A Modified Particle Filter for Simultaneous Robot Localization and Landmark Tracking in an Indoor Environment

Wenyan Hu, Tom Downs, Gordon Wyeth, Michael Milford and David Prasser
School of Information Technology and Electrical Engineering
University of Queensland, St. Lucia 4072, Brisbane, QLD, Australia
{wenyan, td, wyeth, milford, prasserd }@itee.uq.edu.au

Abstract

This paper presents the implementation of a modified particle filter for vision-based simultaneous localization and mapping of an autonomous robot in a structured indoor environment. Through this method, artificial landmarks such as multi-coloured cylinders can be tracked with a camera mounted on the robot, and the position of the robot can be estimated at the same time. Experimental results in simulation and in real environments show that this approach has advantages over the extended Kalman filter with ambiguous data association and various levels of odometric noise.

1 Introduction

A key prerequisite for a truly autonomous robot is that it can simultaneously localize itself and accurately map its surroundings [Kortenkamp *et al.*, 1998; Dissanayake *et al.*, 2001]. The problem of achieving this, which is known as Simultaneous Localization and Mapping (SLAM), has received considerable attention in the past two decades.

One of the first successful attempts at the SLAM problem was introduced by Smith and Cheeseman [Smith and Cheeseman, 1986] who proposed a mathematical formulation of the approach that is still in widespread use today. The paper proposed using the extended Kalman filter (EKF) for incrementally estimating the posterior distribution over robot pose while also estimating the positions of landmarks. Recent work has focussed on scaling this approach to larger environments with more than a few hundred landmarks [Masson *et al.*, 2002] and to algorithms for handling the data association problem [Liu and Thrun, 2003].

One of the limitations of the EKF is that it requires that features in the environment be uniquely identifiable, which is a consequence of the Gaussian noise assumption. If this requirement cannot be satisfied, one has to employ an alternative method such as the particle filter for a better representation of the probability density function.

Particle filters[Doucet *et al.*, 2001], which are Sequential Monte Carlo methods, provide an attractive simulation-based approach for updating distributions in the light of new data. Early successes of particle filters can be found in the area of robot localization, in which a robot's pose has to be recovered from sensor data [Dellaert *et al.*, 1999]. More recently, particle filters have been at the core of solutions to higher dimensional robot problems such as SLAM, which, when phrased as a state estimation problem, involves a variable number of dimensions. Murphy and colleagues adopted Rao-Blackwellized particle filters [Murphy, 1999; Murphy and Russell, 2001] as an effective way of representing alternative hypotheses on robot paths and associated maps. [Montemerlo *et al.*, 2002; Montemerlo and Thrun, 2003] extended this method to efficient landmark-based SLAM using Gaussian representations of the landmarks and were the first to successfully implement it on real robots.

In this paper we present an investigation into the use of a pan-tilt camera mounted on a mobile robot for simultaneous localization and mapping in a structured indoor environment with coloured cylinders as landmarks. Following [Murphy and Russell, 2001], our approach applies a Rao-Blackwellized particle filter to estimate a posterior of the path of the robot, in which each particle has associated with it an entire map. The distributions of landmarks are also represented by particle sets, where separate particles are used to represent the robot and the landmarks. This increases the computational load but the method is still applicable in real-time. The key advantage of this method is that the full posterior over robot poses and maps can be nonlinearly approximated at every point in time by particles. Our practical implementation also shows it can avoid rapid convergence of the particles to the maximum likelihood state.

Measure and motion models in our implementation differ from those in[Montemerlo *et al.*, 2002] because we use a camera to provide measurements instead of laser scans. We also consider the ambiguities in data association and the effect of the robot path on SLAM performance. Results are compared with those of the EKF method applied to the same robot in the same environment and indicate superior performance by the modified particle filter method.

The paper is organized as follows. In the next section, the SLAM problem is briefly reviewed, and then in section 3 the basic particle filter method is introduced. Section 4 describes a modified particle filter and section 5 provides a detailed description of its implementation. Experimental results and discussions are presented in section 6 and 7 respectively with conclusion in section 8.

