# Design and analysis of a framework for real-time vision-based SLAM using Rao-Blackwellised particle filters

Robert Sim, Pantelis Elinas, Matt Griffin, Alex Shyr and James J. Little
Department of Computer Science,University of British Columbia
2366 Main Mall,Vancouver, BC, V6T 1Z4
{simra,elinas,mgriffin,shyr,little}@cs.ubc.ca

## Abstract

*This paper addresses the problem of simultaneous localization and mapping (SLAM) using vision-based sensing. We present and analyse an implementation of a Rao-Blackwellised particle filter (RBPF) that uses stereo vision to localize a camera and 3D landmarks as the camera moves through an unknown environment. Our implementation is robust, can operate in real-time, and can operate without odometric or inertial measurements. Furthermore, our approach supports a 6-degree-of-freedom pose representation, vision-based ego-motion estimation, adaptive resampling, monocular operation, and a selection of odometry-based, observation-based, and mixture (combining local and global pose estimation) proposal distributions. This paper also examines the run-time behavior of efficiently designed RBPFs, providing an extensive empirical analysis of the memory and processing characteristics of RBPFs for vision-based SLAM. Finally, we present experimental results demonstrating the accuracy and efficiency of our approach.*

## 1. Introduction

Recent advances in methods in state estimation have led to a plethora of approaches to solving the simultaneous localization and mapping (SLAM) problem (for example, [1,3,6,7,11,15,16]). However, there are a limited number of *vision-based* solutions that address large-scale real-time mapping, and that can scale up to tens of thousands, or even millions of mapped features. Furthermore, the variety of vision-based modalities, such as monocular versus stereo sensing, and pinhole versus omnidirectional camera models, and the development of increasingly advanced techniques in state estimation and particle filtering, have created the need for general purpose frameworks for implementing, testing and deploying vision-based SLAM solutions. This

paper presents a framework for implementing vision-based SLAM solutions based on Rao-Blackwellised particle filters (RBPFs) that operate in real-time over long trajectories (on the order of 100m or more), resulting in maps consisting of many thousands of landmarks. A key feature of our framework is its flexibility – the user can select from a wide array of operating modalities depending on their hardware platform and enable or disable various advanced particle filtering techniques; these include

- Vision-based options

    - Monocular or stereo sensing
    - Vision-based ego-motion estimation

- Particle filtering options

    - Standard proposal distribution $p(s_t|s_{t-1}, u_t)$.
    - Vision-based proposal distribution (odometry-free ego-motion) $p(s_t|s_{t-1}, z_t)$.
    - Global pose estimation and mixture proposal distributions $p(s_t|m_{t-1})$.

- State and Hybrid map representations

    - 6-DOF pose representation.
    - Ability to construct occupancy grids in addition to landmark-based map [5].

Our approach to solving the SLAM problem with a vision sensor is to combine a Rao-Blackwellised particle filter (RBPF)-based approach to mapping [15], coupled with efficient data structures developed by Montemerlo *et al.* for representing a distribution over maps (referred to as Fast-SLAM [14]). Specifically one of the important contributions of the FastSLAM algorithm is the data structure it employs to share information between trajectory samples with common history. This facilitates real-time performance of the algorithm as the map grows. In addition, we employ fast

IEEE
COMPUTER
SOCIETY

data association techniques for matching the relatively unambiguous feature descriptors obtained using the SIFT feature detector [12]. As an exploratory vehicle moves through the environment, the number of landmarks in its map can grow to number in the hundreds of thousands. This poses a difficult problem for solving the data association problem, where a single feature observation might require comparison with all of the landmarks in the map. Such an extensive comparison might be required when extracted features are generic, without any uniquely defining characteristics (such as those typically employed in mapping algorithms based on laser range sensing). Furthermore, the computed data association is rarely unique, and often highly ambiguous. Previous approaches have addressed this problem by sampling over possible data associations [13]. In our work, we employ the SIFT feature descriptor, which provides descriptions of feature observations that have been shown to be very robust for feature correspondence. In addition, we apply a kd-tree over the space of SIFT features to facilitate approximate nearest-neighbor lookups in time logarithmic in the number of visually distinct landmarks [2].

The main contributions of this paper are two-fold. First, we present an application framework for performing vision-based SLAM based on RBPFs. The framework is extensible and flexible in that it supports a wide range of common operating modalities. Second, we present and discuss experimental results illustrating the behavior of our implementation. These results are important for evaluating FastSLAM as an appropriate method for SLAM in general and vision-based SLAM in particular. It is important and noteworthy that of the existing implementations, including the original implementation by Montemerlo *et al.* [14], none have performed an extensive analysis of the operating characteristics of the filter, such as memory consumption, sample set size, update frequency, and resampling frequency, opting instead to report only successful mapping experiments.

