{p\left(\mathbf{z}_k \,|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}\right)} \quad (2)$$

The most well understood SLAM solution uses the Extended Kalman Filter (EKF) and is known as EKF-SLAM [Dissanayake *et al.*, 2001]. The standard EKF time and observation updates [Thrun *et al.*, 2005] are applied to estimate the vehicle and map locations. The filter linearises the problem about the mean and covariance of the estimates, and represents the estimates at time k as multivariate Gaussian $\mathbf{p}(\mathbf{x}_k)$ with:

- Mean pose and location estimates:

$$\hat{\mathbf{x}}_{k|k} = \left(\hat{\mathbf{v}}_{v,k}^T, \hat{\mathbf{x}}_{m,k}^T\right)^T = E\left[\begin{array}{c} \mathbf{v}_{v,k} \\ \mathbf{x}_m \end{array} \,|\mathbf{Z}_{0:k} \right]$$

- Covariance of estimates,

$$\mathbf{P}_{k|k} = \left[\begin{array}{cc} \mathbf{P}_{vv} & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{mm} \end{array} \right]_{k|k}$$

where $\mathbf{x}_k = \left(\mathbf{v}_{v,k}^T, \mathbf{x}_m^T\right)^T$ is the true state.

Convergence properties are proved, and demonstrated in [Dissanayake *et al.*, 2001]. Intuitively, as more observations are made, the relative location estimates for landmarks should become more certain, as landmarks are observed from multiple directions, and correlated with each other. This decreasing uncertainty can be propagated back to obtain more accurate estimates for the vehicle location within the map. However, consistency of the filter is not guaranteed. Inconsistent behaviour, in which the estimate does not converge to the true state has been observed in [Julier and Uhlmann, 2001; Castellanos *et al.*, 2004]. Recent analysis includes [Huang and Dissanayake, 2007; Bailey *et al.*, 2006], the latter of which shows that inconsistency is much smaller if the error in the vehicle heading estimate is small. Further, a First Estimate Jacobian EKF (FEJ-EKF) was proposed in [Huang *et al.*, 2008] for improved consistency, based on an analysis of the observability matrix of the EKF updates.

2.2 MDPs

MDPs were first introduced by [Bellman, 1957], and were well explored in [Sutton and Barto, 1998]. An MDP is characterised by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$, where:

- $\mathcal{S} = \left\{ s_1, s_2, ..., s_{|\mathcal{S}|} \right\}$ is the set of possible states for the system;
- $\mathcal{A} = \{a_1, a_2, ..., a_{|\mathcal{A}|}\}$ is the set of actions which may be taken by the agent;

- $T(s_i, a, s_j) = p(s_j|s_i, a)$ is the set of transition probabilities for arriving at state s_j after taking action a in state s_i;
- $R(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ is the reward function, which gives the reward for taking action a in state s; and
- $\gamma \in [0, 1]$ is a discount factor used to bias the agent towards more immediate returns, as well as to ensure convergence of policy.

The implicit *Markov assumption*, where the state of the system at any time step is dependent only on the state of the system in the previous time step and the action taken, is represented by the formulation of the transition probabilities.

Consider an arbitrary policy $\pi(s) : \mathcal{S} \rightarrow (\mathcal{A})$, such that for every state s, the policy specifies the action $a_\pi(s) = \pi(s)$ to be taken. Then, for each state, the expected reward of each state may be written recursively as:

$$V_\pi(s) = R\left(s, \pi(s)\right) + \gamma \sum_{s' \in \mathcal{S}} T\left(s, \pi(s), s'\right) V_\pi(s') \quad (3)$$

Suppose the policy is to be improved on, for state s, by changing the current action at state s, $\pi(s)$, to a different action $\pi^+(s)$, while retaining the same policies. Then, for each action $a \in \mathcal{A}$, the expected reward obtained by executing a in state s is given as:

$$Q(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T\left(s, a, s'\right) V_\pi(s') \quad (4)$$

The policy π can thus be improved by replacing $\pi(s)$ with $\text{argmax}_a\, Q(s, a)$. Intuitively, the policy is expected to yield higher returns than the previous policy, as the new policy π^+ takes action equal to or better than policy π at state s, and the same actions elsewhere. As a corollary, the value function V^* for the optimal policy π^* is a solution to the *Bellman optimality equation*:

$$V^*(s) = \max_{a \in \mathcal{A}} \left(R(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s, a, s') V^*(s') \right) \quad (5)$$

This *policy update* can be followed by a value function update, in which the value functions defined by Equation 3 are recalculated. The process is known as *value iteration*. For problems with reachable *goal states*, such that agents in these state transition to the same state deterministically with no reward, and for problems with a non-zero discount factor, the policy converges to the optimum policy as the number of iterations approaches infinity [Sutton and Barto, 1998]. The detailed proofs for convergence are given by [Mendelssohn, 1982].

