

# Multi-Robot Marginal-SLAM

**Ruben Martinez-Cantin** and **José A. Castellanos**  
Dept. Informática e Ingeniería de Sistemas  
Inst. de Investigación en Ingeniería de Aragón (I3A)  
University of Zaragoza  
rmcantin@unizar.es

**Nando de Freitas**  
Dept. of Computer Science  
University of British Columbia  
nando@cs.ubc.ca

## Abstract

This paper has two goals. First, it expands the presentation of the marginal particle filter for SLAM proposed recently in [Martinez-Cantin *et al.*, 2006]. In particular, it presents detailed pseudo-code to enable practitioners to implement the algorithm easily. Second, it proposes an extension to the multi-robot setting. In the marginal representation, the robots share a common map and their locations are independent given this map. The robot’s relative locations with respect to each other are assumed to be unknown. The multi-robot Marginal-Slam algorithm estimates these transformations of coordinates between the robots to produce a common global map.

## 1 Introduction

For many years, Simultaneous Localization and Mapping (SLAM) has occupied the center-stage in robotics research [Durrant-Whyte and Bailey, 2006; Thrun, 2002]. The introduction of particle filters (PFs) gave researchers the power and flexibility to handle nonlinearity and non-Gaussian distributions routinely [Fox *et al.*, 2001]. Moreover, it enabled researchers to exploit conditional independence, using the Rao-Blackwellized particle filtering (RBPF) variance reduction technique, to obtain more efficient Monte Carlo schemes. RBPFs were applied to dynamic maps [Doucet *et al.*, 2000] and subsequently to static maps with the celebrated FastSLAM algorithm [Montemerlo *et al.*, 2003]. The application of RBPF to dynamic maps is only sensible inasmuch as one has a good model to describe the evolution of the *dynamic* map. On the other hand, the application of RBPF to static maps has come into question. It has become popular knowledge that the approach can diverge [Bailey *et al.*, 2006]. In loose terms, learning static variables (the map) by conditioning on increasing histories of the state variables results in an accumulation of Monte Carlo errors and explosion of variance. We will elaborate on this later on and present further arguments. Heuristic approaches to ameliorate the situation [Stachniss *et al.*, 2005; Elinas *et al.*, 2006] have been proposed, but these do not solve the fundamental problem at hand. The problem of PF divergence resulting from learning fixed variables by conditioning

on increasing paths was initially pointed out in [Andrieu *et al.*, 1999].

The same concerns apply to the multi-robot setting. The problem is further exacerbated since we typically do not know the relative position of one robot with respect to the others. It is important to estimate this quantity in order to produce a global map. Some authors simplify the problem by only carrying out the map merging step when two robots make “eye-contact” [Howard, 2005]. In general, however, the only information available is the data association between common features in both maps. In this setting, some authors have adopted Monte-Carlo localization using joint path sampling [Fox *et al.*, 2006]. However, as we have argued earlier, this approach could easily result in degeneracy of the filter. We refer the interested reader to [Fox *et al.*, 2006] for background on the multi-robot SLAM problem.

This paper presents a marginal filtering approach, where the Monte Carlo integration with respect to the robot state (pose) happens in the marginal space. The approach *treats static maps as parameters*, which by necessity are learned using maximum likelihood (ML) or maximum a posteriori inference. The idea of treating maps as parameters is not new. It has been central to the incremental ML method [Thrun, 1993]. However, this method resorts to an ML estimate of the state and hence fails to manage the uncertainty in the robot state properly, as elaborated in [Thrun, 2002]. A variation based on learning the distribution of the robot states and using an ML estimator of the map as a function of the existing and *growing state trajectories* was proposed in [Thrun, 2001]. This alternative unfortunately suffers from the same consistency problems as FastSLAM. Various EM techniques using forward-backward state smoothing and map estimation in the M step have also been proposed; see [Thrun, 2002] for a survey. However, these methods only apply when learning the map off-line. Moreover, it is only very recently that particle smoothing has become feasible [Klaas *et al.*, 2006]. (As an aside, we note that two-filter smoothers do better than forward-backward smoothers in the PF context.)

In the marginal approach to SLAM, we are able to compute the filter derivative (derivative of the filtering distribution) on-line. This is extremely crucial for on-line static parameter estimation, but has only become possible very recently following new advances in particle simulation [Klaas *et al.*, 2005; Poyadjis *et al.*, 2005a]. Another advantage of working on the

marginal space is that we can assume independency of the robot locations given the shared map. Therefore, an independent sample set can be used for each robot; thereby reducing the dimensionality of the sampling space. That is, the filter derivatives and mapping updates are independent despite the fact that the map parameters are shared.

Sections 2 and 3 of this paper are a review of the method proposed in [Martinez-Cantin *et al.*, 2006]. The new extension to multiple robots and new experiments are presented in Sections 4 and 5.

## 2 Problem Formulation

We present a general formulation of the problem that is applicable to both feature-based maps and grid-based maps. A more specific and detailed model appears in the experimental section. The unknown robot pose (location and heading),  $\mathbf{x}_t \in \mathcal{X}$ , is modelled as a Markov process of initial distribution  $p(\mathbf{x}_1)$  and transition prior  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ . The observations,  $\mathbf{y}_t \in \mathcal{Y}$ , are assumed to be conditionally independent given the process  $\{\mathbf{x}_t\}$  and of marginal distribution  $p_\theta(\mathbf{y}_t | \mathbf{x}_t)$ , where  $\theta$  is a vector describing the elements of the map. Hence, the model consists of the following two distributions:

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{x}_{t-1}) \\ p_\theta(\mathbf{y}_t | \mathbf{x}_t) \quad t \geq 1. \end{aligned}$$

We denote by  $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$  and  $\mathbf{y}_{1:t} \triangleq \{\mathbf{y}_1, \dots, \mathbf{y}_t\}$ , respectively, the robot state and the observations up to time  $t$ , and define  $p(\mathbf{x}_1 | \mathbf{x}_0) \triangleq p(\mathbf{x}_1)$  for notational convenience.

In this formulation, we have used  $\theta$  to describe the map only. However,  $\theta$  could be used to include other parameters in the transition and measurement models, as well as, to include data association variables [Thrun, 2002]. For simplicity of presentation, we will assume that the associations are given.

Our objective is to compute sequentially in time the *filtering distribution*  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  and point estimates of the map  $\theta$ .

## 3 Particle Methods

### 3.1 The Joint Path Space Approach

In classical Rao-Blackwellized particle filtering, one first notes the following decomposition of the joint posterior:

$$p(\theta, \mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = p(\theta | \mathbf{x}_{1:t}, \mathbf{y}_{1:t}) p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}).$$

Consequently, given the state path  $\mathbf{x}_{1:t}$ , we can solve for the map  $\theta$  analytically. This leaves us with only having to carry out particle filtering to compute the posterior distribution of the robot state  $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$ .