The remainder of this paper is structured to provide a coverage of the strengths and weaknesses of current methods, elaborate on the details of our implementation, present and discuss experimental results, and finally discuss planned improvements.

## 2. Related Work

In general, two competing map representations have emerged as popular choices in the literature: landmark-based [14, 16, 19] and occupancy grid based [4, 8]. Occupancy grids are effective for dense but ambiguous information while landmarks are more suited to sparse but distinguishable features. Very impressive occupancy grids have been produced online by recent scan matching techniques which also use particle filters for pose estimation [4, 8]. Landmark and vision-based approaches also perform well,
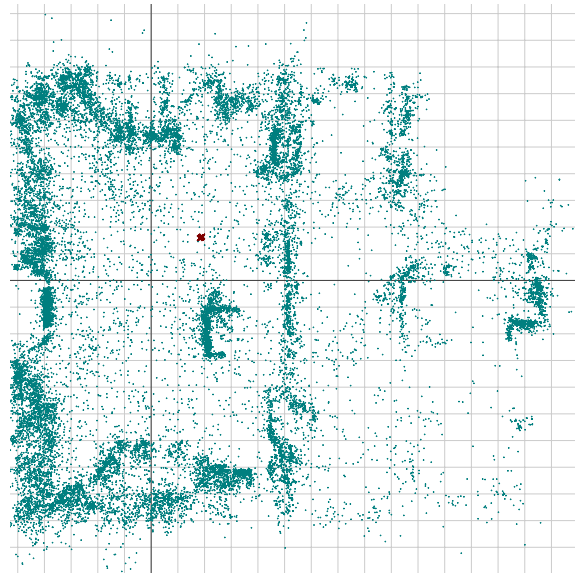


**Figure 1. Example map of two rooms in a laboratory environment. The map consists of 184,211 3D landmarks constructed in real-time using 100 particles. The robot travelled approximately 128m in constructing the map. Grid lines indicate 1m intervals. Refer to the experimental results for an evaluation of the map quality.**

as in [16]. In the latter case, a laboratory environment was successfully mapped by using a Kalman Filter and assuming independence between landmark and pose estimates. For larger environments, this approach is likely to be overconfident and lead to filter divergence. Similarly, Davison implemented a monocular SLAM approach that operates well in small environments, but has difficulties scaling to larger environments [3]. The vSLAM approach employs particle filters to localize in a hybrid metric-topological map [9]. However, it is not clear that the representation can scale well and still provide robust localization everywhere in the world.

This paper is an extension of two previously published papers addressing vision-based SLAM [17, 18], in which we introduced a RBPF-based approach to mapping with a stereo camera. Other most similar to ours is the FastSLAM implementation presented by Barfoot [1]. The chief differences between that work and ours are the real-time performance of our system for up to 2000 particles, and its flexibility in the variety of operating modalities, such as operating without odometry when it is unavailable, or the ability to cope with monocular data. Furthermore, we can demonstrate robust loop-closing capabilities in larger environments than have previousy been encountered using vision.

## 3. Simultaneous Localization and Mapping

This paper represents map estimation as the evolution of a Rao-Blackwellised particle filter [15]. In this context, the trajectory and landmark distribution is modeled as a dynamic Bayes network, where trajectories are instantiated as samples, and the landmark distribution can be expressed analytically for each trajectory. At time $t$, let $s_t$ denote the vehicle pose, $m_t$ the map learned thus far and $x_t = \{s_t, m_t\}$ be the complete *state*. Also, let $u_t$ denote a control signal or a measurement of the vehicle's motion from time $t-1$ to time $t$ and $z_t$ be the current observation. The set of observations and controls from time 0 to $t$ are denoted as $z^t$ and $u^t$ respectively. Our goal is to estimate the density

$$p(s^t, m_t | z^t, u^t) = p(x^t | z^t, u^t) \qquad (1)$$

It has been demonstrated elsewhere that $p(s^t, m_t | z^t, u^t)$ can be approximated by factoring the distribution in terms of sampled trajectories $s_t$, and independent landmark distributions conditioned on the sampled trajectories [15]:

$$p(s^t, m_t | z^t, u^t) \approx p(s^t | z^t, u^t) \prod_k p(m_t(k) | s^t, z^t, u^t) \quad (2)$$

where $m(k)$ denotes the $k-th$ landmark in the map. That is, we instantiate a set of samples $s^t$, propagate them according to $u^t$, and construct maps for each according to $z^t$.