Particular formulations of the MDP also exist. In stochastic shortest path problems, the agent is required to move from one particular initial state to a specified goal state. In these problems, the agent is often penalised with a cost function, $C(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$, instead of encouraged with a reward function, to represent the cost of traversal in the map. Further, the discount factor is not included, as the costs of future traversal should not be diminished, and convergence is provided through the goal states. The agent is then expected to minimise future costs. However, by setting $\gamma = 1$ and $R(s, a) = -C(s, a)$ in the general MDP setup, the cost MDP can be obtained.

Although naive implementations of value iteration are sufficient for problems with small state spaces and yield exact solutions, current MDP solvers rely on other methods. Real-time dynamic programming (RTDP) [Barto *et al.*, 1995] and its variants [Bonet and Geffner, 2003; McMahan *et al.*, 2005; Smith and Simmons, 2006] are popular. The class of algorithms chooses action with the highest Q value at each point, and uses importance sampling for the resulting state, given transition probabilities and other characteristic.

The RTDP class of algorithms is *model-based*, in that an explicit model of transitions between states is required. Algorithms which do not require explicit models of transition for planning, classified as *model-free*, include Temporal Difference learning (TD) [Tesauro, 1995] and Q-learning [Watkins and Dayan, 1992]. These typically delay updates to the state value function until several actions have been made, and evaluate the resulting policy performances.

A key requirement of the MDP framework is that the state of the system is completely observable. A relaxation of this requirement leads to the Partially Observable MDP (POMDP) framework, for which [Kaelbling *et al.*, 1998] remains a seminal work.

2.3 POMDPs

A POMDP is an extension of an underlying MDP, and is characterised by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, \mathcal{Z}, \mathcal{O} \rangle$, such that

- the elements $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma >$ describe the underlying MDP;
- $\mathcal{Z} = \{z_1, z_2, ..., z_{|\mathcal{Z}|}\}$ is the set of observations the agent may make;
- $O(s_j, a, z_i) = p(z_i|s_j, a)$ is the observation probability function, describing the probability of making observation z_i, given action a was taken and the resultant state was s_j.

The Markov assumption in state transition is inherited from the underlying MDP. Further, the approximation of the environment by the agent is represented by the observation probability function.

The POMDP problem is solved over *belief states*, rather than explicit states s in the MDP. At every time step, the agent maintains a belief state b, a probability distribution for the state of the system, such that $b(s)$ represents the probability that the system is currently in state s. While *complete observability* allows MDPs to trivially update system state by using the current state s at every time step, the belief state b in POMDPs must be calculated. Given that action a was taken at previous belief state b, and the resulting observation was z, the update equation for the next belief state b' can be derived from laws of probability, and can be written:

$$b'(s') = \tau\left(b, a, z\right)\left(s'\right) = O(s', a, z)\frac{\sum_{s\in\mathcal{S}} T(s, a, s')b(s)}{p(z|b, a)} \tag{6}$$

In Equation 6, $p(z|b, a)$ is a normalising constant necessary for b' to remain a probability distribution. The term denotes the probability of observing z after executing action a in belief state b, and is calculated as

$$p(z|b, a) = \sum_{s'\in\mathcal{S}} O(s', a, s') \sum_{s\in\mathcal{S}} T(s, a, s')b(s) \tag{7}$$

Given methods for calculating belief state transition, the utility of taking specific actions in a belief state can be calculated. Again, a value function is used to denote the expected returns. However, the value functions are now calculated for belief states. Consider a policy $\pi(b) : \mathcal{B} \rightarrow \mathcal{A}$, mapping from $\mathcal{B}$, the space of belief states, to $\mathcal{A}$, the set of actions, dictating the actions taken for every belief state. Analogous to Equation 3, the value function for policy π can be found as

$$V_\pi(b) = R_B\left(b, \pi(b)\right) + \gamma \sum_{z\in\mathcal{Z}} p\left(z|b, \pi(b)\right) V_\pi\left(\tau\left(b, a, z\right)\right) \tag{8}$$

The function $R_B(b, a)$ represents the expected reward of taking action a in state b, and is found as

$$R_B(b, a) = \sum_{s\in\mathcal{S}} b(s)R(s, a) \tag{9}$$

With the above definitions, the POMDP problem can be reduced to a MDP over belief states, or a *belief MDP*. The belief transition function from b to $\tau\left(b, a, z\right)$ is given by $p(z|b, a)$, and the reward for performing action a under belief b is $R_B(b, a)$. The resulting MDP operates over continuous "states", as belief distributions are in general not discrete. The reduction of POMDPs to continuous MDPs is popular in the literature, and used by [Spaan and Vlassis, 2005; Kaelbling *et al.*, 1998; Ross *et al.*, 2008].

A variety of methods exist for solving POMDPs, mostly variations of value iteration. A witness algorithm to reduce the number of policies considered during value iteration was used in [Kaelbling *et al.*, 1998], arriving at approximately optimal solutions. RTDP-Bel [Bonet and Geffner, 2009] can be considered analogous to RTDP for MDPs, and samples actions at each time step, based on the expected utility of executing each action under the current belief. However, heuristic search methods, surveyed by [Ross *et al.*, 2008], are promising in terms of computational resources required and rapidity of convergence, even though the algorithms discussed only plan for locally optimal solutions and do not use the information gained over successive executions to arrive at solutions over the entire belief space.