If we had a set of samples (or *particles*)  $\{\mathbf{x}_t^{(i)}\}_{i=1}^N$  from  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$ , we could approximate the distribution with the Monte Carlo estimate

$$\hat{p}(d\mathbf{x}_t | \mathbf{y}_{1:t}) = \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_t^{(i)}}(d\mathbf{x}_t),$$

where  $\delta_{\mathbf{x}_t^{(i)}}(d\mathbf{x}_t)$  denotes the delta Dirac function. This estimate converges almost surely to the true expectation as  $N$

goes to infinity. Unfortunately, one cannot easily sample from the marginal distribution  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  directly. Instead, we draw particles from  $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$  and samples  $\mathbf{x}_{1:t-1}$  are ignored. This is a valid way to draw samples from a marginal distribution and is at the core of most Monte Carlo statistical methods. The unknown normalizing constant precludes us from sampling directly from the posterior. Instead, we draw samples from a proposal distribution  $q$  and weight the particles according to the following importance ratio:

$$w_t(\mathbf{x}_{1:t}) = \frac{p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}$$

The proposal distribution is constructed sequentially

$$q(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = q(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) q(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1})$$

and, hence, the importance weights can be updated recursively in time

$$w_t(\mathbf{x}_{1:t}) = \frac{p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) q(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1})} w_{t-1}(\mathbf{x}_{1:t-1}). \quad (1)$$

Given a set of  $N$  particles  $\{\mathbf{x}_{1:t-1}^{(i)}\}_{i=1}^N$ , we obtain a set of particles  $\{\mathbf{x}_{1:t}^{(i)}\}_{i=1}^N$  by sampling from  $q(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1}^{(i)})$  and applying the weights of equation (1).

The familiar particle filtering equations for this model are obtained by remarking that

$$p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) \propto p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = \prod_{k=1}^t p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}),$$

given which, equation (1) becomes

$$\tilde{w}_t^{(i)} \propto \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)} | \mathbf{y}_t, \mathbf{x}_{t-1}^{(i)})} \tilde{w}_{t-1}^{(i)}.$$

This iterative scheme produces a weighted measure  $\{\mathbf{x}_{1:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ , where  $w_t^{(i)} = \tilde{w}_t^{(i)} / \sum_j \tilde{w}_t^{(j)}$ , and is known as Sequential Importance Sampling (SIS).

It has been proved [Doucet, 1998] that *the variance of the importance weights in SIS increases over time*. This causes most particles to have very small probability. A common strategy to solve this *degeneracy*, consists of using a resampling step (SIR) after updating the weights to replicate samples with high probability and prune those with negligible weight [Doucet *et al.*, 2001].

This is the procedure in common use by practitioners. It can be deceptive: although only the state  $\mathbf{x}_t$  is being updated every round, the algorithm is nonetheless importance sampling in the *growing joint path space*  $\mathcal{X}^t$ .

Formally, the resampling step should be done along the full path  $\{\mathbf{x}_{1:t}^{(i)}, w_{1:t}^{(i)}\}_{i=1}^N$ . Since dynamic systems forget the past exponentially fast, several authors carry out resampling over the marginal space  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ . This would be fine if, for example, we were interested in tracking dynamic maps.

Static parameter estimation and model selection do not necessarily exhibit the exponential forgetting behavior. For example, static maps depend on the whole state trajectory.

Resampling over the joint path space is guaranteed to deplete the past in finite time. Alternatively, resampling from the marginal space still leaves us with an accumulation of Monte Carlo errors over time. Some implementations have introduced artificial dynamics or Markov chain Monte Carlo (MCMC) rejuvenation steps to reduce the severity of the problem, but these approaches do not overcome the problem.

In conclusion, whether we resample or not, learning the map as a function of a growing path in Monte Carlo simulation is a bad idea.

The same degeneracy problem arises if we try to obtain estimates of the filter derivative  $\nabla_{\theta} p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t})$  for recursive (online) map estimation. To see this, let  $\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$  denote the gradient vector of the path posterior with respect to the map. Then, we have

$$\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = \frac{\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$$

and, consequently the filter derivative, necessary for online map learning, is given by:

$$\nabla_{\theta} p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) = \int_{\mathcal{X}^{t-1}} \frac{\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) d\mathbf{x}_{1:t-1} \quad (2)$$

This equation says that when using standard particle filters to approximate the filter derivative we are implicitly carrying out importance sampling on a vast growing space with proposal  $p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$  and weight  $\frac{\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}$ . This should be enough reason to call for a new approach. Yet, the problem is even worse.

The filter derivative is a signed-measure, and not a standard probability measure. It consists of positive and negative functions over disjoint parts of the state space and it sums to zero over the entire state space. A serious problem, when carrying out classical particle filtering to estimate this signed-measure, is that particles with positive and negative weights will cancel each other, say, in parts of the space where the derivative is close to zero. This is wasteful and statistically harmful. See Figure 1 of [Poyadjis *et al.*, 2005a] for a beautiful depiction of this problem.

The technique presented in the following section overcomes these deficiencies.

## 3.2 The Marginal Space Approach

### Marginal Particle Filtering

To eliminate the problems discussed in the previous section, we will perform particle filtering *directly* on the marginal distribution  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  instead of on the joint space [Klaas *et al.*, 2005; Poyadjis *et al.*, 2005b; 2005a]. To do so, we begin by noting that the predictive density can be obtained by marginalization:

$$p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \quad (3)$$

To simplify the exposition later on, we introduce the following notation [Poyadjis *et al.*, 2005a]:

$$p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) \triangleq \frac{\xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t})}{\int \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t} \quad (4)$$

Using equation (3) and Bayes rule, the unnormalized filtering distribution can be expanded as follows:

$$\xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) = p_{\theta}(\mathbf{y}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}.$$

The integral in equation (3) is generally not solvable analytically, but since we have a particle approximation of  $p(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$  (namely,  $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}$ ), we can approximate (3) as the weighted kernel density estimate

$$\widehat{p}_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t-1}) = \sum_{j=1}^N w_{t-1}^{(j)} p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(j)}).$$

While we are free to choose any proposal distribution that has appropriate support, it is convenient to assume that the marginal proposal takes a similar form, namely

$$q_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) = \sum_{j=1}^N w_{t-1}^{(j)} q_{\theta}(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1}^{(j)}). \quad (5)$$

We can easily draw particles from this proposal using multinomial or stratified sampling. *Note the importance weights are now defined on the marginal space:*

$$w_t = \frac{p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t})}{q_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t})}.$$

