

# Evaluation of Algorithms for Bearing-Only SLAM

Kostas E. Bekris, Max Glick and Lydia E. Kavraki  
 Computer Science Dept., Rice University, Houston, TX 77005  
 Email: {bekris,mglick,kavraki}@rice.edu

**Abstract**—An important milestone for building affordable robots that can become widely popular is to address robustly the Simultaneous Localization and Mapping (SLAM) problem with inexpensive, off-the-shelf sensors, such as monocular cameras. These sensors, however, impose significant challenges on SLAM procedures because they provide only bearing data related to environmental landmarks. This paper starts by providing an extensive comparison of different techniques for bearing-only SLAM in terms of robustness under different noise models, landmark densities and robot paths. We have experimented in a simulated environment with a variety of existing online algorithms including Rao-Blackwellized Particle Filters (RB-PFs). Our experiments suggest that RB-PFs are more robust compared to other existing methods and run considerably faster. Nevertheless, their performance suffers in the presence of outliers. In order to overcome this limitation we proceed to propose an augmentation of RB-PFs with: (a) Gaussian Sum Filters for landmark initialization and (b) an online, unsupervised outlier rejection policy. This framework exhibits impressive robustness and efficiency even in the presence of outliers.

**Keywords:** Localization, mapping, bearing-only, sensors.

## I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) [18], the procedure of mapping a workspace and localizing a sensor using the map, has received a lot of attention in robotics and has led to impressive robot behaviors. However, in order to see SLAM approaches used widely, we must solve the SLAM problem for flexible, inexpensive, ubiquitous sensors, such as monocular cameras. Cameras exhibit an important limitation, however. A single image provides only the direction, or bearing, to environmental features. This complicates landmark initialization in online SLAM [10], [2], since at least two measurements from known, relatively disparate poses are needed to initialize an estimate. But a good estimate of the scene characteristics is required to compute the poses from which the measurements have been taken. Hence, bearing-only sensors aggravate the typical challenge of SLAM, namely the inter-dependence between mapping and localization. Despite these challenges, there have been various attempts to solve the bearing-only SLAM problem in the robotics literature. Most of these attempts employ an Extended Kalman Filter (EKF) [3] and focus on the proper initialization of the EKF [2]. Nevertheless, it has been argued that online approximations of Expectation-Maximization (EM), such as the Incremental Maximum Likelihood (IML) [17] approach, combined with Particle Filters (PFs) [1], are comparable to the EKF results [8]. An alternative, which combines PFs and the EKF, and is called Rao-Blackwellized PFs (RB-PFs) [6], has exhibited robust performance and time efficiency in the case of range-bearing sensors [20].

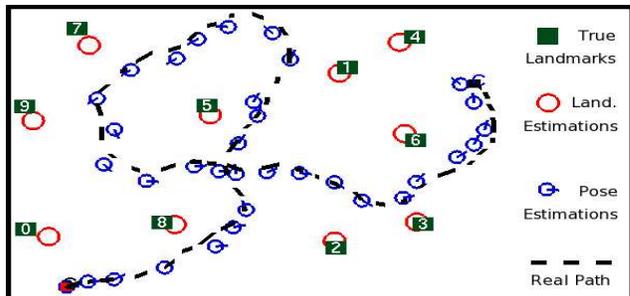


Fig. 1. An experiment with 100 readings and 10 landmarks using RB-PFs. The noise follows a Gaussian distribution with  $\sigma = 3^\circ$ .

The contribution of this paper is to apply multiple state-of-the-art algorithms to the online, bearing-only SLAM problem, compare them, and, based on their evaluation, develop a new algorithm that robustly addresses the problem. In particular:

**A.** We provide a comprehensive experimental comparison in a controlled environment for different noise models, landmark densities and robot paths of three different classes of algorithms that are applicable to the online bearing-only SLAM problem: (a) EKF (b) IML (c) and to the best of our knowledge RB-PFs for the first time. For each class we have implemented various approaches from the existing literature and we report here the best solutions in our setup.

**B.** Our experiments suggest that RB-PFs provide a robust and time efficient solution. However, outliers considerably reduce their performance. We propose the augmentation of RB-PFs with two methods to overcome outlier sensitivity: (a) a Gaussian Sum Filter (GSF) [10], [9] for initializing landmarks and (b) a single-cluster graph-partitioning algorithm for outlier rejection [12]. The combined framework performed better than every other approach across a wide selection of parameters, including a noise model with high occurrence of outliers. Figure 1 shows a SLAM result computed with this algorithm in our simulated environment.

## II. TECHNIQUES FOR BEARING-ONLY SLAM

We first provide an overview of the bearing-only SLAM problem and the classes of algorithms we have considered.