2 The SLAM Problem

In robot SLAM, a state vector contains the *pose* of the robot (its location and orientation) relative to its environment, along with the location of landmarks in the robot's proximity. In what follows, we will represent the vector of state variables at time *t* by x_t. SLAM addresses situations in which state variables are not observable directly. In such situations, the robot has to rely on information obtained from sensor and robot motion. For convenience, let us assume that the information is collected in the two sets of variables:

$$z^t := \{z_1, \cdots, z_t\}$$
$$u^t := \{u_1, \cdots, u_t\} \quad (1)$$

where z_t denotes a sensor measurement taken at time *t*, and u_t specifies the robot motion command asserted in the time interval $[t-1, t)$. The goal of SLAM is to estimate the posterior probability over the state variable *x* at time *t*, written $p(x_t \mid z^t, u^t)$, which can be calculated via the following Bayesian recursive equation (see [Thrun, 2000] for a derivation):

$$p(x_t \mid z^t, u^t) = \eta p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1}) p(x_{t-1} \mid z^{t-1}, u^{t-1}) dx_{t-1} \quad (2)$$

Here η is a normalizing constant. The probability $p(z_t \mid x_t)$ is often referred to as the measurement or perceptual model in robotics, while probability $p(x_t \mid u_t, x_{t-1})$ is usually referred to as the motion model. These two models will be discussed in a later section.

3 Basic Particle Filter

The key idea of the particle filter is to approximate the posterior $p(x_t \mid z^t, u^t)$ by a set of sample states $\{x_t^{[i]}\}$, or particles. Here each $x_t^{[i]}$ is the i^{th} of *N* state samples, where *N* is the size of the particle filter. The particle filter algorithm can be stated as follows:

1. Initialise *N* particles randomly.
2. Apply the motion model to each particle.
3. Predict the observation for each particle; calculate the likelihood (weights) from the measured value.
4. Select (re-sample) the particles that best explain the observation according to their weights.

The above algorithm indicates the simplicity of the particle filter. There is no need to assume Gaussian noise or to perform a linearization as is required in the EKF. The method can, however, run into difficulties. If an insufficient number of particles is employed, there may be a lack of particles in the vicinity of the correct state, leading to divergence of the filter. This is known as the *depletion problem* [Hahnel *et al.*]. Thus, the number of particles must be large enough to allow the posterior distribution to be captured and this required number increases geometrically with the number of system states. This requirement places limitations on what can be achieved in real time.

Another difficulty, called the *impoverishment problem*, can arise in the re-sampling step unless suitable precautions are taken. This is because resampling involves the selection of particles that best explain the observations. This is done by selecting them with probabilities proportional to the perceptual model $p(z_t \mid x_t)$ and, with stationary landmarks, the particles can quickly converge to a single point giving only a suboptimal result. Impoverishment can also arise due to the lack of independence of measurements taken when a robot is stationary, but this is less of a problem in our case because our robot is continuously moving, albeit slowly. Our method, though it cannot solve these problems entirely, reduces the impact of convergence problems.

4 Modified Particle Filter

When particle filters are applied in robot SLAM, the state variable x_t contains two quantities that influence sensor measurements over time: the map (namely landmark position in our research) and robot's pose in the environment. Therefore, if *m* represents *K* landmark positions and *s* the robot's pose, equation (1) can be expressed as follows:

$$p(s_t, m \mid z^t, u^t, n^t) = \eta p(z_t \mid s_t, m, n_t) \int p(s_t \mid u_t, s_{t-1}) p(s_{t-1}, m \mid z^{t-1}, u^{t-1}, n^{t-1}) ds_{t-1} \quad (3)$$

where $n_t \in \{1, \cdots, K\}$ is the index of the landmark perceived at time *t*. For mathematical convenience, this paper assumes that landmarks are uniquely identifiable, and that the number of landmarks *K* is known.