3 Sparse resurvey problem

3.1 Characteristics

Where the density of features in the map is high, the vehicle is typically able to have better localisation and produce higher quality maps as more observations are received. If accurate maps are required, resurvey may be needed when few observations have been made, with low quality sensors. This may arise due to a large search area, in which a vehicle with limited sensor range has had to traverse long distances between features, and as such may have large uncertainty in pose when features are observed. Low quality maps may also be produced in surveys with relatively few features such that there are few chances for observations to be made.

As the problem is one of resurvey it would be appropriate to assume some prior estimate for the map exists from a previous SLAM survey with exhaustive coverage of the search area. The assumption that prior estimates exist for all features of interest in the map is thus explicitly made.

Consider now the initial state of the rediscovery problem. One formulation would include a completely new survey, with initial vehicle estimates decoupled from the map. Although correlations between vehicle pose and map would be lost, as SLAM allows a build up of correlations between vehicle pose estimates and map estimates [Durrant-Whyte and Bailey, 2006], correlations between features and vehicle pose can be slowly built up again. However, the proposed solution must also be flexible enough to be applicable where the resurvey takes place immediately after the initial survey, when the vehicle is still in the field. In such scenarios, a strong case may be made for the implementation of online algorithms, as there would be neither the time nor the resources for offline policy construction.

The rediscovery problem should thus have the following characteristics:

- Large survey area;
- Sparse features;
- Limited range sensors; and

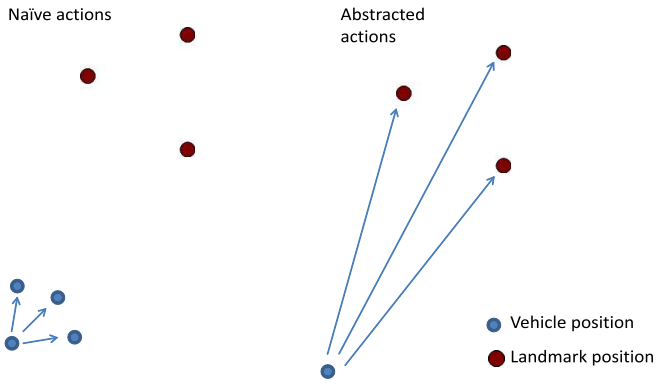

Figure 1: a) Naive formulation of the action: survey area too large relative to change in vehicle location between time steps, hence depth of policy search prohibitive; b) Abstraction of actions as the feature to visit next;

- Inherited map estimates from previous survey.

Under the stated conditions, the problem is to produce a survey strategy for rediscovery of map features and refinement of position estimate for the features.

3.2 Actions

The naive formulation of actions available to the vehicle would involve the control signal at the next time step. For example, in a non-holonomic vehicle model with constant speed this may be the steering angle for the next time step. However, given that the survey area is noted to be relatively large with sparse features it may require a large number of actions until any feature observations are made. As the changes in map estimates are derived only from observations, the potential depth of search required for different policies to yield distinct results may be prohibitive. Instead, the actions may be abstracted such that the autonomous agent chooses the feature to visit next. The difference between the two formulations is illustrated in Figure 1.

3.3 Costs

The formulation of the cost should reflect the objective of the problem. As stated, the problem aims to refine estimates of the map. As we ultimately wish to reduce the error in map estimates, one particular formulation may involve penalising each action according to the resulting error in the estimates of map features such that actions resulting in greater map error become discouraged. However, this particular formulation of the cost is unrealistic because the true location of the features are unavailable during the survey.

Alternatively, the agent could actively attempt to reduce uncertainty in the system. The motivation for a cost representing the uncertainty in the system arises from the fact that as observations are fused into the filter the estimates become more certain. In the limit, the feature location estimates become fully correlated. Thus, another cost commonly considered is the *differential entropy* of the estimate, given in [Darbellay and Vajda, 2000] for multivariate Gaussian state estimate $\mathbf{p}(\mathbf{x})$ with n features as:

$$H(\mathbf{p}(\mathbf{x})) = \frac{1}{2}\ln\left[(2\pi\mathrm{e})^{\mathrm{n}}\left|\mathbf{P}_{\mathrm{mm}}\right|\right]$$

This cost depends only on the estimates resulting from actions, and can thus be calculated from only the information available to the agent. The trace was not used as a cost because it does not reflect the information gain possible through stronger correlations between features. Note that for this formulation estimates with lower differential entropy have lower cost. Thus, actions leading to greater information gain, by the measure of entropy reduction, is encouraged.