Assume a robot moving among  $n$  static landmarks. Let  $\xi = (x, y, \theta)$  denote the robot state and  $\xi(t_i : t_j)$  denote a trajectory from time  $t_i$  to  $t_j$ . The robot stores a map  $m$  with the landmarks' coordinates:  $\{l^1 = (x^1, y^1), \dots, l^n = (x^n, y^n)\}$ . The data  $d_T = (o(0 : T), u(0 : T - 1))$  available to the robot up to time  $T$  are the observations  $o(0 : T)$  and control inputs

$u(0 : T - 1)$ . A robot input  $u_t = (u_t^f, u_t^\theta)$  corresponds to a translation  $u_t^f$  followed by a rotation  $u_t^\theta$  and determines the kinematic model  $f_t(s_{t-1}, u_t)$ :

$$(x_{t+1}, y_{t+1}, \theta_{t+1}, m) = (x_t + u_t^f \cdot \cos(\theta_t), y_t + u_t^f \cdot \sin(\theta_t), \theta_t + u_t^\theta, m)$$

An observation  $o_t^i$  of landmark  $l_i$  from state  $\xi_t = (x_t, y_t, \theta_t)$  corresponds to the following computation (observation model) where  $w_t$  represents noise:

$$o_t^i = \text{atan2}(y_i - y_t, x_i - x_t) - \theta_t + w_t.$$

The goal is to concurrently estimate the map  $m$  and the robot state  $\xi_T$  at time  $T$ :  $s_T = (\xi_T, m)$ . The general Bayes filter computes a belief distribution  $B_T = P(s_T | d_T)$  at time  $T$  over  $s_T$  given the data  $d_T$  [19]. The computation requires (a) an initialization  $B_0$ , (b) a motion model  $P(\xi' | u, \xi)$  describing the probability that the robot is at location  $\xi'$  if it was previously at  $\xi$  and executed  $u$ , and (c) the observation model  $P(o | \xi, m)$  describing the likelihood of observing  $o$  when the robot is at  $\xi$  and the map  $m$  is correct. Then for a static map and a normalization factor  $\eta$  we get:

$$B_T = \eta P(o_T | \xi_T, m) \int P(\xi_T | u_{T-1}, \xi_{T-1}) B_{T-1} d\xi_{T-1} \quad (1)$$

The computational cost of integrating over all states renders the explicit computation of Eq. 1 inefficient. Most online algorithms simplify the problem by approximating Eq. 1 and three of the most popular approximations are described below.

#### A. Extended Kalman Filter

Assume the motion and observation models in Eq. 1 are expressed as  $s_t = f_t(s_{t-1}, u_t) + v_t$  and  $o_t = h_t(s_t) + w_t$ , where  $f_t, h_t$  are linear functions and  $v_t, w_t$  correspond to white, Gaussian distributed noise with zero mean and covariance matrices  $V_t$  and  $W_t$ , respectively. In this setting Eq. 1 can be solved optimally by the Kalman filter equations. We know, however, that  $f_t$  and  $h_t$  are non-linear for the bearing-only SLAM problem. Cheeseman et al. [15] proposed the Extended Kalman Filter for SLAM which linearizes  $f_t, h_t$  by using a Taylor approximation. If  $F_t = [\frac{\partial f}{\partial s}]$  and  $H_t = [\frac{\partial h}{\partial s}]$  the Jacobian matrices of  $f_t$  and  $h_t$  respectively, then the EKF estimates the mean state vector  $\bar{s}_t$  and covariance matrix  $P_t$  of  $B_t$  according to the following two steps:

$$\text{Predict : } \hat{s}_t = f_t(\bar{s}_{t-1}, u_t) \quad \hat{P}_t = F_t P_{t-1} F_t^T + V_t$$

$$\text{Update : } \bar{s}_t = \hat{s}_t + R_t \nu_t \quad P_t = \hat{P}_t - R_t H_t \hat{P}_t$$

where  $\nu_t = o_t - h_t(\hat{s}_t)$  and  $R = \hat{P}_t H_t^T (H_t \hat{P}_t H_t^T + W_t)^{-1}$ .

EKF has been a popular solution for bearing-only SLAM [3], [16], combined in some cases with other techniques like bundle adjustment [4]. The literature focuses a lot on how to initialize a landmark estimate. Costa et al. [2] use observations from two poses without data association. Alternatively a Gaussian Sum Filter (GSF) can be initialized, which prunes Gaussian hypotheses when they become invalid [9], [10].

#### B. Incremental Maximum Likelihood

An alternative to the EKF is Expectation Maximization (EM), which performs hill climbing in the space of trajectories  $\xi(0 : T)$  and maps  $m$ , computing the pair that maximizes the likelihood of producing the sensor readings [19]. The algorithm alternates between (a) estimating the trajectory based on the current map and (b) estimating a maximum likelihood map based on the trajectory. EM algorithms are applied offline due to their high computational cost.

An online heuristic based on EM is the Incremental Maximum Likelihood (IML) method [17]. In IML the state at time  $T$  is constructed from the  $(T - 1)$ -th map and pose via maximization of the marginal likelihood:

$$(\xi_T, m_T) = \underset{\xi_T, m_T}{\text{argmax}} P(o_T | \xi_T, m_T) P(\xi_T | u_{T-1}, \xi_{T-1}). \quad (2)$$

The above equation follows from Eq. 1 if the previous state is known. The algorithm does not provably converge and its weakness is cyclic environments where the error in robot poses grows unbounded. As a result, specific loop closing techniques are applied. IML is popular due to its simplicity and real-time implementations have worked well in practice. IML's limitations can be partially overcome by using Particle Filters (PFs) [5], [13], [1] to represent uncertainty. A PF approximates a distribution with particles  $p_i = (s_i, w_i)$  where  $s_i$  is a possible system state and  $w_i$  is  $s_i$ 's likelihood  $\sum_i w_i = 1$ . Kwok and Dissanayake [8] applied IML with PFs for bearing-only SLAM.