A simplistic approach to running an RBPF for SLAM would yield an update complexity of $O(NK)$, where $N$ is the number of samples at each step and $K$ is the number of landmarks. However, Montemerlo et al. introduced a tree-based structure which refines the amortized complexity to $O(N \log K)$ by sharing landmark estimates between samples [14]. Each sample in the filter will share unaltered landmark estimates with other samples (those landmarks that have not been observed since the time the samples became siblings). Each landmark observation results in a landmark being copied and updated but the rest of the map remains unaltered.

## 4. Implementation

This section will describe our implementation. Given the wide array of user options, we will limit our description to using a standard proposal distribution, and stereo sensing. In subsequent sections we will describe the special considerations involved in implementing the additional optional features, such as monocular sensing. The RBPF operates at each time step by stochastically sampling a motion for each sample from a proposal distribution, and subsequently weighting each sample according to an observation model. In the following sections we describe the procedure for these operations.
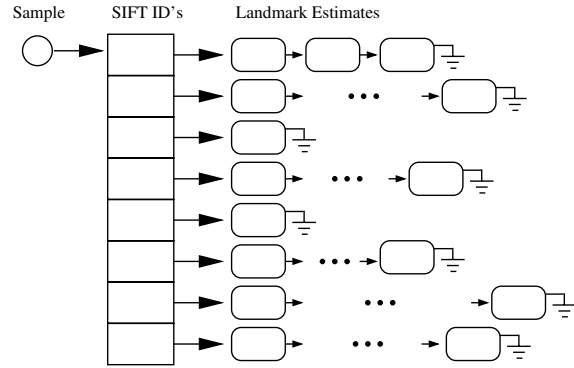


**Figure 2. Conceptually, each sample has an associated map, organized by SIFT descriptor. Each SIFT descriptor might have multiple landmark estimates, each spatially distinct.**

### 4.1. Map Representation

We employ a data structure similar to that described in [14] as a map representation. Conceptually, each particle has associated with it a set of landmark estimates, described by Gaussian distributions. We take advantage of the descriptive power of SIFT (described below), enabling us improve the quality of data association. In this formulation, each sample maintains a list of SIFT IDs, and these IDs in turn point to a linked list of one or more 3D landmark estimates (Figure 2). Note that one SIFT ID can point to multiple landmarks – landmarks that have similar appearance but are spatially distinct. The implementation of the map structure is identical to that of the original reference-counted FastSLAM binary search tree, with the exception that indexing a landmark by SIFT ID returns a linked list of the landmarks associated with that ID. Individual landmark estimates are represented as 3D Gaussian distributions using the Extended Kalman Filter.

### 4.2. State Representation and Proposal Distributions

We describe samples of the vehicle's pose with the set $s_t = \{T, R\}$, where $T = [x\ y\ z]$ is the robot's position and $R = [\rho\ \theta\ \phi]$ is the robot's heading described by Euler angles in 6-DOF.

At each time step, the $N$ pose samples are propagated according to the proposal distribution $q(s_t | s_{t-1}, z_t, m_{t-1})$. We will refer to the *standard proposal* as the distribution based on the robot's motion model:

$$q(s_t | s_{t-1}, z_t, m_{t-1}) = p(s_t | s_{t-1}, u_t).$$

Over time the distribution of samples can become non-Gaussian, and even multi-modal. Other proposal distri-

butions have been suggested, and our system facilitates a vision-based ego-motion estimator to produce a visual odometry model:

$$q(s_t|s_{t-1}, z_t, m_{t-1}) = p(s_t|s_{t-1}, z_t),$$

and a global pose estimator to support mixture proposal distributions:

$$q(s_t|s_{t-1}, z_t, m_{t-1}) = \alpha p(s_t|s_{t-1}, z_t) + (1-\alpha)p(s_t|m_{t-1})$$

Note that the visual odometer is independent of odometry inputs, facilitating SLAM in the absence of odometry or IMU measurements. We describe the design and implementation of the alternative proposal distributions elsewhere [5].

For the case of the standard proposal distibution, the specific action sequence is dependent on the robot's exploration policy. For this paper, we drive the robot by hand, and either infer the robot's actions from odometry measurements between observations.

After taking an observation $z_t$ of a landmark (described in the next section), each particle in the current generation of particles is weighted according to the probability of the current observation $z_t$, conditioned on that particle's trajectory:

$$
\begin{aligned}
w_{i,t} &= \frac{p(z_t|s_{i,t}, m_{i,t-1})p(s_t|s_{t-1}, u_t)}{q(s_t|s_{t-1}, u_t, z_t, m_{t-1})} \, w_{i,t-1} \quad (3)\\
&= p(z_t|s_{i,t}, m_{i,t-1}) \, w_{i,t-1} \quad (4)\\
&= k \exp(-0.5\Delta z^T \Sigma^{-1} \Delta z) \, w_{i,t-1} \quad (5)
\end{aligned}
$$

where we are assuming the standard proposal, $\Delta z = h(s_{i,t}) - z_t$, $h(\cdot)$ is a generative model of the observation as a function of pose, $\Sigma$ is the sum of the measurement covariance and prediction covariance. The particle is weighted according to how well the current observation is consistent with the map constructed from that particle's trajectory.