Unfortunately, computation of the full posterior in equation (3) is not tractable in general and, as stated above, we here employ the Rao-Blackwellized particle filter approach [Murphy and Russell] which provides an efficient way of estimating the posterior. The key idea of this approach is to solve the recursive Bayes filter update by the following equation:

$$\begin{aligned} p(s_t, m \mid z^t, u^t, n^t) &= p(m \mid s^t, z^t, u^t, n^t) p(s^t \mid z^t, u^t, n^t) \\ &= p(s^t \mid z^t, u^t, n^t) \prod_{i=1}^{N} p(m_i \mid s^t, z^t, u^t, n^t) \quad (4) \end{aligned}$$

Here the SLAM problem has been decomposed into a robot path estimation problem and a collection of landmark estimation problems that are conditioned on the robot path estimation. By doing this, dependencies between the estimate of robot pose and the landmark location estimates are fully accounted for, while the complexity of the estimation algorithm in the number of landmarks *N* remains linear [Montemerlo *et al.*, 2002]. The robot path posterior $p(s^t \mid z^t, u^t, n^t)$ is represented by a set of particles, and the distributions $p(m_i \mid s^t, z^t, u^t, n^t)$ are represented by particle sets, where each set is attached to one particular robot particle. This sequence in which the calculations are carried out will now be discussed in detail.

4.1 Particle Filter Path Estimation

First of all a plain particle filter is employed for estimating the path posterior $p(s^t \mid z^t, u^t, n^t)$ in (4). The path posterior is denoted by S_t and each particle $s_t^{[i]} \in S_t$ represents an estimate of the robot's path:

$$S_t = \{s^{t,[i]}\} = \{s_1^{[i]}, s_2^{[i]}, \cdots, s_t^{[i]}\}_{i=1,\cdots,M} \qquad (5)$$

where the superscript $[i]$ refers to the *i*-th particle in the set and *M* is the number of particles.

Following [Montemerlo *et al.*, 2002] the particle set is calculated incrementally using the set S_{t-1}, the control u_t and the measurement z_t. This is done by using each particle in S_{t-1} to generate a probabilistic guess at the robot's pose at time t:

$$s_t^{[i]} \sim p(s_t \mid u_t, s_{t-1}^{[i]}) \qquad (6)$$

by sampling from the motion model. This gives a "temporary" set of M particles. Assuming the set S_{t-1} is distributed according to $p(s^{t-1} \mid z^{t-1}, u^{t-1}, n^{t-1})$, the temporary set is distributed according to $p(s^t \mid z^{t-1}, u^t, n^{t-1})$. The new set S_t is then obtained by sampling from the temporary set. Each particle $s^{t,[i]}$ is drawn with a probability proportional to an "importance factor" whose calculation is explained in Section 4.3.

4.2 Particle Filter Landmark Location Estimation

A particle filter is next employed in estimating landmark location. Since this estimate is conditioned on the robot pose, the landmark particle filters are attached to individual pose particles in S_t. The full posterior over paths and landmark positions can be represented by the sample set

$$S_t = \{s^{t,[i]}, m_1^{[i][j]}, m_2^{[i][j]}, \cdots, m_k^{[i][j]}\}_{\substack{i=1,\cdots,M \\ j=1,\cdots,N}} \qquad (7)$$

where $m_k^{[i][j]}$ is *k*-th landmarks pose, the superscript notation $[j]$ refers to the *j*-th particle in the landmark particle set; *N* is the number of particles in each of landmark particle sets.

The posterior over the *k*-th landmark pose m_k depends on whether or not the landmark was observed. If the landmark is observed, we obtain:

$$p(m_{k=n_t} \mid s^t, z^t, u^t, n^t) \\ \propto p(z_t \mid m_{n_t}, s_t, n_t)\, p(m_{n_t} \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \qquad (8)$$

If landmark *k* is not observed, leave the Gaussian unchanged:

$$p(m_{k\neq n_t} \mid s^t, z^t, u^t, n^t) = p(m_{k\neq n_t} \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \qquad (9)$$

The probabilities in (8) and (9) can be used to estimate landmark positions in more than one way. [Montemerlo *et al.*, 2002] used an EKF to obtain this estimate. In this paper we instead resample from the landmark particle sets using a set of importance weights in the usual way. These weights are proportional to the probability ratio below [Metropolis *et al.*, 1953]:

$$w_t^{[j]} \propto \frac{p(m_{n_t}^{[j]} \mid s^t, z^t, u^t, n^t)}{p(m_{n_t}^{[j]} \mid s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1})} \qquad (10)$$

and from (8), this can be written

$$w_t^{[j]} \propto p(z_t \mid m_{n_t}^{[j]}, s_t, n_t) \qquad (11)$$

$p(z_t \mid m_{n_t}^{[j]}, s_t, n_t)$ can be easily calculated using a likelihood function.