#### C. Rao-Blackwellization

If the goal of the SLAM problem is to compute  $B_T = P(s_T | d_T) = P(\xi(0 : T), m | d(0 : T))$ , then a key observation is that if  $\xi(0 : T)$  is known then the map  $m$  can be analytically computed. Furthermore, for a known trajectory we can also marginalize each landmark estimate in the map. Then instead of Eq. 1 we have the following expression for the belief distribution:

$$B_T = P(\xi(0 : T) | d(0 : T)) P(m | \xi(0 : T), d(0 : T)) = P(\xi(0 : T) | d(0 : T)) \prod_i P(l^i | \xi(0 : T), o^i(0 : T)). \quad (3)$$

This factorization implies that the computation can be decomposed into  $n + 1$  filters, an estimator over robot paths  $P(\xi(0 : T) | d(0 : T))$ , and  $n$  estimators over landmark locations  $P(l^i | \xi(0 : T), o^i(0 : T))$  conditioned on the path estimate. The above idea is known in the statistics literature as Rao-Blackwellization [6]. Thrun et al. [20] have provided an implementation of Eq. 3 with PFs where EKFs represent the uncertainty on the landmark estimates.

### III. IMPLEMENTED ALGORITHMS

We have implemented a series of SLAM algorithms and applied them to the case of bearing-only sensors. Here we describe the details of our implementation for each class of algorithms from section II

### A. Extended Kalman Filter

**1. EKF:** We have implemented a straightforward version of the EKF filter where the mean state vector  $\bar{s}_t$  is of dimension  $3 + 2n$  and the covariance matrix  $P_t$  of dimension  $(3 + 2n, 3 + 2n)$ , where 3 parameters are needed for the pose and 2 for each landmark. The pose is initialized at  $(0, 0, 0)$  and each landmark is initialized at a fixed distance  $\rho_c$  from the robot on the first observation ray. The covariance matrix is initialized so that the uncertainty around the initial pose is a circle and the uncertainty for each landmark is an ellipse. The length of the ellipse extends between the minimum  $\rho_{min}$  and maximum  $\rho_{max}$  assumed depth of our sensor. The width of the ellipse corresponds to two standard deviations of the Gaussian distribution that represents the noise model. Fig. 2(a) shows the initialization of the EKF and the state of the filter after 10 observations. For every new observation we execute the prediction and update steps from section II.A.

**2. GSF:** An important limitation of the EKF is its sensitivity to the initial distribution. In order to improve landmark initialization we have applied a Gaussian Sum Filter (GSF) [9], [10]. The idea is to use an EKF for tracking the pose and the global map and each new landmark is initialized with a separate GSF. The GSF is eventually pruned to a single Gaussian distribution and then the landmark is added to the global map. Although a single observation does not give any depth information, we can approximate a uniform distribution over the depth  $\rho$  of a recently discovered landmark with a sum of Gaussians  $\sum_i \Gamma(\rho_i, \sigma_{\rho_i})$ , where

$$\rho_0 = \frac{\rho_{min}}{1 - \alpha}, \quad \rho_i = \beta^i \cdot \rho_0, \quad \sigma_{\rho_i} = \alpha \cdot \rho_i, \quad w_i \propto \rho_i. \quad (4)$$

In our implementation we propagate all the Gaussian distributions defined in Eq. 4 during a training period. At the end of the training period we select the hypothesis with the highest likelihood and we prune the rest. The likelihood of  $\Gamma_i$  is the product of the likelihoods  $L_i^t$ , each one defined for an observation  $o^t$ :

$$L_i^t = \frac{1}{\sqrt{2\pi|S_i|}} \exp\left(-\frac{1}{2}(o^t - \hat{o}_i^t)^T S_i^{-1} (o^t - \hat{o}_i^t)\right), \quad (5)$$

where  $\hat{o}_i^t$  is the expected observation given the hypothesis and  $S_i$  is the covariance of the Kalman innovation:  $o^t - \hat{o}_i^t$ . Note that initially no landmark is present in the global map so the robot state is propagated according to odometry. Fig. 2(b) shows an example of initializing and tracking with GSFs.

### B. Incremental Maximum Likelihood

**1. SIR:** In order to implement Eq. 2 we have to decide how to represent the pose and the map. Some IML approaches represent the pose with PFs and then solve for the map analytically. We have applied PFs both for the estimation of the pose and the map. Since the robot pose is assumed known at each mapping step we can marginalize the map and use an independent PF for each landmark. Our version of IML with PFs is shown in Fig. 3. The PF follows the Sampling Importance Resampling (SIR) algorithm, also known in robotics as the Monte Carlo Localization (MCL) algorithm [19]. SIR suffers