3.4 Rediscovery as MDP

Given the above formulations, the problem can be described as a cost MDP/POMDP. It may be demonstrated, by considering the cost and transition functions, that the problem is a POMDP with "states" s described by the estimate distributions and the underlying true map positions and vehicle pose. However, simplifying approximations can be made to reduce the problem to an MDP.

Recall that the reward/cost for the MDP framework, and by extension the POMDP framework, $R(s,a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$, maps from the set of states and the set of actions. Consider now the "state" s. In planning for uncertainty reduction, the cost is necessarily a function of some property of the estimate distribution representing the uncertainty. For example, in the current case the cost is calculated from the covariance of the distribution. In general, as the reward/cost function in the MDP framework is some function of the set of s over which planning occurs, the set of s in planning for uncertainty reduction must be be partially described by *the estimate distribution itself*, or *some property of the estimate distribution representing uncertainty*. With the cost formulation adopted in this work, s is thus partially described by either the Gaussian distribution for vehicle pose and map estimates, or the covariance of the map estimates.

Suppose now that s for the uncertainty reduction MDP/POMDP is fully described by the estimate distribution. That is, the "states" over which planning occurs are fully described as probability distributions for vehicle pose and landmark locations, in the form:

$$\mathbf{p}_{j|j} = p\left(\mathbf{v}_{v,j}, \mathbf{x}_m \left| \mathbf{Z}_{0:j}, \mathbf{U}_{0:j-1}, \mathbf{v}_{v,0}\right.\right)$$

Consider the case when only one action $\mathbf{a}_{k-1}$ has been taken, while the estimate distribution is $\mathbf{p}_{k-1|k-1}$. From Equations 1 and 2, the posterior $\mathbf{p}_{k|k}$ can be calculated, given observation $\mathbf{z}_k$. The transition function $T(\mathbf{p}_{k-1|k-1}, \mathbf{a}_{k-1}, \mathbf{p}^*_k)$, giving the probability of transition to particular

$$\mathbf{p}^*_{k|k} = p\left(\mathbf{v}_{v,k}, \mathbf{x}_m \,|\mathbf{Z}_{0:k-1}, \mathbf{z}^*_k, \mathbf{U}_{0:k-1}, \mathbf{v}_{v,0}\right)$$

can be thus be written:

$$T(\mathbf{p}_{k-1|k-1}, \mathbf{a}_{k-1}, \mathbf{p}^*_{k|k}) = p\left(\mathbf{z}^*_k \left|\mathbf{v}^t_{v,k}, \mathbf{x}^t_m\right.\right) \tag{10}$$

for $p\left(\mathbf{z}^*_k \left|\mathbf{v}^t_{v,k}, \mathbf{x}^t_m\right.\right)$ the probability of obtaining $\mathbf{z}^*_k$, with true vehicle pose $\mathbf{v}^t_{v,k}$ after the action, and true map locations $\mathbf{x}^t_m$.

The transition function required for the MDP and POMDP frameworks is thus a function of the true vehicle pose and map locations. The probability of transition from the current estimate to the next is not determined only by the current estimate. Using only the estimates as "states" thus does not meet the Markov assumption, and the "state" can not be completely described by the estimates.

Consider now if the "states" s were composed of the estimate distribution, and the truth values for vehicle and map locations. Specifically, let s_{k-1} be composed of the estimate $\mathbf{p}_{k-1|k-1}$ and the truth $\mathbf{x}^t_{k-1}$. The cost function can thus be calculated for each "state" from the estimate distribution as outlined above. Consider now the transition between states. The estimates can be updated as outlined above. Assuming constant map locations, only the true vehicle pose changes, and can be described by the vehicle model. Then, given that the true vehicle pose transitions according to the vehicle model $p\left(\mathbf{v}_{v,k} \,|\mathbf{v}_{v,k-1}, \mathbf{a}_{k-1}\right)$ and the estimate distribution transitions as in Equation 10, the transition function from state s_{k-1} to state s^*_k may be written:

$$\begin{aligned} &T(s_{k-1}, \mathbf{a}_{k-1}, s^*_k) = \\ &\quad p\left(\mathbf{z}^*_k \left|\mathbf{v}^{t*}_{v,k}, \mathbf{x}^t_m\right.\right) p\left(\mathbf{v}^{t*}_{v,k} \left|\mathbf{v}^t_{v,k-1}, \mathbf{a}_{k-1}\right.\right) \end{aligned} \tag{11}$$

where s^*_k composed of $\mathbf{p}^*_{k|k}$ and $\mathbf{x}^{t*}_k$ is the resulting true state. This formulation of the "states" in the MDP framework is thus sufficient for both the calculation of the reward/cost functions, and the transition probabilities.

From the view of the planning agent, s has two components:

- the fully observable component, the estimate for the true vehicle pose and landmark positions; and
- the partially observable component, the true vehicle pose and landmark positions.

The problem is thus a POMDP over the composite state.