### The Filter Derivative

In order to obtain the gradient vector with respect to the map variables, we apply standard differentiation rules to equation (4), yielding:

$$\begin{aligned} \nabla_{\theta} p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) &= \frac{\nabla_{\theta} \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t})}{\int \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t} \\ &- p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) \frac{\int \nabla_{\theta} \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t}{\int \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t}. \end{aligned} \quad (6)$$

Similarly, using the expansions for the derivatives of logs, the gradient of  $\xi(\cdot)$  can be written as follows:

$$\begin{aligned} \nabla_{\theta} \xi_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) &= p_{\theta}(\mathbf{y}_t | \mathbf{x}_t) \nabla_{\theta} \log p_{\theta}(\mathbf{y}_t | \mathbf{x}_t) \\ &\times \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \\ &+ p_{\theta}(\mathbf{y}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) \nabla_{\theta} p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \end{aligned} \quad (7)$$

*Note that the importance sampling process now happens in the marginal space.* The last integral in equation (4) can be expanded using the score identity:

$$\int p(\mathbf{x}_t | \mathbf{x}_{t-1}) \nabla_{\theta} \log [p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})] p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}$$

That is we sample from the marginal filtering distribution and weight with  $\beta \triangleq \nabla_{\theta} \log [p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})]$ . Contrast this with equation (2).

The other thing to note, as pointed out in [Poyadjis *et al.*, 2005a] is that the marginal filter derivative allows us to obtain a particle approximation of the Hahn-Jordan decomposition. This implies that we can surmount the problem of particles of opposite signs cancelling each other out in infinitesimal neighborhoods of the state space. Once again we refer the reader to Figure 1 of [Poyadjis *et al.*, 2005a] for an illustration of this phenomenon.

## Monte Carlo Implementation

We are now ready to present the particle algorithm of [Poyadjis *et al.*, 2005a] for approximating the filter derivative efficiently. Assume that at time  $t - 1$ , we have the following approximations of the filter and its gradient:

$$\begin{aligned}\widehat{p}_\theta(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) &= \sum_{i=1}^N w_{t-1}^{(i)} \delta_{\mathbf{x}_{t-1}^{(i)}}(\mathbf{x}_{t-1}) \\ \widehat{\nabla_\theta p}_\theta(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) &= \sum_{i=1}^N w_{t-1}^{(i)} \beta_{t-1}^{(i)} \delta_{\mathbf{x}_{t-1}^{(i)}}(\mathbf{x}_{t-1}).\end{aligned}$$

Then, we can sample from the proposal in equation (5) and compute the new unnormalized importance weights:

$$\begin{aligned}\widehat{w}_t^{(i)} &= \frac{p_\theta(\mathbf{y}_t|\mathbf{x}_t^{(i)}) \sum_{j=1}^N w_{t-1}^{(j)} p(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(j)})}{q_\theta(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t})} \\ \widehat{\rho}_t^{(i)} &= \frac{p_\theta(\mathbf{y}_t|\mathbf{x}_t^{(i)}) \sum_j w_{t-1}^{(j)} p(\mathbf{x}_t^{(j)}|\mathbf{x}_{t-1}^{(j)}) [\nabla_\theta \log p_\theta(\mathbf{y}_t|\mathbf{x}_t^{(i)}) + \beta_{t-1}^{(j)}]}{q_\theta(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t})}\end{aligned}$$

These weights lead to the following approximations:

$$\begin{aligned}\widehat{\xi}_\theta(\mathbf{x}_t, \mathbf{y}_{1:t}) &= \frac{1}{N} \sum_{i=1}^N \widehat{w}_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t) \\ \widehat{\nabla_\theta \xi}_\theta(\mathbf{x}_t, \mathbf{y}_{1:t}) &= \frac{1}{N} \sum_{i=1}^N \widehat{\rho}_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t)\end{aligned}$$

Finally, normalizing the weights and substituting the above Monte Carlo estimates into the expression for the derivative of  $p_\theta$  in terms of  $\xi_\theta$ , we obtain:

$$\widehat{\nabla_\theta p}_\theta(\mathbf{x}_t, \mathbf{y}_{1:t}) = \sum_{i=1}^N w_t^{(i)} \beta_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t)$$

where,

$$w_t^{(i)} \beta_t^{(i)} = \frac{\widehat{\rho}_t^{(i)}}{\sum_j \widehat{w}_t^{(j)}} - w_t^{(i)} \frac{\sum_j \widehat{\rho}_t^{(j)}}{\sum_j \widehat{w}_t^{(j)}}$$

## On-Line Map Learning

Armed with Monte Carlo estimates of the filter derivative, we can now attack the problem of developing recursive map estimators. Here, we choose to maximize the predictive likelihood (also known as the innovations or evidence):

$$p_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \int \int p_\theta(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1}) p_\theta(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1:t}$$

To accomplish this, we adopt the following stochastic approximation algorithm:

$$\theta_t = \theta_{t-1} + \gamma_t \nabla_\theta \log \widehat{p}_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1})$$

Provided that the step size  $\gamma_t$  satisfies standard stochastic approximation criteria, see for example [Bertsekas and Tsitsiklis, 1996; Spall, 2005], it can be shown that  $\theta_t$  converges to the set of global or local maxima of  $l(\theta)$ , where [Poyadjis *et al.*, 2005a]

$$l(\theta) = \lim_{k \rightarrow \infty} \frac{1}{k+1} \sum_{t=1}^k \log p_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1})$$

A detailed analysis is presented in [Tadic and Doucet, 2005]. The only remaining detail is to describe the Monte Carlo approximation of the gradient of the predictive distribution. It can be derived as follows:

$$\begin{aligned}\nabla_\theta \log \widehat{p}_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1}) &= \frac{\widehat{\nabla_\theta p}_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1})}{\widehat{p}_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \\ &= \frac{\int \widehat{\nabla_\theta \xi}_\theta(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t}{\int \widehat{\xi}_\theta(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t} \\ &= \frac{\sum_{j=1}^N \widehat{\rho}_t^{(j)}}{\sum_{j=1}^N \widehat{w}_t^{(j)}}\end{aligned}\quad (8)$$

## Pseudo-Code for Marginal Slam

The Marginal-SLAM algorithm is depicted in Figure 1. Note that it is linear in the number of features. It has an  $O(N^2)$  complexity in terms of the number of samples, but this can be reduced to  $O(N \log N)$  using the fast multipole expansions and metric tree recursions proposed in [Klaas *et al.*, 2005].

The marginal particle filter is an old idea [Fearnhead, 1998; Pitt and Shephard, 1999]. Yet, because of its large computational cost, it was not fully explored until the introduction of fast methods [Klaas *et al.*, 2005]. When using the transition prior as proposal, the marginal filter and classical particle filter are equivalent [Khan *et al.*, 2004], but this is no longer true when computing the derivative of the filter as outlined in [Poyadjis *et al.*, 2005b] and this paper.