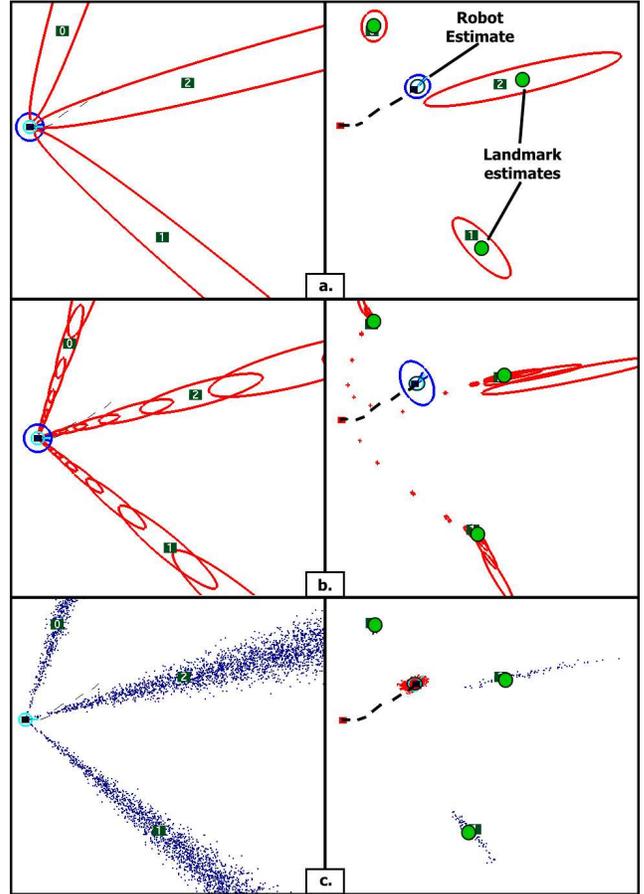


Fig. 2. SLAM for a scene with 3 landmarks. From top to bottom: EKF, GSF, SIR. The left column shows the initialization of the corresponding technique and the right column shows the state of the filters after 10 observations.

from the “particle impoverishment” problem since particles quickly collapse to the same state. This is especially true for particles that estimate a stationary state such as the mapping PFs in the algorithm of Fig. 3.

There are many alternatives for implementing PFs and we have also tested the “Sequential Importance Resampling” (SIS) algorithm with “Effective Sampling Size” (ESS) resampling [5], [13]. This technique does not sample on every iteration as in lines 1 and 6 of Fig. 3. Instead, a heuristic approximates how many of the particles have non-negligible weights and only when this number drops below a threshold does resampling occur. The SIS-ESS algorithm suffers from low variance of the particle population. On the other hand, “Mixture-MCL” [19] uses an observation directly to influence the particles’ states in that some particles are sampled not from the previous set, but from the region described by the last observation.

**2. Rough:** In order to overcome the impoverishment problem with the mapping PFs we have applied “roughing” [5]. This simple technique maintains variation in the particle population by applying random noise to the particle state instead of a propagation step and it performed better than simple SIR.

**3. Reset:** “Resetting” [7] behaves similarly to the “Mixture-MCL” approach but samples from the observation only when

<p><b>Require:</b> A PF representing the uncertainty over <math>\xi_{T-1}</math> and <math>n</math> PFs representing the uncertainty for each <math>l^i</math>.</p> <p><b>Ensure:</b> Maximum likelihood <math>s_T = (\xi_T, m_T)</math>.</p>
<p><b>Input:</b> Odometry <math>u_{T-1}</math> and sensor <math>o_T</math> readings.</p>
<ol style="list-style-type: none"> <li>1: Sample poses <math>s_{T-1}</math> given the current pose PF.</li> <li>2: Propagate samples given odometry data <math>u_{T-1}</math>.</li> <li>3: Re-weight samples given the observation model and the maximum likelihood map: <math>P(o_T s_T, m_{T-1})</math>.</li> <li>4: Estimate new pose <math>\xi_T</math> as the maximum weight particle.</li> <li>5: <b>for</b> every landmark <math>l_{T-1}^i</math> <b>do</b></li> <li style="padding-left: 20px;">6: Sample landmark locations given the <math>i</math>-th PF.</li> <li style="padding-left: 20px;">7: Re-weight samples given: <math>P(o_T^i s_T, l_{T-1}^i)</math></li> <li style="padding-left: 20px;">8: Estimate new landmark estimate <math>l_T^i</math>.</li> <li>9: <b>end for</b></li> <li>10: Compute map: <math>m_T = \{l_T^1, \dots, l_T^n\}</math>.</li> </ol>

Fig. 3. One step of the IML algorithm with SIR PFs for representing pose and map uncertainty.

it deviates substantially from the previous distribution. In particular, the resetting approach maintains two estimates, a long-term average of observation likelihoods for each landmark estimate  $P_l = P_l + \eta_l (P - P_l)$  and  $P_s$  the short-term counterpart.  $P$  represents the likelihood of the current landmark estimate (Eq. 5) given the last observation and  $\eta_l$  is the long-term smoothing parameter. The technique, resamples  $\tilde{n}$  samples according to a parameter  $\nu$  that controls the level at which samples are added:  $\tilde{n} = n \max(0, 1 - \nu \cdot \frac{P_s}{P_l})$ .