However, a simplifying approximation was made to reduce the problem back to a MDP. Note that the partially observable component, the true pose and landmark positions, is approximated by the fully observable component, the estimate distributions. Then, Equation 10 may be approximated as:

$$\begin{aligned} &T_{approx}(\mathbf{p}_{k-1|k-1}, \mathbf{a}_{k-1}, \mathbf{p}^*_{k|k}) = \\ &\quad \int \left(p\left(\mathbf{v}_{v,k}, \mathbf{x}_m \,|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{v}_{v,0}\right) \times \right. \\ &\qquad \left. p\left(\mathbf{z}^*_k \,|\mathbf{v}_{v,k}, \mathbf{x}_m\right)\right) d\mathbf{v}_{v,k}, \mathbf{x}_m \end{aligned} \tag{12}$$

for $p\left(\mathbf{v}_{v,k}, \mathbf{x}_m \,|\mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{v}_{v,0}\right)$, the posterior of the Time Update in Equation 1.

It may be noted that the approximated transition function can be calculated using only the estimate distribution maintained by the agent. Thus with this approximation, both the reward/cost function and the transition probability are functions only of the estimate distribution. Given the approximation, the underlying MDP/POMDP "states" can be fully described by the estimate distributions. As the estimate distribution is fully observable, the approximation reduces the problem to a MDP.

However, an exact solution is still difficult. There is no direct method for calculating transition between estimates given the action abstractions. This is because the exact control actions $\mathbf{a}_{k-1}$ are not known beforehand, as these are calculated by the path following controller at each time interval based on updated vehicle position estimates. Further, the problems typically have high dimensionality. For example, in a complete description of multivariate Gaussians estimates for a 2D resurvey problem with two map features and vehicle pose described by position and heading there are 7 variables associated with estimate means and 28 variables associated with the 7×7 covariance matrix. The problem must plan over 35 continuous dimensions. Thus, even small problems may become intractable. Given the unknown transitions and the intractable nature of the problem, Monte Carlo methods seem appropriate.

3.5 Proposed solution

The solution implemented in this work uses Monte Carlo sampling for evaluating the policy with one-step look-ahead, as described in Algorithm 1. Sampling is performed at the start of the resurvey, each time the agent has reached its waypoint.

One of the reasons for the implementation of Monte Carlo sampling is the difficulty in determining transitions between estimates. Due to large uncertainties in the map, as well as control and observation noise, it is difficult to estimate the observations received by executing any action. Thus, Monte Carlo sampling would

allow transitions to be simulated. Another motivation is the intractability of the problem. Exact solutions over the entire problem space would be computationally prohibitive and unrealistic for real-time implementations. However, Monte Carlo algorithms restrict examination to the most probable successor states as these will be more likely to occur in sampling. This is similar to the idea behind RTDP in which the value functions are approximated for the most relevant states. Unlike RTDP, no learning occurs in the current implementation. However, due to the monotonically decreasing determinant in the covariance of features [Dissanayake *et al.*, 2001] the particular estimates are unlikely to recur. Thus, no benefits are gained for learning given that planning is online.

Algorithm 1 Monte Carlo rediscovery planning

Input:

- **Current estimate $\mathbf{p}(\mathbf{x})$;**
- **Available actions $\mathcal{A}$= $\{a_1, a_2, ..., a_{|\mathcal{A}|}\}$;**
- **Number of map samples m;**

Output:

- **Best action a;**

1. **while** $m > 0$
   (a) **sample true state $\mathbf{x_t}$ from $\mathbf{p}(\mathbf{x})$**
   (b) $i = 1$
   (c) **while** $i \leq |\mathcal{A}|$
      i. **Simulate action a_i, and let the resulting estimate be $\mathbf{p}(\mathbf{x}^*)$**
      ii. **Let** $c_i(m) = H(\mathbf{p}(\mathbf{x}^*))$
      iii. $i = i + 1$
   (d) $m = m - 1$
2. **Calculate** $\bar{c}_i = mean(c_i), i = 1, 2, ..., |\mathcal{A}|$
3. **Choose** $a = a_j$ **where** $\bar{c}_j = \min \bar{c}_i, i = 1, 2, ..., |\mathcal{A}|$

3.6 Related work

There has been some research into SLAM-informed planning. However, there are differences between the work presented here and previous work, the most significant being the scarcity of features in the current problem.

A FastSLAM implementation with occupancy grids was used in [Stachniss *et al.*, 2005] to balance map exploration with map accuracy. The problem allowed a solution to be found with what is reportedly the equivalent of a one-step POMDP even though the reward was also calculated according to the properties of the estimates rather than the underlying true states. Further, due to the sparseness of the map in the current work the number of observations are relatively small and the EKF-SLAM is preferred. Particle-filter based methods may quickly become over-confident [Bailey *et al.*, 2006], such that actions chosen may cause the agent to miss the landmarks entirely by underestimating the uncertainty of estimates during simulations.