## 4.3. Weight Normalization and Adaptive Resampling

Special consideration must be taken when computing the particle weight, particularly where large numbers of feature observations, with significant potential for outlier correspondences, are present. We accumulate the log likelihood of observations over time, and employ a normalization technique described below to prevent catastrophic numerical outcomes.

$$
\begin{aligned}
\log w_{i,t} &= \log p(z_t|s_{i,t}, m_{i,t-1}) + \log w_{i,t-1} \quad (6)\\
&= -0.5 \min(T_l, \Delta z^T \Sigma^{-1} \Delta z) + \log w_{i,t-1} \quad (7)
\end{aligned}
$$

where the maximum observation deviance $T_l$ is selected so as to prevent outlier observations from significantly affecting the observation likelihood. However, given the potentially large numbers of observations, even with a reasonable
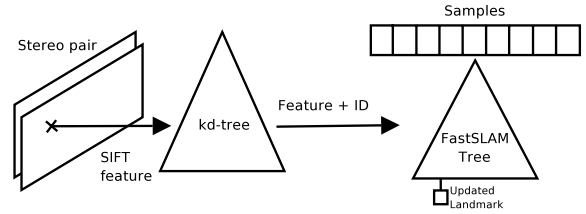


**Figure 3. Observation update (refer to text for details)**

setting for $T_l$, the magnitude of the log-likelihood can be such that raising it to the exponential to evaluate the $i$-th particle weight results in zero – we employ a normalization procedure described in [17] that preserves accuracy while preventing catastrophic numerical results.

Once particle weights are calculated, the filter determines whether it is appropriate to resample. Resampling too often can result in sample starvation in the filter, resulting in poor modelling of the posterior distribution over states. We use the technique suggested in [21] to evaluate the effective sample size $N_{eff}$:

$$N_{eff} = \frac{1}{\sum_i w_i^2}$$

The general heuristic is to resample when $N_{eff}$ falls less than $N/2$, where $N$ is the number of samples in the filter. Resampling involves sampling probabilistically with replacement to produce the next generation of particles. When any particle is not chosen for advancement it is pruned, and all nodes in the FastSLAM tree to which it refers have their reference counts decremented, and are deleted if their reference counts reach zero.

## 4.4. Observation Model and Data Association

Figure 3 and Algorithm 1 summarize the observation update process. We extract SIFT features using the difference of Gaussian detector described in [12]. In the stereo case, we perform a linear search of the keys in the left image for the best match to each key in the right, subject to epipolar constraints, and determine its 3D position and covariance according to the well-known stereo equations:

$$Z = fB/d, \ \ X = uZ/f, \ \ Y = vZ/f$$

where $f$ is the focal length of the camera, $B$ is the base-line of the stereo head, $d$ is the disparity between SIFT keys in the left and right images and $[u \ v]$ is the pixel position of the key in the right camera.

---

**Algorithm 1** Observation update procedure

$F$:= Extract SIFT keys and positions $f = \{k, p\}$ from image.
**for all** features $f$ in $F$ **do**
    $id$:= kd_tree_lookup($f.k$) {Index into kd-tree.}
    **for all** Samples $s$ **do**
        List $L$:= $s$.map_lookup($id$)
        Find most likely landmark estimate $l$ in $L$, given $f.p$ {Maximizing observation likelihood.}
        Copy $l$ if necessary {If shared with other samples.}
        Update $l$ with $f.p$ using Kalman Filter update.
        Update $w_t$ for $s$ according to observation likelihood.
    **end for**
**end for**

---

Once landmark observations are extracted from the stereo pair[1], the landmark estimates must be updated for the individual samples. To efficiently store and access what can quickly become a large number of SIFT keys we use a kd-tree. The kd-tree facilitates nearest-neighbor matching in time logarithmic in the size of the tree, and has been demonstrated to be reliable for object recognition tasks [2]. The disadvantage of using a kd-tree is that it can sometimes produce not the nearest match but a close match. We maintain a single tree for the sensor and associate an arbitrary integer ID with each SIFT identifier we add. New keys are considered to be *candidate keys* and are not passed as an observation to the particle filter until they have been observed for a sufficient number of frames.