## 4 Multi-robot Implementation

Hereinafter, we focus on feature-based representations because our goal is to apply the method to visually guided mobile robots [Lowe, 2004; Bay *et al.*, 2006; Newman *et al.*, 2006]. The robot motion model is based on simple differential drive vehicle

$$\begin{aligned}X_t &= X_{t-1} + d_t \cos(\psi_{t-1}) \\ Y_t &= Y_{t-1} + d_t \sin(\psi_{t-1}) \\ \psi_t &= \psi_{t-1} + \alpha_t,\end{aligned}$$

where  $\mathbf{x}_t = [X_t, Y_t, \psi_t]$  denotes the robot pose and heading and  $\mathbf{u}_t = [d_t, \alpha_t]$  denotes the translation and rotation motion commands (odometry) at time  $t$  with corresponding Gaussian noise  $v_t \sim \mathcal{N}(0, \sigma_R I)$ , where  $\sigma_R = [\sigma_d, \sigma_\alpha]$ . The transition model is Gaussian in terms of  $\mathbf{u}_t$ , but it involves a nonlinear transformation between  $\mathbf{u}_t$  and  $\mathbf{x}_t$ . Specifically, it is given by the following expression:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) = \frac{1}{2\pi\sigma_d\sigma_\alpha} \exp \left[ \frac{(d_t - \hat{d}_t)^2}{-2\sigma_d^2} + \frac{(\alpha_t - \hat{\alpha}_t)^2}{-2\sigma_\alpha^2} \right], \quad (9)$$

where

$$\begin{aligned}\hat{d}_t &= \sqrt{d_x^2 + d_y^2} \\ \hat{\alpha}_t &= \arctan \left( \frac{-d_x \sin(\psi_{t-1}) + d_y \cos(\psi_{t-1})}{d_x \cos(\psi_{t-1}) + d_y \sin(\psi_{t-1})} \right).\end{aligned}$$

with  $d_x = X_t - X_{t-1}$  and  $d_y = Y_t - Y_{t-1}$ . In the work, we will use this transition prior as the proposal distribution.

### Marginal-SLAM

- For  $i = 1, \dots, N$ , sample the robot state from the proposal

$$\mathbf{x}_t^{(i)} \sim \sum_{j=1}^N w_{t-1}^{(j)} q_{\theta}(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1}^{(j)})$$

- For  $i = 1, \dots, N$ , evaluate the importance weights

$$\tilde{w}_t^{(i)} = \frac{p_{\theta}(\mathbf{y}_t | \mathbf{x}_t^{(i)}) \sum_{j=1}^N w_{t-1}^{(j)} p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(j)})}{q_{\theta}(\mathbf{x}_t^{(i)} | \mathbf{y}_{1:t})}$$

$$\begin{aligned} \tilde{\rho}_t^{(i)} &= \frac{p_{\theta}(\mathbf{y}_t | \mathbf{x}_t^{(i)})}{q_{\theta}(\mathbf{x}_t^{(i)} | \mathbf{y}_{1:t})} \\ &\cdot \sum_{j=1}^N w_{t-1}^{(j)} p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(j)}) \cdot \\ &\cdot [\nabla_{\theta} \log p_{\theta}(\mathbf{y}_t | \mathbf{x}_t^{(i)}) + \beta_{t-1}^{(j)}] \end{aligned}$$

- Normalise the importance weights

$$\begin{aligned} w_t^{(i)} &= \frac{\tilde{w}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}} \\ w_t^{(i)} \beta_t^{(i)} &= \frac{\tilde{\rho}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}} - w_t^{(i)} \frac{\sum_j \tilde{\rho}_t^{(j)}}{\sum_j \tilde{w}_t^{(j)}} \end{aligned}$$

- Update the map vector

$$\theta_t = \theta_{t-1} + \gamma_t \frac{\sum_j \tilde{\rho}_t^{(j)}}{\sum_j \tilde{w}_t^{(j)}}$$

- Update the learning rate  $\gamma_t$ .

Figure 1: The Marginal-SLAM algorithm at time  $t$ .

We do this for both Marginal-SLAM and FastSLAM so as to provide a fair comparison. One can do better however by adopting approximations of the optimal proposal distribution that account for the latest observations. Typical approximations of this type include the extended and unscented Kalman filters.

We assume a simple range and bearing sensor  $\mathbf{y}_t = [\rho_t, \phi_t]$  with a point feature detector:

$$\begin{aligned} \hat{\rho}_t &= \sqrt{\Delta_x^2 + \Delta_y^2} \\ \hat{\phi}_t &= \arctan\left(\frac{-\Delta_x \sin(\psi_t) + \Delta_y \cos(\psi_t)}{\Delta_x \cos(\psi_t) + \Delta_y \sin(\psi_t)}\right), \end{aligned}$$

where  $\Delta_x = \theta_x - X_t$  and  $\Delta_y = \theta_y - Y_t$ , with  $[\theta_x, \theta_y]$  denoting the feature location. The sensor has white Gaussian noise  $v_t \sim \mathcal{N}(0, \sigma I)$  in the measurement space, where  $\sigma = [\sigma_{\rho}, \sigma_{\phi}]$ . Hence, the observation likelihood for a single fea-

ture is:

$$p(\mathbf{y}_t | \mathbf{x}_t) = \frac{1}{2\pi\sigma_{\rho}\sigma_{\phi}} \exp\left[\frac{(\rho_t - \hat{\rho}_t)^2}{-2\sigma_{\rho}^2} + \frac{(\phi_t - \hat{\phi}_t)^2}{-2\sigma_{\phi}^2}\right] \quad (10)$$

The overall observation likelihood is given by a product of the individual likelihood terms. The gradient of the log-likelihood can be obtained by simple differentiation:

$$\nabla_{\theta} \log(p(\mathbf{y}_t | \mathbf{x}_t)) = \frac{1}{\rho_t} \begin{bmatrix} \frac{\Delta_x(\rho_t - \hat{\rho}_t)}{\sigma_{\rho}^2} - \frac{\Delta_y(\phi_t - \hat{\phi}_t)}{\rho_t \sigma_{\phi}^2} \\ \frac{\Delta_y(\rho_t - \hat{\rho}_t)}{\sigma_{\rho}^2} + \frac{\Delta_x(\phi_t - \hat{\phi}_t)}{\rho_t \sigma_{\phi}^2} \end{bmatrix}. \quad (11)$$

One of the main SLAM difficulties is partial observability of the parameters. The choice of learning rates  $\gamma_t$  is affected by this partial observability. In our case, we chose to implement separate learning rates for each landmark. Each individual rate depends on the number of times its corresponding feature is observed. In addition, when there is no observation, the likelihood function of the unseen features is assumed to be uniform over the whole space, then  $\nabla_{\theta} \log(p(\mathbf{y}_t | \mathbf{x}_t)) = 0$ .