In [Kollar and Roy, 2008], EKF-SLAM was used in conjuction with Policy Search Dynamic Programming. The resulting algorithm allowed the robot to plan paths such that map coverage is maximised and the entropy of the map is minimised. The policy constructed is powerful in that it approximated the optimal policy for a large set of estimates, as learning occurred over randomly sampled starting estimates. However, this method of learning is expensive and hence necessarily offline. Further, it is not required in the rediscovery problem because the restricted starting estimate in the problem allows policy search to occur over the most reachable estimates, as outlined above. Lastly, learning over a large set of estimates was possible in [Kollar and Roy, 2008] because of the relatively small size of the map, as smaller numbers of samples were able to approximate the most relevant estimates.

A combination of SLAM and finite and receding horizon look-aheads was used in [Martinez-cantin *et al.*, 2007] to plan paths which gave the greatest reduction in feature uncertainty. However, the main difference to that presented here, as with all the others, was the size, and sparsity of the problem. Where the action was set up as waypoints within short distances of the current vehicle position estimates in the previous work, the actions are abstracted in this work by necessity. This eliminates the local minima encountered in the [Martinez-cantin *et al.*, 2007], as each action takes the vehicle to the vicinity of a feature and thereby forces exploration.

4 Simulation results

In the trials, five features were randomly generated with positions $\mathbf{x_i}$= (x_i, y_i) for $i = 1, 2, 3, 4, 5$, such that $x_i, y_i \in [-200, 200]$. An exhaustive SLAM survey of the area was performed to provide a first estimate as seen in Figure 2a). The vehicle was assumed to be a non-holonomic rear steering vehicle moving at a nominally constant speed with heading observations, making range-bearing observations, with:

- Nominal speed 2.6m/s;
- Steering angle error (1σ) of 1 degrees;
- Speed error (1σ) of 0.3m/s;
- Vehicle heading observation error (1σ) of 5 degrees;
- Observation range error (1σ) of 1m;
- Observation bearing error (1σ) of 3 degrees; and
- Maximum sensor range of 30m;

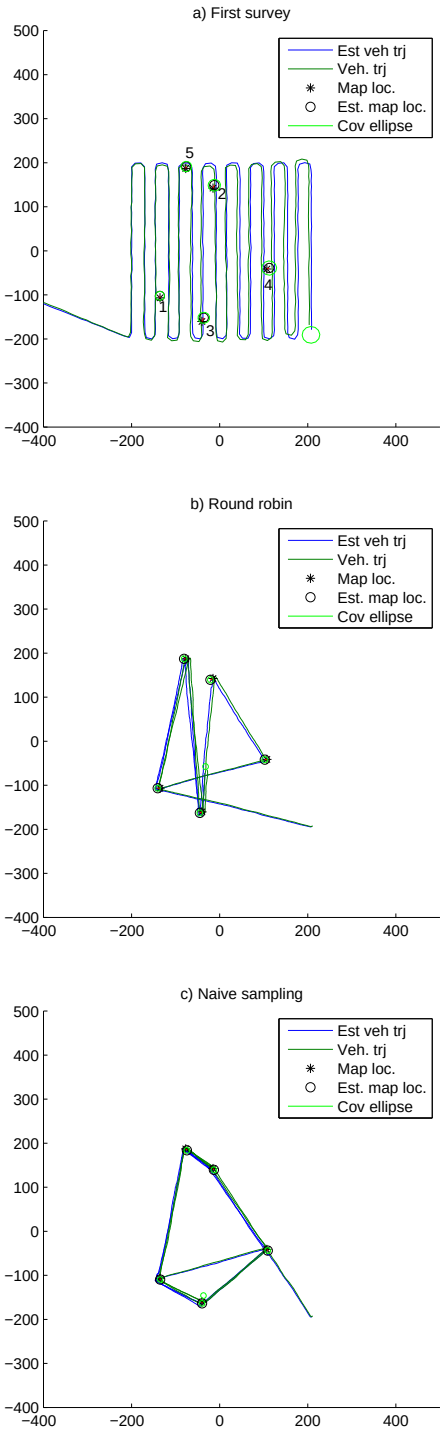

Figure 2: The results from: a) the first survey; b) example resurvey based on round robin observations; and c) Monte Carlo resurvey. Coverage of features depends on expected entropy reduction in the Monte Carlo resurvey. The round robin resurvey plan enforces even coverage.