Each particle's map is indexed by a set of IDs associated with SIFT descriptors and each node contains a linked list of 3D landmarks sharing that descriptor. Multiple data associations can be entertained by the filter because each particle determines the specific landmark to which an observation corresponds. A sample's weight is updated for a given landmark observation according to Equation 5 by first selecting from the linked list for the matched landmark ID the landmark estimate that is most likely to have generated the observed point. If the observation deviance exceeds a particular threshold, or if no previous landmark estimates exist for a particular SIFT ID, a new landmark estimate is instantiated using the observation as the initial estimate.

### 4.5. Monocular SLAM

Several authors have demonstrated that in order to successfully map an environment with a bearings-only sensor, it is important to maintain several initial landmark hypotheses until a substantially wide baseline is achieved between observations [3, 10, 20]. We take advantage of the framework's ability to assign multiple landmarks per SIFT ID

---

[1]In the monocular case, the complete list of SIFT keys is returned along with their pixel positions- no depth computation is performed.

in order to perform undelayed landmark initialization for accomplishing SLAM using a monocular camera (that is, performing bearings-only SLAM). We employ the initialization method proposed by Sola *et al.*, which aims to accurately represent the full probability distribution of a landmark's position using a geometric progression of Gaussian distributions along the cone projected from the camera's focal point through the pixel location of the observation [20].

Specifically, when a landmark is initialized, its probability distribution is a cone with Gaussian cross-section, lying along the ray projected from the focal point of the camera through the pixel corresponding to the observation such that the intersection of the cone with the image plane represents the measurement covariance R. This distribution is approximated by defining a progression of Gaussian distributions such that

$$p(l_t|z_t) = \sum_{1}^{N_g} N(s_j, \sigma_j)$$

where $s_j = \beta^{j-1} s_1$, $\sigma_j = \beta^{j-1} \sigma_1$ and $N_g$, $\beta$, $s_1$, and $\sigma_1$ are user-defined constants calibrated according to the scale of the environment being mapped.

Observation updates in the monocular case are performed by updating all of the individual landmark estimates and recording the observation likelihood for each in a vector $\Lambda$. The landmark whose observation likelihood is highest is used to weight the sample, and over time landmarks whose $\Lambda_i$ drops below a threshold are pruned, until a single landmark estimate is determined. Note that for the monocular case, this approach precludes the estimation of multiple distinct landmark estimates (corresponding to distinctly different 3D points) per SIFT ID.

In this paper we will present results for the monocular case, however our current implementation requires that odometry be available to resolve the well-known scale ambiguity in monocular SLAM and structure from motion (SFM) problems.

## 5. Experimental Results

For the purposes of our experiments, we used an RWI B14 robot with a BumbleBee stereo head from Point Grey Research. The robot was driven by a human operator through a laboratory environment consisting of two rooms of total size approximately 19m by 16.5m, and the robot collected 8500 stereo images along a trajectory of approximately 128m. Figure 4 depicts some sample images from the sequence. Note the presence of motion blur and saturation. The entire sequence and odometry log was saved for testing under the various operating modes of our system. All of the filter tests were conducted on a Pentium Xeon 3.2GHz PC with 4GB of RAM.

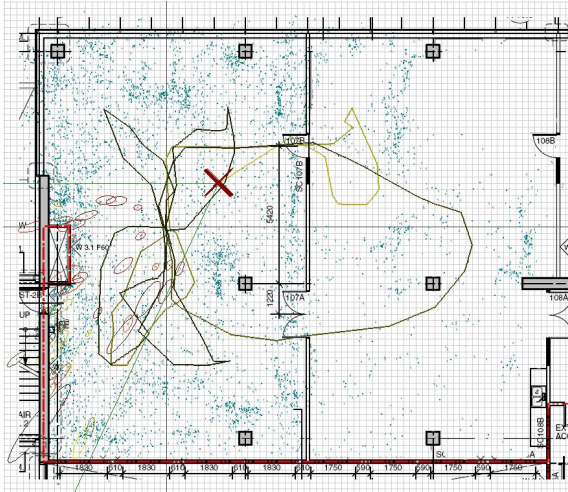**Figure 4. Sample frames from our test sequences.**



**Figure 5. Map constructed using standard proposal distribution, with overlaid architectural drawing. Note that apparently spurious landmarks correspond to furniture or points on the ceiling or floor. The grid marks 25cm intervals. The trajectory of the filter is plotted, with the set of samples marked by red 'x's. Note that the robot successfully closes a large loop (going out one door and returning through another), although the precise location of one door is off by about 0.5m, and also successfully locates the top door on three occasions.**

## 5.1. FastSLAM Behavior using a standard proposal

Our first set of results illustrates the behavior of our system using a standard proposal distribution (that is, camera motions are sampled stochastically based only on odometry). Figure 5 presents the map constructed by our system using 100 particles, with an architectural drawing overlaid to illustrate the features of the environment.