4.3 The Importance Weights for Path Estimation

Now that the landmark locations have been estimated, the importance weights for resampling from the robot path particles can now be derived, again using [Metropolis *et al.*, 1953]. This gives

$$w_t^{[i]} \propto \frac{p(s^{t,[i]} \mid z^t, u^t, n^t)}{p(s^{t,[i]} \mid z^{t-1}, u^t, n^{t-1})} \qquad (12)$$

And, using Bayes theorem and the Markov property

$$w_t^{[i]} = \int p(z_t \mid m_{n_t}^{[i]}, s_t^{[i]}, n_t)\, p(m_{n_t}^{[i]})\, dm_{n_t} \qquad (13)$$

And (13) is easily cauculated since $p(z_t \mid m_{n_t}^{[i]}, s_t^{[i]}, n_t)$ was already approximated using a particle filter, in contrast to [Montemerlo *et al.*, 2002],who used the EKF. $p(m_{n_t}^{[i]})$ is the Gaussian posterior.

4.4 Algorithm Process

Our algorithm for this approach proceeds as follows:

1. Initialise N_r particles representing robot pose with normally distributed random numbers around the start position, each particle being a 3 by 1 state vector consisting of a position and orientation. Initialise *M* sets of N_l particles representing *M* landmarks, each particle being a 2 by 1 state vector initially set to random positions within the environment.
2. Apply the motion model to each of the particles created in step 1.
3. For each particle representing robot pose,
    a. Predict the observation for each particle in the particle sets representing landmark position; calculate the likelihood (weights) of particles from the measured value
    b. Select (re-sample) the particles that best explain the observation according to their likelihoods.
4. For all particles representing robot pose, calculate the weights from measured values and estimated landmark position.
5. Re-sample the particles representing robot pose that best explain the observation according to their

likelihoods.
6. Go to step 2.

Four issues must be resolved when we implement SLAM using the particle filter. The forms of the measurement model and the motion model mentioned earlier need to be fixed because these two models describe the response and actions of the particle filter to sensor observations. A procedure for data association must also be established to maintain the correspondence between a measurement and a landmark. In addition, appropriate robot behaviours such as path control must be developed for SLAM to operate successfully. Details on the implementation of these requirements are given in the following section.

5 Implementation Details

5.1 Motion model

The motion models $p(s^t \mid z^t, u^t)$ and $p(m_i \mid s^t, z^t, u^t)$ predict the movement and status over time of the robot and its landmarks. When a control *u*, consisting of forward and angular velocity is applied to the robot, we employ the well-known transition equations to predict the robot moves[Dissanayake *et al.*, 2001]:

$$\begin{bmatrix} x_r^{[i]}(t) = x_r^{[i]}(t-1) + v\Delta T\cos(\phi^{[i]}(t-1) + \gamma(t-1)\Delta T) + w_x \\ y_r^{[i]}(t) = y_r^{[i]}(t-1) + v\Delta T\sin(\phi^{[i]}(t-1) + \gamma(t-1)\Delta T) + w_y \\ \phi^{[i]}(t) = \phi^{[i]}(t-1) + \gamma(t-1)\Delta T) + w_\phi \end{bmatrix} \quad (14)$$

where $(x_r^{[i]}(t-1), y_r^{[i]}(t-1))$ and $\phi^{[i]}(t-1)$ is the robot's location and bearing at time *t-1*, for all robot particles $i = 1,\cdots,N_r$; v is the velocity, γ is the angular velocity, ΔT is the time step and $w_{xy\phi}$ are noise terms of the form $N(0,\sigma_{xy\phi})$ which lump together the effects of un-modelled characteristics such as control response and wheel slip. Figure 1 shows a schematic diagram of the robot in the process of observing a landmark.