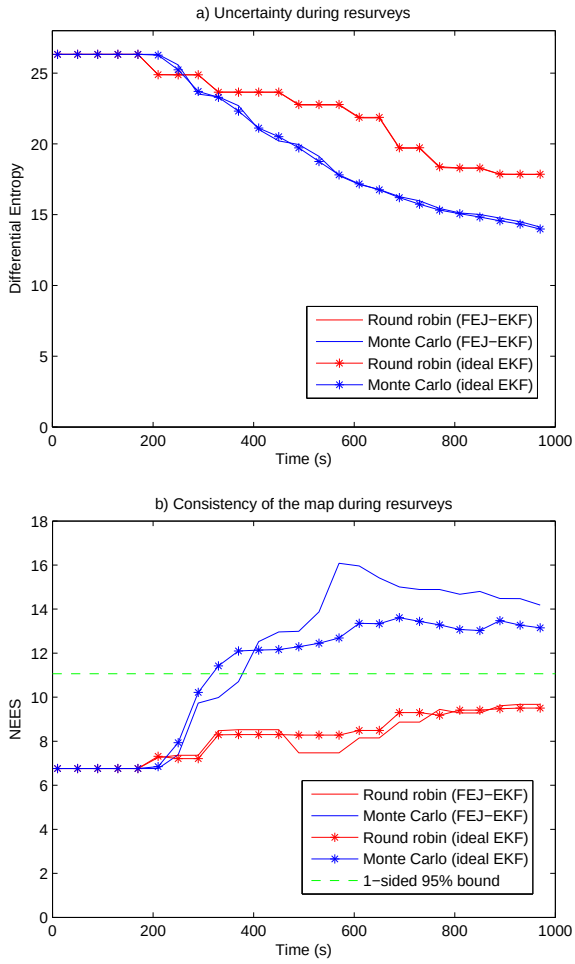

Figure 3: a) shows the changes in differential entropy averaged over 50 trials for the round robin algorithm (red) and Monte Carlo sampling (blue). Both the FEJ-EKF filter and the ideal EKF filter were considered. The Monte Carlo methods give greater reductions in uncertainty as measured by entropy, because actively planning for entropy reduction is performed. Note that entropy reduction is similar regardless of linearisation; b) shows the NEES of the resurveys at selected times, averaged over 50 trials. While inconsistency does not seem to be catastrophic, the Monte Carlo trials are mostly inconsistent, as the averaged NEES were mostly outside of the 1-sided 95% bound. An argument may thus be made for alternative filters.

	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
Initial survey	4.49	7.32	9.39	5.61	4.52
Round robin (FEJ-EKF)	2.80 ± 1.55	4.00 ± 2.30	4.81 ± 2.44	3.62 ± 2.36	3.31 ± 1.95
Round robin (ideal EKF)	2.82 ± 1.50	3.82 ± 1.87	4.40 ± 2.19	3.38 ± 1.69	3.19 ± 1.66
Monte Carlo (FEJ-EKF)	3.34 ± 1.71	3.30 ± 1.41	3.68 ± 1.68	3.33 ± 1.95	3.22 ± 1.45
Monte Carlo (ideal EKF)	3.10 ± 1.67	3.15 ± 1.97	3.46 ± 1.93	3.07 ± 1.89	3.16 ± 1.95

Table 1: Average error (m) in map location estimates, resulting from the initial run, the round robin resurveys and the Monte Carlo resurveys. The resurvey errors were averaged over 50 trials.

The resulting estimate of map and vehicle locations was important as it captured the correlations between features and the vehicle, which is difficult to generate without simulation. Two resurvey planning methods were then compared: a round robin approach which revisits features in a set pattern, and the Monte Carlo method. The implementation used FEJ-EKF SLAM, with regular heading observations, to reduce the effect of inconsistency. It may be noted from [Huang and Dissanayake, 2007] that linearising about estimates rather than true states may lead to degraded performance. Thus, the results were also compared against those obtained from resurveys with ideal EKF linearisations, in which linearisation occurs about the true states.

The resurveys had the same vehicle and observation characteristics as before, except the maximum sensor range was reduced to 15m. The vehicle is initially uncorrelated with the map. Each method was allowed 2000 time steps of 0.5 seconds in each trial, and results were collected over 50 trials. Example plans are shown in Figure 2.

In the round robin resurveys, the vehicle does not adopt adaptive planning and even coverage is enforced. This leads to less entropy reduction over time than the Monte Carlo methods, as seen in Figure 3a), as the agent does not actively plan for entropy reduction. However, the Monte Carlo methods have approximations of the expected changes in estimate resulting from each action and the agent is able to choose actions which lead to larger reduction in entropy. It may be noted that similar entropy reduction is observed regardless of linearisation.

Consistency of map estimates was measured by the Normalised Estimation Error Squared (NEES) metric [Bar-Shalom *et al.*, 2001]. The averaged NEES of the system for the resurveys are shown in Figure 3b). As there were 10 degrees of freedom (2 for each landmark location), the expected distribution with 50 trials was $\frac{1}{50}\chi^2\,(10 \times 50)$. As the one-sided 95% interval is 11.0625, the Monte Carlo resurveys seemed to suffer from inconsistency. The entropy reductions were thus not matched by corresponding decreases in map estimate errors. As inconsistency also occurs with ideal EKF linearisation, the problem may be inherent to EKF and alternative filters may be necessary in further investigations.

The resulting errors for feature estimate averaged over 50 resurvey simulations are shown in Table 1. For the same filters, the Monte Carlo approaches yield lower map error and standard deviation than those derived from the round robin method except for Feature 1. The Monte Carlo method thus consistently leads to greater overall improvements in map feature estimates, despite suffering from inconsistency.