Figure 6 illustrates the run-time performance of the filter as we varied the number of samples in the filter, from values
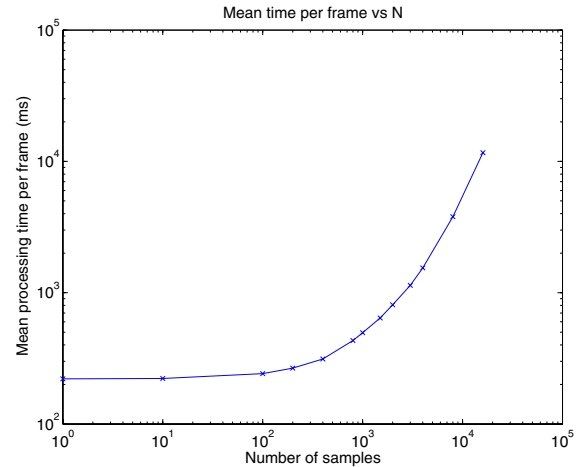


**Figure 6. Mean processing time per frame vs number of samples**

of 1 up to 16000, roughly doubling the number of samples at each increment. The performance is measured in terms of the mean processing time per frame, in milliseconds, over the run of 8500 frames. In all cases the number of landmarks represented *per map* at the end of each of these runs was 12,316. Note that the plot is in log-log format.

It is worthwhile noting that the filter can run at 1Hz at approximately 3000 samples, enabling real-time performance for large sample sets. It is also interesting to note that the slope of the log-log plot is less than 1 up to approximately 2000 samples, but begins to demonstrate superlinear behavior beyond this value. Note that the theoretical performance of the RBPF and FastSLAM map representation is $O(NK \log K)$, hence the performance should scale linearly for fixed $K$ and varying $N$. The main reason for the superlinear performance appears to be an increased cache and page fault frequency as $N$ grows.

Figure 7 illustrates the memory consumption of the filter for varying $N$, represented in terms of the total number of nodes in the FastSLAM tree, averaged over all time instances. As $N$ increases, the number of nodes required
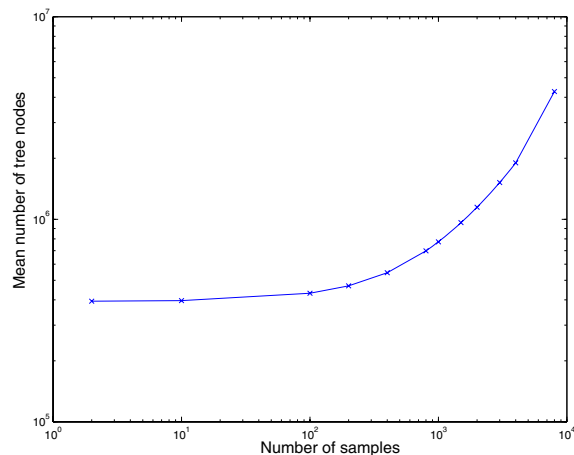
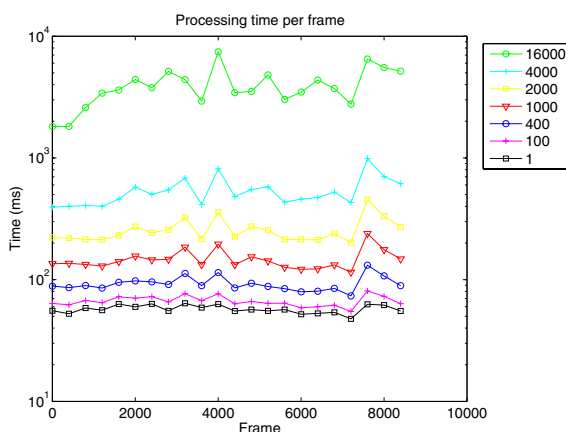**Figure 7. Mean nodes in the FastSLAM tree versus number of samples.**



**Figure 8. Processing time versus time instance for varying values of sample set size $N$.**

for map maintenance increases, and while the mean number of nodes is much smaller than the theoretical maximum of $O(NK)$, the slope of the plot at 8000 samples is 1.17, suggesting that the consumption for larger numbers of samples may not scale well.

The main concern for many SLAM implementations is the performance as the number of landmarks increases. Figure 8 shows the average processing time per frame over 400 frame intervals for a selection of sample set sizes $N$. The rate of growth in the number of landmarks is approximately constant over the entire run (at most 5 new landmarks per frame). This plot demonstrates that while there is a modest increase in the cost of maintaining the map over time, the filter performance does not degrade significantly over time.