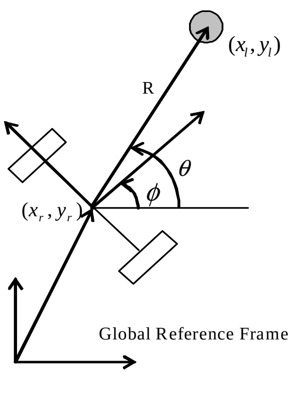

Figure 1: Robot and observation kinematics

Since the landmarks in the environment are stationary cylinders, the motion model is as follows:

$$\begin{aligned} x_l^{[j]}(t) &= x_l^{[j]}(t-1) \\ y_l^{[j]}(t) &= y_l^{[j]}(t-1) \end{aligned} \quad (15)$$

where $(x_l^{[j]}(t-1), y_l^{[j]}(t-1))$ is the location of landmark *l* at time *t*-1, for all landmarks $l = 1,\cdots,M$ and landmark particles $j = 1,\cdots,N_l$. The landmark particles are initialised to values lying in the region of uncertainty around the positions given by camera measurements.

5.2 Measurement model

When using a camera as a measuring device, we can obtain an approximate range and bearing for a landmark. Using the approach of [Prasser and Wyeth, 2003] we can obtain projected error in both range and bearing. The range error is larger than for a laser range finder but, as we show here, SLAM based on a single camera is still feasible.

The measurement model can be written as [Dissanayake *et al.*, 2001]

$$\begin{bmatrix} R = \sqrt{(x_r - x_l)^2 + (y_r - y_l)^2} + w_r \\ \theta = \arctan(\dfrac{x_r - x_l}{y_r - y_l}) - \phi + w_\theta \end{bmatrix} \quad (16)$$

where (x_r, y_r) is the robot's location and ϕ is its bearing; (x_l, y_l) is the landmark's location; w_r and w_θ are the noise sequences associated with the range and bearing measurements.

The following equation is used as a likelihood function to evaluate the likelihood of the observation given the state represented by each particle

$$w_i \propto \frac{1}{\sqrt{2\pi(\sigma_r^2 + \sigma_\theta^2)}} \exp(\frac{-(R-\hat{R})^2}{\sigma_r^2} + \frac{-(\theta-\hat{\theta})^2}{\sigma_\theta^2}) \quad (17)$$

where $\hat{R}$, $\hat{\theta}$ are the expected range and bearing for a particle, and σ_r, σ_θ are the range and bearing uncertainty.

5.3 Data Association

Data association concerns the correspondence between a measurement and a landmark. In this study, we assume that all cylinders in the environment are distinguishable (that is, the same coloured cylinder won't appear in more than one location), therefore simplifying the data association problem. However, since our vision system is trained to recognize rectangular areas of solid uniform colour, it occasionally picks up background objects or gives indistinguishable information about a cylinder's colour. In this case the EKF method generally breaks down. Our modified particle filter approach is more likely to recover, due to its use of resampling. When a new measurement comes in, our particle filter method seeks matching particles from formerly memorized cylinders and calculates their matching degree (weight). Indistinguishable information usually has very small weight and is ignored during resampling.

5.4 Robot behaviours

In order to implement robot SLAM successfully, we have to design suitable robot behaviours for exploration. Such behaviours might based on wandering with obstacle avoidance, or more specifically designed to explore frontiers of mapped areas. In this study, a simple behaviour that executes a circular path around the area under investigation is used.

6 Experimental Results

The experimental results were gathered in two parts: simulated results and real robot results. The simulated results are useful to verify the operation of the robot against ground truth, and to run the experiment over longer durations. The real robot results show that the system readily translates from the high fidelity simulation to the real robot.

6.1 Experimental Setup

The experiments are performed on a Pioneer2-DXE robot from ActivMedia incorporating a 400 MHz AMD K6-2 processor. Vision processing and motor control are performed on the on-board computer, while a 2.6GHz laptop connected to the robot by a wireless link provides the main processing power for the SLAM software. A Sony PTZ colour camera mounted at the front of the robot with an effective field of view of about 40 degrees is used for detecting the visual landmarks.