### 4.1 Non-Stationarity and Estimate Recovery

Although our goal has been to develop a method for static maps using decreasing learning rates, it is possible to adopt small constant learning rates to estimate time-varying parameters (maps). This enables us to attack maps that contain some moving parameters, such as chairs and doors [Martinez-Cantin *et al.*, 2006].

The same strategy for dealing with non-stationarity can also be adopted to correct for modelling errors. For example, the ability to track moving features enables Marginal-SLAM to recover from errors in data association, *provided that the correct correspondences are obtained in subsequent steps*. This is crucial because in the early steps of the algorithm there aren't enough observations to guarantee global map consistency. Figure 6 in the experimental section, illustrates how Marginal-SLAM is robust with respect to reasonable errors in matching.

### 4.2 Map Merging

Our goal is to merge the map estimates of each robot into a single global map. When the relative robot locations are known, the problem is simple and reduces to standard monolithic SLAM. In practice, however, we do not know the transformations of coordinates that would allow all the robots to have the same reference frame for the global map. We must therefore estimate the parameters of these linear transformations (rotation and translation) between the poses of the various robots.

Some authors simplify this problem by only carrying out the map merging step when two robots make ‘‘eye-contact’’ [Howard, 2005]. In general, however, the only information available is the data association between common features in both maps. In this setting, some authors have adopted Monte-Carlo localization using joint path sampling [Fox *et al.*, 2006]. However, as we have argued in earlier sections, this approach could easily result in degeneracy of the filter. We point out in passing that map merging is only one of the

1. For each robot, compute their pose and map:

- For  $i = 1, \dots, N$ , particles

1. Sample new particles from the prior mixture proposal:

- (a) Select mixture components using stratified resampling:

$$\left\{ \mathbf{x}'_{t-1}, \frac{1}{N} \right\}_{i=1}^N \leftarrow \left\{ \mathbf{x}_{t-1}, w_{t-1}^{(i)} \right\}_{i=1}^N$$

- (b) Simulate the stochastic dynamics for each selected component  $p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}'_{t-1})$ :

$$\begin{aligned} [d_t^{(i)}, \alpha_t^{(i)}] &= [d_t, \alpha_t] + v_t^{(i)} ; v_t^{(i)} \sim \mathcal{N}(0, \sigma I) \\ X_t^{(i)} &= X_{t-1}^{(i)} + d_t^{(i)} \cos(\psi_{t-1}^{(i)}) \\ Y_t^{(i)} &= Y_{t-1}^{(i)} + d_t^{(i)} \sin(\psi_{t-1}^{(i)}) \\ \psi_t^{(i)} &= \psi_{t-1}^{(i)} + \alpha_t^{(i)} \end{aligned}$$

2. Evaluate the importance weights using the expressions in equations (9), (10) and (11) for the transition prior (odometry), likelihood and gradient:

$$\begin{aligned} \tilde{w}_t^{(i)} &= p_\theta(\mathbf{y}_t | \mathbf{x}_t^{(i)}) \\ \tilde{\rho}_t^{(i)} &= \tilde{w}_t^{(i)} \nabla_{\theta} \log p_\theta(\mathbf{y}_t | \mathbf{x}_t^{(i)}) \\ &+ \frac{p_\theta(\mathbf{y}_t | \mathbf{x}_t^{(i)}) \sum_{j=1}^N w_{t-1}^{(j)} \beta_{t-1}^{(j)} p(\mathbf{x}_t^{(i)} | \mathbf{u}_t^{(i)}, \mathbf{x}_{t-1}^{(j)})}{\sum_{j=1}^N w_{t-1}^{(j)} p(\mathbf{x}_t^{(i)} | \mathbf{u}_t^{(i)}, \mathbf{x}_{t-1}^{(j)})} \end{aligned}$$

- For  $i = 1, \dots, N$ , normalise the importance weights:

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}} ; w_t^{(i)} \beta_t^{(i)} = \frac{\tilde{\rho}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}} - w_t^{(i)} \frac{\sum_j \tilde{\rho}_t^{(j)}}{\sum_j \tilde{w}_t^{(j)}}$$

- Update the individual map vector:

$$\theta_t = \theta_{t-1} + \gamma_t \frac{\sum_j \tilde{\rho}_t^{(j)}}{\sum_j \tilde{w}_t^{(j)}}$$

- Update the learning rates  $\gamma_t$ .

2. If maps share features:

- Compute the relative linear transformation between the coordinates of both robots.
- Transform all maps to a common reference frame using the estimated linear transformations.
- Place all transformed features in a common global map. All robots use this map at step  $t + 1$ .

Figure 2: The multi-robot Marginal-SLAM algorithm at time  $t$ . This version uses stratified resampling to sample from the mixture proposal. The proposal in this case is simply a mixture of transition priors.

components of the more general approach presented in [Fox *et al.*, 2006].

Here, we adopt standard vision algorithms to compute the correspondences between features [Hartley and Zisserman, 2000; Brown and Lowe, 2005]. In order to simplify the presentation, let us for the time being consider only two robots:  $R_1$  and  $R_2$ . When the two robots share a common set of map features for which we have computed the correspondences, we apply standard algorithms to estimate the transformation of coordinates between the poses or  $R_1$  and  $R_2$ . (We refer the reader to either [Castellanos and Tardós, 1999, Appendices A and B] for a detailed explanation or [Brown and Lowe, 2005] for a brief yet comprehensive explanation on how to carry out feature matching and estimation of transformations between different reference frames.)

Once we know the transformation of coordinates, we map  $R_2$ 's pose to the reference frame of  $R_1$ . We now have a common reference frame for the poses of both robots. We also transform the estimates of the location of map features to this common reference frame. Finally we merge the estimates of the map feature locations of both robots in order to produce the common global map.

As mentioned previously in Section 4.1, we noticed that Marginal-SLAM was able to converge empirically to the true map, even in the presence of errors in the estimates of the relative transformations.

The multi-robot Marginal-SLAM algorithm is detailed in figure 2. For clarification, the pseudo-code is based on the robot and sensor models presented in this paper, but it can be easily modified to treat other models.

## 5 Experiments

In this preliminary work, experiments are based on simulations in a controlled environment, where all the error sources and data associations are known. This simulated environment allows us to carry out experiments for one and multiple robots using different sensor and motion noise models.