From these results we can conclude that while Fast-SLAM scales well for large numbers of landmarks (for the environments considered), it is not clear that the filter will scale well for very large numbers of samples. This may explain in part the difficulty other researchers have had in implementing FastSLAM with more than a small number of particles. The cause for the superlinear behavior shown in Figure 7 is under investigation.

### 5.2. Accuracy

The main goal of any SLAM architecture is robust, accurate mapping. However, without an accurate map in the first instance, it is difficult to reliably evaluate the performance of a SLAM algorithm. Our approach to map evaluation was to record the filter's estimate of the robot's position as it visited a set of pre-determined waypoints. Map accuracy was then defined in terms of the accuracy of the robot's trajectory, as evaluated by its ability to localize at the waypoints. The set of waypoints and their positions are marked with the numbers 1-5 in Figure 9. The waypoints were visited in the sequence 1,3,1,2,1,4,1,5,1,1,3,1 where the robot traversed a large loop through an adjacent room in the 1,1 phase of the sequence.

For the following set of experiments, we ran the filter with 100 samples, and varied the operating modality of the filter. We defined five waypoints located in the corners and center of the main room that the robot explored and measured their positions. Table 1 summarizes the configuration of the filter for each run, along with the mean error in the trajectory estimate. Similarly, Figure 10 summarizes the results. Figure 9 a),b) and c) plot the localization estimates at each of the waypoints for the standard proposal, visual odometry, and monocular camera runs, respectively. The green '*'s correspond to the dead reckoning estimate (identical in all four plots), and the blue 'x's indicate the filter pose estimate at the time instances the robot visited the waypoints. The waypoints themselves are plotted with a red 'o', and a line connects each pose estimate to the corresponding waypoint. In all three cases the number of mapped SIFT features was 13957. The "Mapped Landmarks" column in Table 1 indicates the number of landmarks in each case that had more than three observations.

It is important to note that even the worst performing approach (SLAM with a monocular camera) out-performed the robot's dead reckoning. The erroneous monocular estimates are largely due to the robot's failure to re-localize when it closed the large loop through the second room. We hope that advanced mixture proposal techniques, such as that describe in our related work [5], will lead to improved performance with the monocular modality. We also note that the accuracy of the visual odometry approach facilitates the use of non-robotic platforms, such as hand-held cam-

**Table 1. Summary of operating modes and experimental results for evaluating map accuracy.**

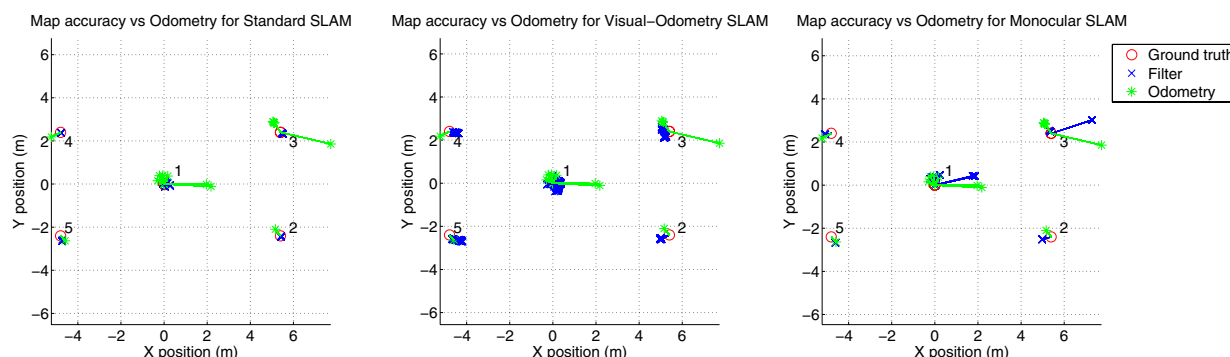| Run Title | Known Odometry | Visual Odometry | Stereo/ Monocular | Mean map error (m) | Mapped Landmarks |
|---|---|---|---|---|---|
| Dead Reckoning | n/a | n/a | n/a | 0.70 | n/a |
| Monocular | yes | no | monocular | 0.61 | 4051 |
| Visual Odometry | no | yes | stereo | 0.26 | 11266 |
| Standard | yes | no | stereo | 0.10 | 6228 |



**Figure 9. Map accuracy for a) standard proposal, b) visual odometry, and c) monocular camera.**
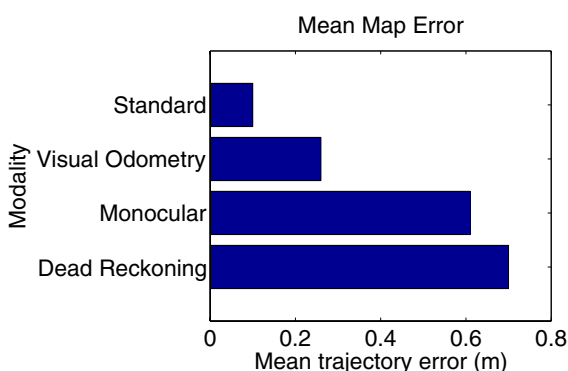


**Figure 10. Summary of map accuracies for various operating modalities**

eras, and other devices where odometry may not be available.