The simulator used for these experiments is based on the high fidelity simulator used in the RatSLAM project[Milford *et al*., 2004]. The simulator is a "plug-in" replacement for the Pioneer, where the wireless connection that normally connects to the Pioneer's on-board computer is instead connected to another computer that runs the simulation. In the simulation experiments, the simulator was given a systematic odometry error that increased over time. The simulated vision system also reported landmarks with uncertainty characteristic of the real vision system.

The test environment is a robot laboratory with limited space and a wide variety of objects in the room, show in Figure2. The coloured cylinders are scattered around the

Figure 2: Pioneer 2 DXE robot in test environment. Note the coloured cylinders that are used as visual landmarks.

robot in a hexagon occupying about 3 by 3 metres. The colour, distance and bearing of cylinders can be obtained through the vision system which recognizes rectangular areas of solid colour [Prasser and Wyeth, 2003]. In order to compare our method with EKF, we employ up to 6 cylinders with different colour combinations so that they are uniquely identifiable by the vision systemthese experiments, the number of time steps for both methods is set to 400 and the PF method employs 100 by 1000 particles, which means 100 particles for estimation of robot path, and 1000 particles for estimation of each landmark's position. The number of particles for the former is far less than the latter because the estimation uncertainty for the robot path is much less than for the landmark's position since we assume the robot's start position is known as (0,0).

6.2 High Fidelity Simulation Results

The performance of the two algorithms, the modified particle filter and the extended Kalman filter, were compared by running a simulation of the system to be tested over a prolonged period. Note that this test would not be practical with the real robot at this point, as we have not developed behaviours that would provide a path that reliably visits each landmark – a shorter real robot test is described below. It is important to note that the simulator captures the uncertainty of the vision process, and has an artificial odometry error.

Figure 3 shows the effect of the odometry error. Without re-localization, the robot localization error increases in an unbounded fashion. Both the algorithms succeed in bounding the localization error, with the particle filter doing better at representing the absolute localization of the robot. The reason for this is best shown in Figure 4, which shows the estimates of the positions of the landmarks. The particle filter rapidly converges to the correct solution, while the extended Kalman filter retains some error in the measurement.

6.3 Real Robot Results

Video1 shows the estimation procedure by using both the EKF and the PF method in a continuous circular path. The robot path view window has been enlarged 5 times more than the two bottom windows to make sure it can be read clearly. The white line is the robot odometry path, while the magenta line and green line describe the estimated robot path with PF and EKF methods respectively. The estimated cylinders' positions are expressed with stars in the window of Estimated Landmarks View by EKF and dots (particles) in the window of Estimated Landmarks View by PF. The uncertainties of estimation are described with ellipses. The white crosses in both windows are the cylinders' real positions measured manually. Although our vision system cannot provide precise geometric information about distance and bearing to all objects, both EKF and PF methods can still be implemented on a real robot in real time simultaneously.

The ground truth of the robot path was measured manually by stopping the robot and manually measuring its position on the floor; this process was subject to significant error. Figure 5 shows the robot path obtained from odometry, EKF, PF and the ground truth measurements in 50 time steps. Figure 6 shows the estimated landmarks position by EKF and PF methods in real world. These figures indicate comparable results to the simulation work.

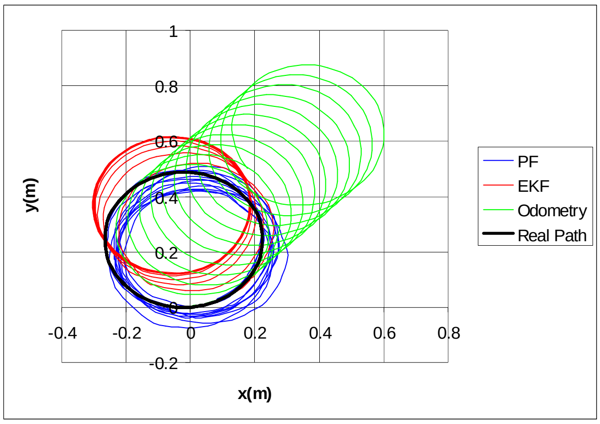

Figure 3: Comparitive localization paths of the robot comparing the particle filter with the extended kalman filter. The real path and the path derived from pure odometry are also shown.