### C. Rao-Blackwellization

**1. RB-PFs:** A PF can be used for computing Eq. 3, where each particle specifies both the robot’s trajectory  $\xi(0 : T)$  and the map  $m$ . In reality, we do not have to store the entire trajectory but only the most recent state  $\xi$ . For the map we can use  $n$  low-dimensional EKFs to track the landmarks as in the work by Thrun et al. [20]. Then the  $p$ -th particle state  $s_t^p$  corresponds to the tuple  $s_t^p = \langle \xi_t^p, l_{1,t}^p, \Sigma_{1,t}^p, \dots, l_{n,t}^p, \Sigma_{n,t}^p, w_t^p \rangle$ , where  $l_{i,t}^p$  and  $\Sigma_{i,t}^p$  are the mean state vector and covariance matrix of the EKF for the  $i$ -th landmark and  $w_t^p$  is a weight. RB-PFs are updated as follows:

- Propagation of the robot pose  $\xi_T$ .
- For each landmark  $l^i$ : Given  $\xi_T, o_T^i$ , update EKF that corresponds to  $l^i$  (update step from section II.A).
- Assign weights to particles and resample the population.

We have implemented two alternatives for propagating the pose and reweighting the particles in RB-PFs. The first and simplest one is the “odometry-based” approach, where we use only odometry information for pose propagation  $P(\xi_T^p | \xi_{T-1}^p, u_{T-1})$  and the weights are updated according to  $q_T^p \propto P(o_T | \xi_T^p, m_{T-1}^p)$ . On the other hand, the “sensor-based” solution is optimal in terms of minimizing the variance of the weights and takes the observation into account for the pose propagation  $P(\xi_T^p | \xi_{T-1}^p, u_{T-1}, o_T)$ , and sets the weights as follows  $q_T^p \propto P(o_T | \xi_T^p, m_{T-1}^p)$ . The two approaches correspond to algorithms FastSLAM 1.0 and FastSLAM 2.0 [20], respectively. Our implementation is similar to these two algorithms but without the data association components.

**2. Robust:** Our experiments suggested that some of EKF’s drawbacks, such as sensitivity to erroneous initialization, negatively effect the performance of RB-PFs as well. Another EKF limitation is its inability to handle erroneous observations. Given these observations we propose a complete RB-PF-based framework for robust, online, bearing-only SLAM which is shown in Fig. 4. The first alteration compared to typical RB-PFs is that we replace EKFs with GSFs. We have already seen that GSFs can be used to approximate a uniform distribution in the depth of the landmarks. For outlier rejection we have implemented an unsupervised algorithm for Single-Cluster Graph Partitioning (SCGP) [12]. The online version of the algorithm considers at each step a small set of observations per landmark, usually the last ones, and attempts to estimate whether the latest observation for each landmark is an outlier or not. In order to do so, each observation is considered as a node in a graph. An edge in this graph has a weight of 0 if the observation rays corresponding to the two nodes intersect and 1 otherwise, defining in this way the graph’s adjacency matrix. Then the technique computes the dominant eigenvector of the graph’s adjacency matrix and interprets the vector’s values as indicators of whether the corresponding observations are inliers or not. If the value corresponding to the last observation is below an automatically computed threshold then it is considered an outlier. SCGP is an approximate  $O(N^2)$  solution to outlier rejection and has been shown by Olson et al. [12] to be more effective than RANSAC in various settings including range-only SLAM.

The new “Robust” algorithm employs a training period  $t \in [1 : T_1]$  during which each landmark is tracked by a GSF instead of an EKF and the RB-PF follows the “Odometry-Based” approach for propagating the pose and reweighting the particles. After the training period the GSF is pruned and only one of the Gaussian hypothesis is kept, the one with the highest likelihood. During the regular operation of the algorithm the landmark estimates are considered trustworthy enough so that the “Sensor-based” approach is used for the proposal distribution. Note that during the training period any operation on the GSF (e.g., computing the likelihood of the GSF given the last observation), can be redirected to the EKF with the highest likelihood.

## IV. EXPERIMENTAL SETUP

We have created a simulator to test the performance of the algorithms we have described in a controlled environment under the following parameters.

**1. Environment:** The simulated environment is a rectangular region containing the robot and landmarks. We have considered two types of scenes: sparse (5 landmarks) and dense scenes (100 landmarks). The landmarks are dispersed uniformly at random inside the rectangular region.

**2. Sensing model:** We compute an observation as the actual bearing to a landmark given the robot pose and we add noise so as to simulate the sensor’s uncertainty. We apply three different noise models: (a) low Gaussian noise  $\Gamma_{low}(0, \sigma_{low})$ , where we add to the true observation a noise value selected from a Gaussian distribution  $\Gamma(x, \sigma_{low})$  with mean  $x$  and st.