5 Concluding remarks

This work has presented a solution to the problem of resurveying features under uncertainty. The characteristics of sparse resurvey problems have been examined and it was shown that the full formulation of the problem is a POMDP over the estimates and the true states. It has also been shown that approximations can be made to reduce the problem to a MDP in the estimates. A solution with abstracted actions and based on Monte Carlo sampling of transitions is presented. A comparison between the Monte Carlo approach and a round robin approach, representing an arbitrary revisit plan, shows that the Monte Carlo approach leads to greater reductions in differential entropy and errors in estimated map location. Although the results are encouraging, the inconsistency of the filter has led to map accuracy results which may be improved. However, this should not detract from the resurvey planning method as implementation of filters which address inconsistency is expected to lead to correspondingly more accurate map estimates.

References

[Bailey *et al.*, 2006] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the EKF-SLAM algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3562–3568, 2006.

[Bar-Shalom *et al.*, 2001] Y. Bar-Shalom, X.R. Li, X.R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation.* Wiley-Interscience, 2001.

[Barto *et al.*, 1995] A.G. Barto, S.J. Bradtke, and S.P. Singh. Learning to act using real-time dynamic programming. Technical Report 1-2, 1995.

[Bellman, 1957] R. Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6:679–684, 1957.

[Bonet and Geffner, 2003] B. Bonet and H. Geffner. Labeled RTDP: Improving the convergence of real-time dynamic programming. In *Internation Conference for Automated Planning and Scheduling*, volume 3, pages 12–21, 2003.

[Bonet and Geffner, 2009] B. Bonet and H. Geffner. Solving POMDPs: RTDP-bel vs. point-based algorithms. In *International Joint Conference on Artifical intelligence*, pages 1641–1646, 2009.

[Castellanos *et al.*, 2004] J.A. Castellanos, J. Neira, and J.D. Tardos. Limits to the consistency of EKF-based SLAM. In *IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.

[Darbellay and Vajda, 2000] G. A. Darbellay and I. Vajda. Entropy expressions for multivariate continuous distributions. In *IEEE Transactions on Information Theory*, pages 709–712, 2000.

[Dissanayake *et al.*, 2001] M. Dissanayake, P. Newman, S. Clark, HF Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[Durrant-Whyte and Bailey, 2006] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.

[Huang and Dissanayake, 2007] S. Huang and G. Dissanayake. Convergence and consistency analysis for extended Kalman filter based SLAM. *IEEE Transactions on Robotics*, 23(5):1036–1049, 2007.

[Huang *et al.*, 2008] G.P. Huang, A.I. Mourikis, and S.I. Roumeliotis. Analysis and improvement of the consistency of extended Kalman filter based SLAM. In *IEEE International Conference on Robotics and Automation*, pages 473–479, 2008.

[Julier and Uhlmann, 2001] S.J. Julier and J.K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *IEEE International Conference on Robotics and Automation*, pages 4238–4243, 2001.

[Kaelbling *et al.*, 1998] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[Kollar and Roy, 2008] T. Kollar and N. Roy. Efficient optimization of information-theoretic exploration in SLAM. In *National Conference on Artificial Intelligence*, pages 1369–1375, 2008.

[Mandelert *et al.*, 2010] N. Mandelert, J. Ferrand, and P. Cooper. Autonomy for operational MCM AUVs, based on high resolution sonar. In *Oceans 2010*, 2010.

[Martinez-cantin *et al.*, 2007] R. Martinez-cantin, O De Freitas, A. Doucet, and J. A. Castellanos. Active policy learning for robot planning and exploration under uncertainty. In *Robotics: Science and Systems*, 2007.

[McMahan *et al.*, 2005] H. B. McMahan, M. Likhachev, and G. Gordon. Bounded real-time dynamic programming: RTDP with monotone upper bounds and performance guarantees. In *International Conference on Machine Learning*, pages 569–576, 2005.

[Mendelssohn, 1982] R. Mendelssohn. An iterative aggregation procedure for Markov decision processes. *Operations Research*, 30(1):62–73, 1982.

[Ross *et al.*, 2008] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32(1):663–704, 2008.

[Smith and Simmons, 2006] T. Smith and R. Simmons. Focused real-time dynamic programming for MDPs: Squeezing more out of a heuristic. In *National Conference on Artificial Intelligence*, volume 2, page 1227, 2006.

[Spaan and Vlassis, 2005] M.T.J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24(1):195–220, 2005.

[Stachniss *et al.*, 2005] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems*, pages 65–72, 2005.

[Sutton and Barto, 1998] R.S. Sutton and A.G. Barto. *Reinforcement learning*. MIT Press, 1998.

[Tesauro, 1995] G. Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press, 2005.

[Wang *et al.*, 2003] C.C. Wang, C. Thorpe, and S. Thrun. On-line simultaneous localisation and mapping with detection and tracking of moving objects. In *IEEE International Conference on Robotics and Automation*, pages 2918–2924, 2003.

[Watkins and Dayan, 1992] C.J.C.H. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.