We first present a brief comparison of Marginal-SLAM and FastSLAM. For a more detailed comparison, see [Martinez-Cantin *et al.*, 2006]. Figure 3 shows a comparison between the number of effective particles  $N_{eff} = 1 / \sum_{i=1}^N w_i^2$  using Marginal-SLAM and FastSLAM. This quantity measures the degeneracy of sequential Monte Carlo algorithms. To obtain a fair comparison, we use the same transition prior  $p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1})$  as the proposal distribution in both algorithms. Clearly, the marginal particle filter reaches a steady state, but FastSLAM loses particle diversity at a very fast rate. These results are based on the following parameter settings:  $L = 200, \sigma_d = 0.1m, \sigma_\alpha = 0.5deg, \sigma_\rho = 0.025m, \sigma_\phi = 3deg$ , where  $L$  denotes the number of landmarks.

In the multi-robot case, we used only 50 landmarks to ensure that we have a demanding testing scenario. That is, the robots share very few features in common. For simplicity, we use only two robots, but the algorithm extends naturally to more robots.

We assume that the initial relative robot location is unknown. As shown in Figure 4, we have to learn the linear transformation (rotation and translation) of coordinates between the two robots using standard geometry. When the robots detect common features, the estimated linear transforma-

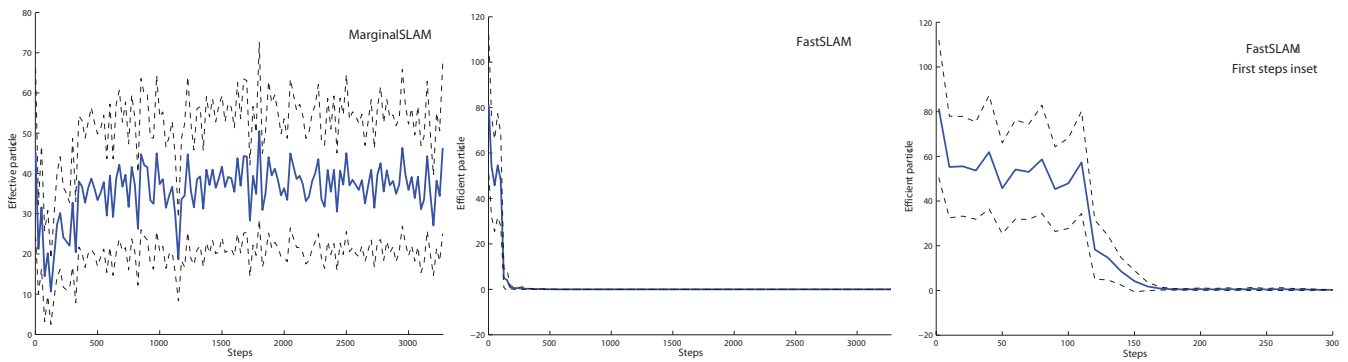


Figure 3: Average number of effective particles  $N_{eff}$ , with corresponding confidence intervals, for 25 simulations. The number of particles is 200 in both cases. The left plot corresponds to Marginal-SLAM, where the filter achieves a steady state with a 25% effective number of particles. The middle plot shows the same experiment for FastSLAM. The effective sample size quickly drops and FastSLAM fails to maintain particle diversity after 200 steps. The right plot is a zoomed view of the middle plot.

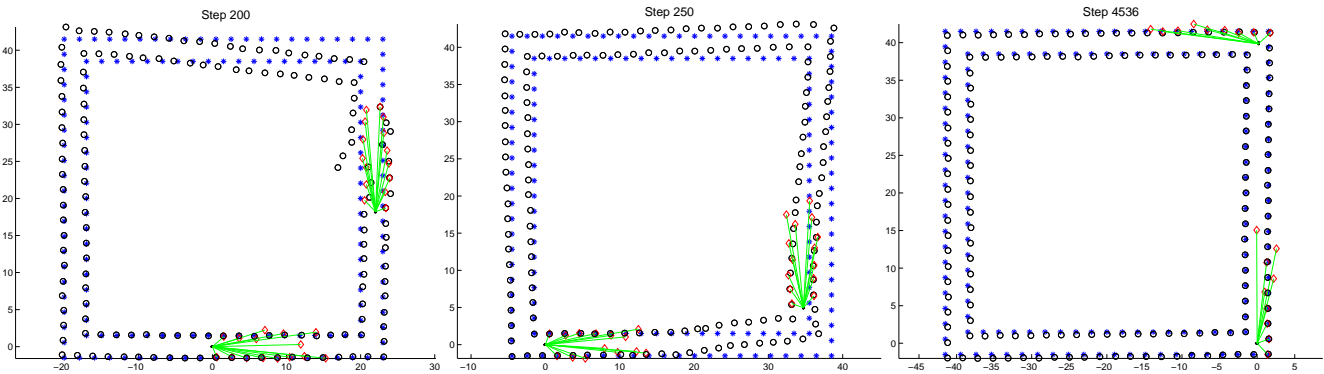


Figure 6: Loop closing in Multi-robot Marginal-SLAM with two robots. Despite the error introduced during map merging, the robot is able to close the loop (left plot). Although the map estimate after loop closing is still inaccurate (middle plot), Marginal-SLAM is able to minimize the average error and converge to the true solution. The right plot displays the estimate and true maps map after 10 loops. The blue stars represent the true map and the black circles correspond to the most recent map estimate. The red diamonds are the current observations.

tion is used to represent both maps with respect to a common reference frame (robot  $R_1$ 's position in our case). The two maps are then merged into a single global map. This is illustrated in Figure 5. In this figure, the estimated transformation is biased, because the estimates of the maps have not converged and the observations are very noisy. Therefore, the map merging step introduces some error in the global map. However, as discussed in Section 4.1, Marginal-SLAM is able to recover from these errors. Figure 6 reveals how  $R_1$  and  $R_2$  close the loop and keep updating the shared map, while reducing the error introduced in the map merging step. The plots are with respect to  $R_1$ 's location. There is no need for absolute location or reference information.

Finally, Figure 7 shows the average mean Euclidean distance of the landmarks estimate with respect to the true location after ten repetitions of the experiment. We use a robocentric representation based on the expectation of the first robot's location. The peaks represent points with high heading error

in the loop closing area for this robot. The convergence of the filter is empirically verified in this last experiment.

We close this section by stating that Marginal-SLAM is able to build accurate maps with large range and bearing noise. However, very large sensor noise (for example sonar) demands a high number of particles to ensure convergence. In addition, many update steps are required to obtain reasonable errors. We believe that data driven proposals and adaptive stochastic approximation could be applied in the future to confront this problem and speed up the convergence rate.