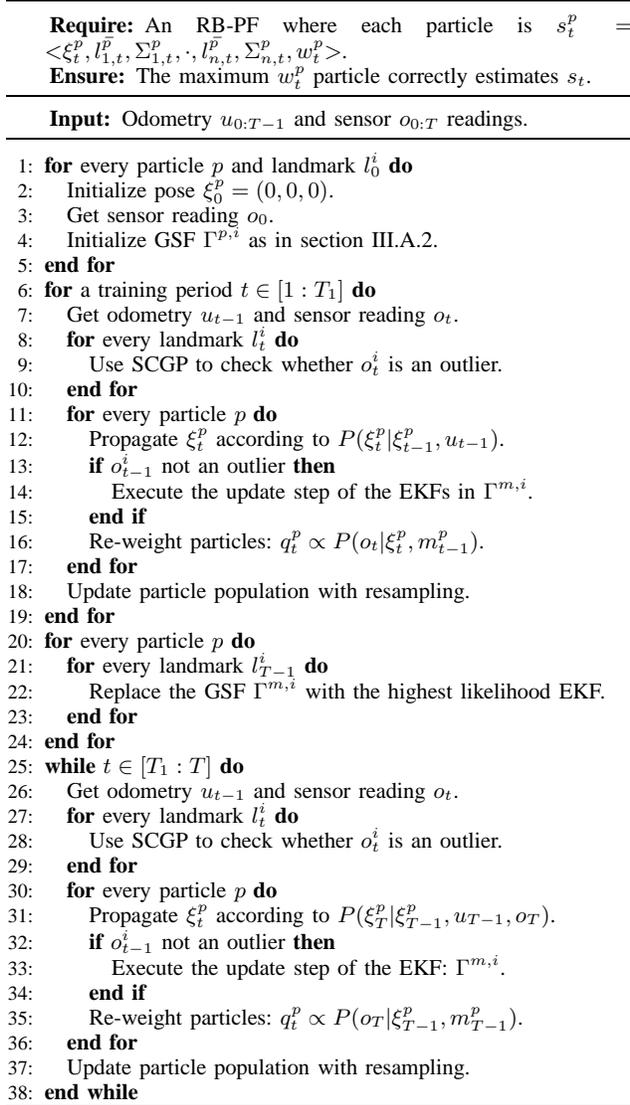


Fig. 4. Robust Rao-Blackwellized Particle Filter Algorithm.

dev.  $\sigma_{low} = 0.2^\circ$  (b) high Gaussian noise  $\Gamma_{high}(0, \sigma_{high})$  ( $\sigma_{high} = 1^\circ$ ) and (c) the “random” noise model, where measurements are produced from the low Gaussian noise model but 20% of them are discarded and replaced by bearings produced uniformly. We are not focusing on the data association issue in our experiments since landmarks correspond to unique identities. One of the purposes, however, of the random noise model is to simulate the effects of having problematic data association. Although observations due to wrong data association do not necessarily follow a uniformly random distribution, this error model is considered to create similar difficulties to SLAM algorithms as failed data association [7]. Furthermore, we assume that each observation returns a bearing to every landmark present in the environment.

**3. Robot’s path:** During each run, the robot senses the environment on prespecified paths. These paths are either loops (circular or rectangular) or random paths. The random

paths are constructed so that the distance between two observations follows a Gaussian distribution with mean of 10 pixels and a st. dev. of 3. The difference in orientation is centered at zero and has a st. dev. of  $10^\circ$ . The robot does not exit the bounding rectangle that defines the environment. If it collides with the boundary it bounces back towards the scene. The error model for the odometry uses the assumption that between two observations the robot moves forward a certain distance and then rotates by some angle. Both actions are represented independently with Gaussian distributions. The st. dev. corresponding to the rotational term is the same as the st. dev. of the sensing model. The translational st. dev. is 1 pixel in the case of  $\Gamma_{low}$  and 3 in the case of  $\Gamma_{high}$ .

## V. RESULTS

To test an algorithm from section III, we run it on 50 different scenes for each noise setting (low Gaussian, high Gaussian, 20% random noise), landmark density (sparse and dense) and robot path type (loop or random path). At the end of each run we record average localization and mapping errors along with the runtime. We have defined the following criteria to distinguish the performance of the algorithms. (a) *Robustness*: the percentage of successful runs. A run is considered successful when the localization and mapping errors are below two empirically chosen thresholds (100 pixels and 200 pixels respectively). We followed this definition of success because we observed a binary behavior: An algorithm will either fail resulting in very high errors or it will succeed in reconstructing the scene fairly accurately. (b) *Time efficiency*: the average amount of time it takes to run an algorithm in a particular setting.

### A. Robustness

We will first present a comparison of the various alternative approaches within each family of algorithms, EKF, IML and RB-PFs, in terms of robustness.

1) *EKF solutions (Fig 5)*: The straightforward implementation of the EKF satisfactorily solves most problems with sparse scenes under Gaussian noise model but its performance deteriorates with denser scenes. This is due to the landmark initialization issues of the algorithm. When there are only a few landmarks, the mapping errors do not propagate fast enough to the localization estimate allowing for an eventual convergence in the mapping estimates as more observations

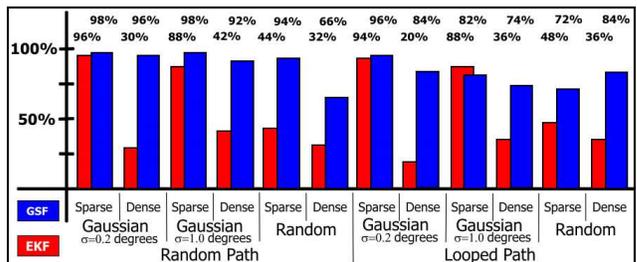


Fig. 5. Performance of the EKF approaches.