## 6. Conclusion

This paper has presented the design and analysis of an application framework for conducting experiments in vision-based SLAM using Rao-blackwellised particle filters. The experimental results presented illustrate the capability of the system for operating in real-time with significant numbers of samples. We also demonstrate successful mapping for a variety of modalities, including vision-based ego-motion estimation and monocular sensing. This work represents a significant step beyond the state of the art in terms of both its flexibility and the scale and density of the environments we are successfully mapping with vision.

There are a number of features of our system that, due to space limitations, we have not presented. These include the incorporation of global pose estimation in a mixture proposal distribution, and the construction of coarse occupancy grids that enable an exploring robot to plan and navigate safely. Our current work exploits these features for enabling autonomous navigation. We are also examining how the monocular mapping system can be improved to facilitate accurate mapping without odometry. Ultimately, we hope to deploy a system that can accurately map large environments using a freely moving monocular camera without odometric inputs.

## References

[1] T. Barfoot. Online visual motion estimation using fastslam with sift features. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, pages 3076–3082, Edmonton, AB, August 2005. IEEE/RSJ, IEEE Press.

[2] J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, Peurto Rico, June 1997. IEEE, IEEE Press.

[3] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the IEEE Int. Conf. on Computer Vision*, pages 1403–1410, Nice, France, 2003.

[4] A. I. Eliazar and R. Parr. DP-SLAM 2.0. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, 2004. IEEE Press.

[5] P. Elinas, R. Sim, and J. J. Little. $\sigma$SLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, FL, May 2006. IEEE, IEEE Press. To appear.

[6] R. Eustice, H. Singh, and J. Leonard. Exactly sparse delayed-state filters. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2428–2435, Barcelona, Spain, April 2005.

[7] J. Folkesson, P. Jensfelt, and H. I. Christensen. Graphical SLAM using vision and the measurement subspace. In *Intl Conf. on Intelligent Robotics and Systems (IROS)*, Edmundton, Canada, Aug 2005. IEEE/JRS.

[8] D. Hähnel, D. Fox, W. Burgard, and S. Thrun. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proc. of the Conference on Intelligent Robots and Systems (IROS)*, 2003.

[9] N. Karlsson, E. D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vSLAM algorithm for robust localization and mapping. In *2005 IEEE International Conf. on Robotics and Automation, ICRA 2005*, 2005.

[10] N. Kwok and G. Dissanayake. An efficient multiple hypothesis filter for bearing-only slam. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, pages 736–741, Sendai, Japan, October 2004. IEEE Press.

[11] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1442–1447, Osaka, Japan, November 1991.

[12] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Int. Conf. on Computer Vision*, pages 1150–1157, Corfu, Greece, September 1999. IEEE Press.

[13] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Proceedings of the 2003 International Conference on Robotics and Automation*, pages 1985–1991, 2003.

[14] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conf. on Artificial Intelligence*, pages 593–598, Edmonton, Canada, 2002. AAAI.

[15] K. Murphy. Bayesian map learning in dynamic environments. In *1999 Neural Information Processing Systems (NIPS)*, pages 1015–1021, 1999.

[16] S. Se, D. G. Lowe, and J. J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Int. J. Robotics Research*, 21(8):735–758, 2002.

[17] R. Sim, P. Elinas, M. Griffin, and J. J. Little. Vision-based SLAM using the Rao-Blackwellised particle filter. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR)*, pages 9–16, Edinburgh, Scotland, 2005.

[18] R. Sim, M. Griffin, A. Shyr, and J. J. Little. Scalable real-time vision-based SLAM for planetary rovers. In *IEEE IROS Workshop on Robot Vision for Space Applications*, pages 16–21, Edmonton, AB, August 2005. IEEE, IEEE Press.

[19] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.

[20] J. Solà, A. Monin, M. Devy, and T. Lemaire. Undelayed initialization in bearing only slam. In *Proc. Int. Conf on Intelligent Robots and Systems (IROS)*, pages 2751–2756, Edmonton, AB, August 2005. IEEE/RSJ, IEEE Press.

[21] 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.

IEEE
COMPUTER
SOCIETY