## 6 Conclusions

The experiments and arguments in this paper and [Martinez-Cantin *et al.*, 2006] indicate that Marginal-SLAM is an important new direction in the design of particle methods for SLAM. It is clear that algorithms designed to work on the marginal space behave better than the ones designed to work on the path space. They also lend themselves naturally to the

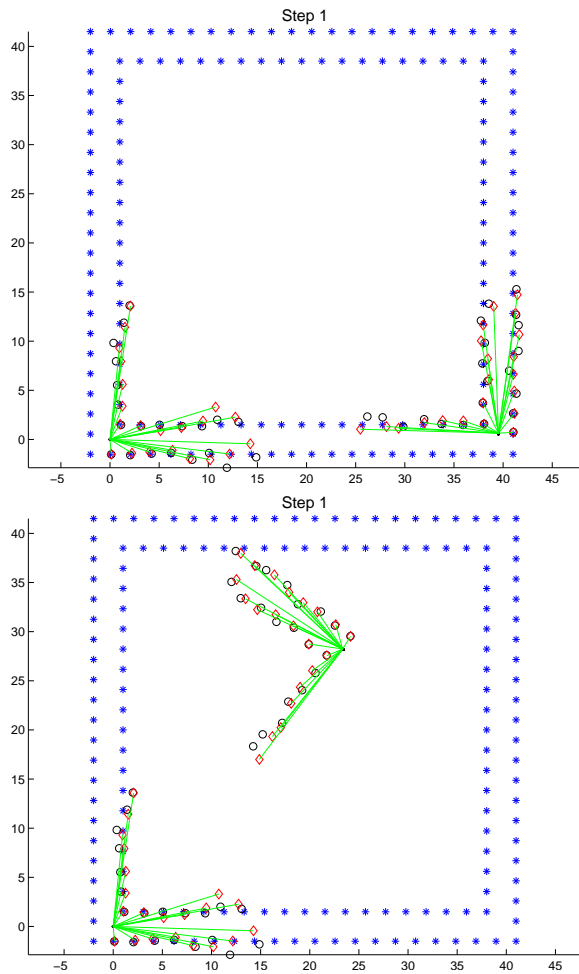


Figure 4: In the first iteration of the multi-robot experiment, the unknown relative position of the robots is modelled in terms of an unknown linear transformation. The upper plot represents the true robot locations. The lower plot shows the robot locations, while emphasizing that we don't know the linear transformation of coordinates at this early stage. The blue stars represent the true map and the black circles correspond to the most recent map estimate. The red diamonds are the current observations.

global map estimation in the multi-robot setting.

Marginal-SLAM seems to be robust with respect to errors introduced during the map joining step. This is important as computer vision matching algorithms are not perfect. We are planning to deploy Marginal-SLAM using the visual features and matching procedure presented in [Brown and Lowe, 2005]. This will enable us to obtain 3D maps of the environment.

In future work, we plan to adopt better proposal distributions using extended and unscented Kalman filters and to implement Hessian updates so as to potentially maximize the predictive likelihood more efficiently.

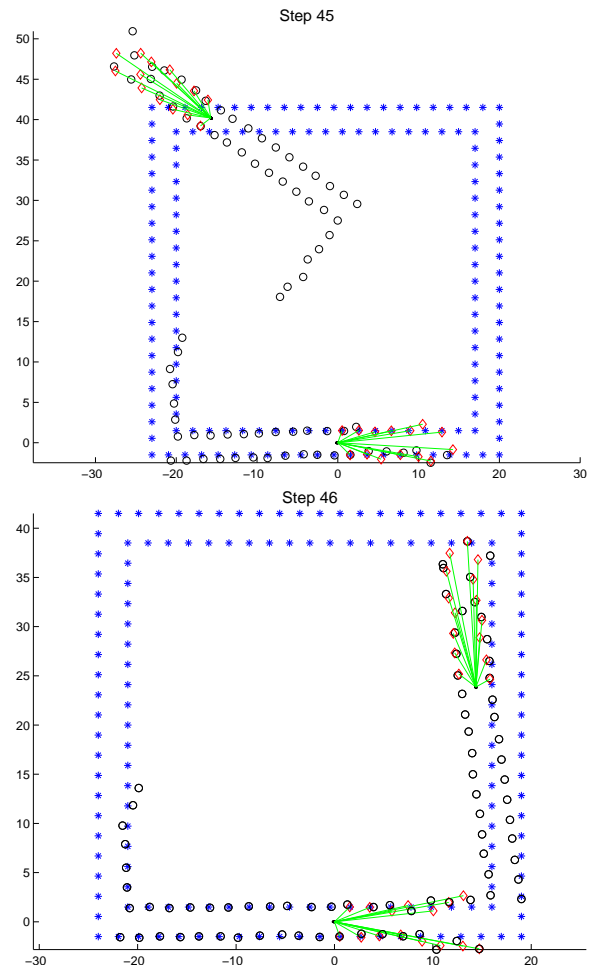


Figure 5: Map merging in multi-robot Marginal-SLAM. The unknown relative robot location is recovered using the estimated transformation between common features in both maps. In the lower plot, maps have been merged. The resulting map is biased because of the poor relative location estimate at this early stage. Again, the blue stars represent the true map and the black circles correspond to the most recent map estimate. The red diamonds are the current observations.

## 7 ACKNOWLEDGMENTS

We are very thankful to Arnaud Doucet for many thoughtful discussions. We are also indebted to Frank Dellaert and Tim Bailey for valuable feedback on an earlier version of this paper.

This work was supported in part by the Dirección General de Investigación of Spain under project DPI2003-07986 and by NSERC.

## References

- [Andrieu *et al.*, 1999] C Andrieu, N de Freitas, and A Doucet. Sequential MCMC for Bayesian model selection. In *IEEE Higher Order Statistics Workshop*, pages 130–134, Caesarea, Israel, 1999.