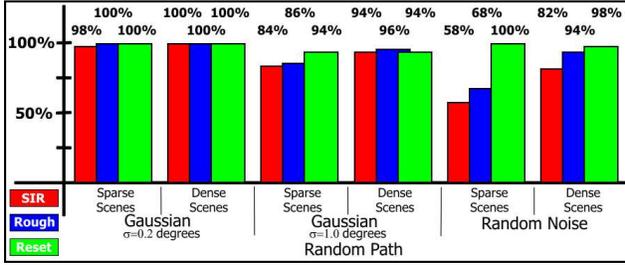


Fig. 6. Performance of the IML approaches.

are coming. In dense scenes the influence of wrong landmark estimates is stronger. Furthermore, the algorithm cannot handle the “random” noise model although we have also used the Single-Cluster Graph Partitioning (SCGP) algorithm for outlier rejection. Initializing landmarks with a GSF considerably improves the performance of the EKF overall, although the effect is less obvious with random noise. This is not surprising since a GSF is compromised from multiple EKFs and a small set of outliers is enough to cause an EKF to lose track of the landmark’s state completely.

2) *IML solutions (Fig 6)*: We have applied the SIR algorithm both for the pose and the mapping filters and the results show the weakness of the IML approach in looped paths. The benefits of applying PFs are obvious since the algorithm is able to solve many runs under the random noise model that the EKF solutions could not, with the best performance for random paths and high density scenes. We have also noticed a degradation of performance as we increase the st. dev. in the Gaussian noise models. We experimented with multiple PF resampling algorithms but we were not able to improve the performance of the IML algorithm when applied them for the pose PFs. However, there was an improvement when we used techniques that allow the state of the mapping filters to change. By applying “Rough” we have managed to get results comparable to the GSF solution for the low Gaussian noise model, while solving many random noise model problems. The technique which exhibited the best behavior, however, was “Reset”. Although it still suffers from the loop problem, it has produced impressive results in the case of random noise. The results actually suggest that the perturbation in the input due to the random noise improves the performance of the algorithm compared to the high Gaussian noise model.

3) *RB-PF solutions (Fig 7)*: The family of RB-PF algorithms performed the best overall, exhibiting very high robust-

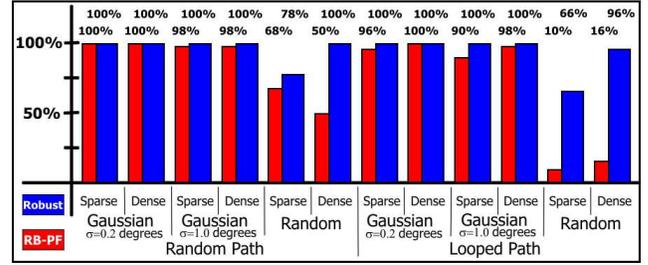


Fig. 7. Performance of the RB-PF approaches.

ness for all the scenes and paths when a Gaussian noise model was applied. Fig. 7 shows the robustness levels of two RB-PF methods: (a) an implementation that uses the “Odometry-Based” approach and initializes landmark estimates with a single Kalman filter (RB-PF) and (b) an implementation that follows the description in Fig. 4 (Robust). In particular, for the Gaussian noise model, the various RB-PFs approaches exhibit small differences in performance with a small advantage for the methods that use a GSF to initialize landmarks and for the “Sensor-based” approach. Nevertheless, RB-PFs face considerable challenges in the presence of random noise and this is again mainly due to the inability of the Kalman filter to handle random noise.

The advantage of using the “Robust” algorithm as described in Fig. 4 lies in the case of outliers. The typical RB-PF approach manages to solve only 10-16% of the problems with the random noise model and for looped paths. With the changes we have incorporated in the algorithm, the performance in this case reaches between 66-96%. Notice that at the same time that the “Robust” framework manages to solve every experiment under the Gaussian noise models. Furthermore, among all the RB-PF based algorithms we have tried the “Robust” framework has achieved the best accuracy levels.

From the comparison of all the methods, EKF, IML and RB-PF based, it is obvious that different approaches face difficulties with different parameters of the experimental settings. For example, loops are problematic for IML algorithms, dense scenes are challenging for EKF, while sparse scenes pose more difficulties for the typical RB-PF algorithm under the random noise model. Nevertheless, it is apparent that the “Robust” RB-PF-based framework we have proposed in this paper has the most satisfactory behavior across all parameter settings. It must also be noted, however, that the “Reset” IML algorithm is the best solution for sparse scenes when the sensor follows the random noise model.

### B. Efficiency

Table I shows the average amount of time in milliseconds that each method spends per step. The reported performance has been achieved on a dual AMD Athlon 1900MPs with one gigabyte of memory each. All of the nodes ran Debian Linux with kernel 2.4.21. We already know that the computational complexity of the Kalman filter is cubic so we expect that it is going to be slow for densely populated scenes,

rendering it an inappropriate online algorithm for the SLAM problem when there are hundreds of landmarks. The results support this expectation. The EKF, and consequently the GSF solution as well, are the slowest algorithms for scenes with hundreds of landmarks. The GSF is slightly faster because during the training period, there is no high dimensional EKF matrix to be updated which reduces the average delay per step. The IML solutions are more than three times faster for the densely populated scenes compared with the EKF approach. Of course, the time efficiency of PFs depends upon the number of particles. In our experiments we've used 1000 particles both for pose and mapping PFs. Note, that even in the case of sparse environments the amount spent by the PF algorithms is noticeable, mainly due to the cost of updating the pose PF. Finally, the RB-PF based algorithms are much faster than the other approaches. In particular, the typical RB-PF is up to 27 times faster than the EKF solution. The "Robust" algorithm is a little bit slower due to the call to the outlier rejection process, the Gaussian sum filter and mainly due to the use of the "Sensor-based" approach instead of the "Odometry-based" one. "Robust" is still 10 times faster than the EKF solution and is possible to be called at a rate of 13 times per second even in the presence of hundreds of landmarks.

TABLE I  
EFFICIENCY

Millisec/Step	EKF	GSF	SIR	Reset	RBPF	Robust
Sparse	0.15	0.15	12.41	17.24	1.09	6.97
Dense	733.42	708.14	204.54	295.86	26.72	73.52

## VI. DISCUSSION

We have performed in this paper an extensive experimental comparison of state-of-the-art techniques for the online, bearing-only SLAM problem in a controlled environment and for various parameters concerning landmark density, robot path and sensor characteristics. We have focused on three classes of algorithms that are applicable to this problem and which are popular in the related literature: EKF, IML and RB-PF based methods. The RB-PF approach exhibited the best performance in our experiments but seemed to perform poorly in the presence of outliers. In order to overcome this problem we proposed a combination of RB-PF with an online outlier rejection policy and a GSF for more robust landmark initialization. The overall algorithm performed optimally in our experiments for a noise model that followed a Gaussian distribution and exhibited considerably improved performance in the presence of outliers compared to a typical RB-PF implementation.

There are many interesting problems related to bearing-only SLAM and which can be the focus of future research. For example, an important issue with bearing-only SLAM is how to control the motion of a robot in order to improve the quality of the localization estimate [14]. Furthermore, achieving bearing-only SLAM without odometry is very important for localizing cameras that are not mounted to a robot but are carried by a human operator. However, in this case the solution can be computed only up to a scale factor. Another interesting

challenge is the case of moving targets. Then instead of computing only a two-dimensional state for each landmark, the algorithms must estimate a four-dimensional state, one that includes velocities as well. These moving targets could be members of a robotic team that attempts to achieve relative self-localization using cameras. Finally, a related and equally important problem to bearing-only SLAM is the range-only counterpart which has also received attention recently [11].

**Acknowledgements:** Lydia Kavradi and Kostas Bekris are supported in part by NSF-0308237 and NSF-0205671. Max Glick is supported by NSF-0308237 and a fellowship from the Brown School of Engineering at Rice University.

## REFERENCES

- [1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb. 2002.
- [2] A. Costa, G. Kantor, and H. Choset. Bearing-only landmark initialization with unknown data association. In *ICRA-04*, pages 1764–1769, New Orleans, LA, April 2004.
- [3] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *9th ICCV*, volume 2, pages 1403–1410, October 13-16 2003.
- [4] M. Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In *ISER-00*, 2000.
- [5] A. Doucet, N. De Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001.
- [6] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Uncertainty in AI*, 2000.
- [7] J.-S. Gutmann and D. Fox. An experimental comparison of localization methods continued. In *IROS-02*, volume 1, pages 454–459, EPFL, Switzerland, 30 Sept.- 5 Oct. 2002.
- [8] N. M. Kwok and G. Dissanayake. Bearing-only slam in indoor environments using a modified particle filter. In *Australasian Conference on Robotics and Automation*, Brisbane, Australia, Dec. 1-3 2003.
- [9] N. M. Kwok, G. Dissanayake, and Q. P. Ha. Bearing-only slam using a sprt based gaussian sum filter. In *ICRA-05*, pages 1121–1126, Barcelona, Spain, April 2005.
- [10] T. Lemaire, S. Lacroix, and J. Sola. A practical 3D bearing-only SLAM algorithm. In *IROS'2005*, pages 2757–2762, Edmonton, Canada, 2-6 Aug. 2005.
- [11] P. Newman and J. Leonard. Pure range-only sub-sea slam. In *ICRA-03*, volume 2, pages 1921–1926, Taipei, Taiwan, Sept. 2003.
- [12] E. Olson, M. Walter, S. Teller, and J. Leonard. Single-cluster spectral graph partitioning for robotics applications. In *Robotics: Science and Systems, RSS-05*, MIT, Cambridge, MA, June 2005.
- [13] I. M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, Montreal, Quebec, Canada, 2002.
- [14] R. Sim. Stable exploration for bearings-only slam. In *ICRA-05*, pages 2422–2427, Barcelona, Spain, April 2005.
- [15] R. C. Smith and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *IJRR*, 5(4):56–68, 1987.
- [16] S. Thrun and A. Zelinsky. Accurate local positioning using visual landmarks from a panoramic sensor. In *ICRA-02*, pages 2656–2661, Washington, DC, May 2002.
- [17] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *IJRR*, 20(5):335–363, 2001.
- [18] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. 2002.
- [19] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *AI*, 128(1-2):99–141, 2000.
- [20] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 2004. to appear.