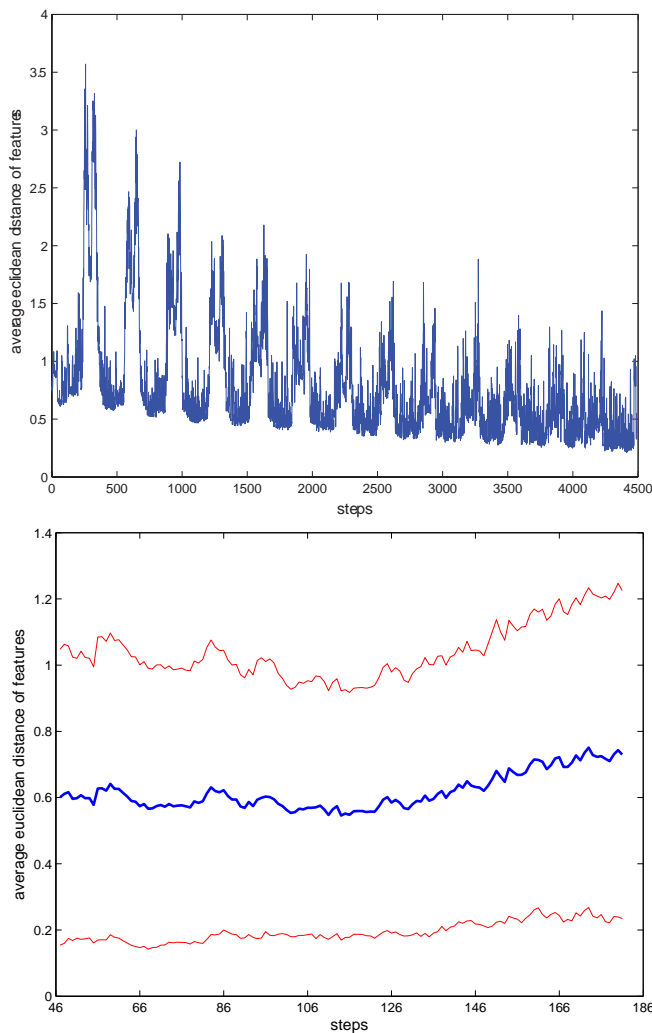


Figure 7: Evolution of the average map error in 10 runs. The error is measured in terms of the Euclidean distance between the robocentric map and the ground truth map. Therefore, the robot location error is implicitly included in this plot. The top plot shows that the algorithm converges asymptotically. The bottom plot is a zoomed version of the top plot with 95% confidence intervals, which indicate the robustness of the algorithm.

[Bailey *et al.*, 2006] Tim Bailey, Juan Nieto, and Eduardo Nebot. Consistency of the FastSLAM algorithm. In *IEEE International Conference on Robotics and Automation*, 2006.

[Bay *et al.*, 2006] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proceedings of the ninth European Conference on Computer Vision*, May 2006.

[Bertsekas and Tsitsiklis, 1996] D P Bertsekas and J N Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[Brown and Lowe, 2005] M Brown and D Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *5th International Conference on 3D Imaging and Modelling*, Ottawa, Canada, 2005.

[Castellanos and Tardós, 1999] J.A. Castellanos and J.D. Tardós. *Mobile Robot Localization and Map Building. A Multisensor Fusion Approach*. Kluwer Academic Publishers, 1999.

[Doucet *et al.*, 2000] A Doucet, N de Freitas, K Murphy, and S Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In C Boutilier and M Godszmidt, editors, *Uncertainty in Artificial Intelligence*, pages 176–183. Morgan Kaufmann Publishers, 2000.

[Doucet *et al.*, 2001] A Doucet, N de Freitas, and N J Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[Doucet, 1998] A Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR 310, Department of Engineering, Cambridge University, 1998.

[Durrant-Whyte and Bailey, 2006] Hugh Durrant-Whyte and Tim Bailey. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. *Robotics and Automation Magazine*, June 2006.

[Elinas *et al.*, 2006] Pantelis Elinas, Robert Sim, and James J. Little. sigmaSLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution,. In *IEEE International Conference on Robotics and Automation*, 2006.

[Fearnhead, 1998] P Fearnhead. *Sequential Monte Carlo Methods in Filter Theory*. PhD thesis, Department of Statistics, Oxford University, England, 1998.

[Fox *et al.*, 2001] D Fox, S Thrun, W Burgard, and F Dellaert. Particle filters for mobile robot localization. In A Doucet, N de Freitas, and N J Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[Fox *et al.*, 2006] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart. Distributed multi-robot exploration and mapping. *Proc. of the IEEE*, 2006.

[Hartley and Zisserman, 2000] R Hartley and A Zisserman. *Mobile Robot Localization and Map Building. A Multisensor Fusion Approach*. Kluwer Academic Publishers, 2000.

[Howard, 2005] A. Howard. Multi-robot simultaneous localization and mapping using particle filters. In *In Proc. Robotics: Science and Systems Conference*, 2005.

[Khan *et al.*, 2004] Z. Khan, T. Balch, and F. Dellaert. A Rao-Blackwellized particle filter for EigenTracking. 2004.

[Klaas *et al.*, 2005] Mike Klaas, Nando de Freitas, and Arnaud Doucet. Toward practical  $n^2$  Monte Carlo: The marginal particle filter. In *Uncertainty in Artificial Intelligence*, 2005.

[Klaas *et al.*, 2006] Mike Klaas, Mark Briers, Nando de Freitas, Arnaud Doucet, Simon Maskell, and Dustin Lang. Fast particle smoothing: If i had a million particles. In *International Conference on Machine Learning*, 2006.

[Lowe, 2004] D.G. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 60, pages 91–110, 2004.

[Martinez-Cantin *et al.*, 2006] R. Martinez-Cantin, N. de Freitas, and J.A. Castellanos. Marginal-SLAM: A Convergent Particle Method for Simultaneous Robot Localization and Mapping, 2006.

[Montemerlo *et al.*, 2003] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint*

- [Newman *et al.*, 2006] Paul Newman, David Cole, and Kin Ho. Outdoor slam using visual appearance and laser ranging. In *International Conference on Robotics and Automation (ICRA)*, Florida, 2006.
- [Pitt and Shephard, 1999] M K Pitt and N Shephard. Filtering via simulation: Auxiliary particle filters. *JASA*, 94(446):590–599, 1999.
- [Poyadjis *et al.*, 2005a] G. Poyadjis, A. Doucet, and S.S. Singh. Maximum likelihood parameter estimation using particle methods. In *Joint Statistical Meeting*, 2005.
- [Poyadjis *et al.*, 2005b] G. Poyadjis, A. Doucet, and S.S. Singh. Particle methods for optimal filter derivative: Application to parameter estimation. In *IEEE ICASSP*, 2005.
- [Spall, 2005] J C Spall. *Introduction to Stochastic Search and Optimization*. Wiley, 2005.
- [Stachniss *et al.*, 2005] C. Stachniss, G. Grisetti, and W. Burgard. Recovering particle diversity in a Rao-Blackwellized particle filter for SLAM after actively closing loops. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 667–672, Barcelona, Spain, 2005.
- [Tadic and Doucet, 2005] V B Tadic and A Doucet. Exponential forgetting and geometric ergodicity for optimal filtering in general state-space models. *Stochastic Processes and Their Applications*, 115:1408–1436, 2005.
- [Thrun, 1993] Sebastian Thrun. Exploration and model building in mobile robot domains. In E Ruspini, editor, *Proceedings of the ICNN*, pages 175–180, 1993.
- [Thrun, 2001] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.
- [Thrun, 2002] S. Thrun. Robotic mapping: A survey. In G. Lake-meyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.