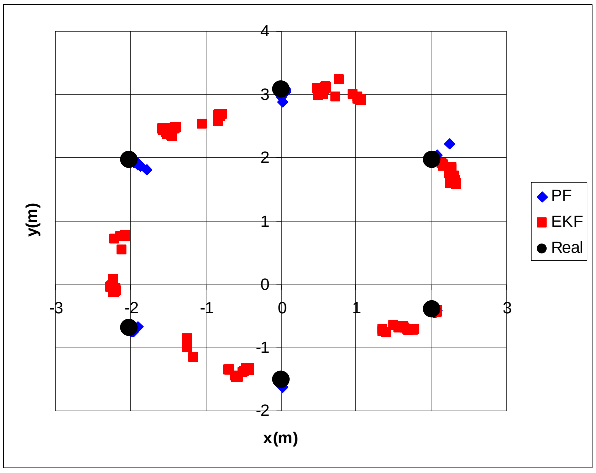

Figure 4: Comparison of map built of landmarks using the particle filter and extended Kalman filter approaches against the real positions of the landmarks.

7 Discussion

These preliminary results show the promise of the modified particle filter method. In particular, the method has been shown to perform as well (if not outperform) the well-established EKF approach to SLAM. The most promising aspect though is the robustness of the method with respect to data association. During experimental testing, vision errors would occasionally cause a cylindrical marker to be misclassified as being of the wrong colour. Such a misclassification proved disastrous for the EKF system, whereas the PF system was able to recover by filtering misclassified information during re-sampling. However, the modified PF method increases the computation, which will be a challenge when doing SLAM in a larger environment in the future.

8 Conclusions

A modified particle filter SLAM algorithm has been outlined in this paper. Preliminary experimental results in a

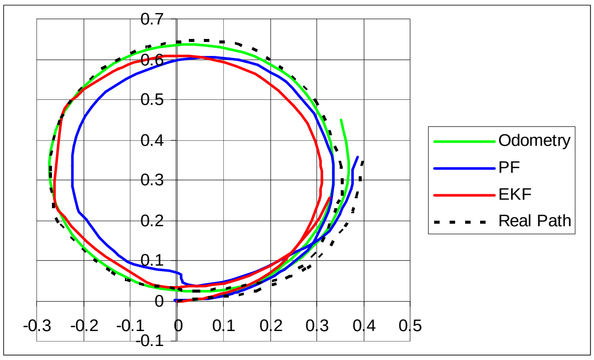

Figure 5: Comparison of SLAM methods against the odometry and ground truth measurements.

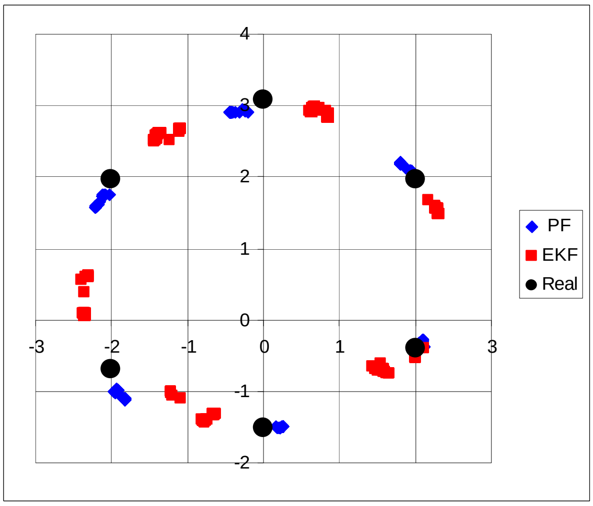

Figure 6: Comparison of landmark measurements for particle filter and extended kalman filter techniques against the actual landmark position.

small area for this method have been obtained on a real robot in real time, and compared with results obtained using an EKF method. These preliminary results show promising performance, particularly in light of the problems associated with dealing with ambiguous landmarks using the extended Kalman filter method.

Future work on this project will test performance over a larger area, using low level behaviours that enable autonomous exploration. The system could also be coupled with the natural landmark system developed in our laboratory[Prasser and Wyeth, 2003]. It would also be interesting to produce results that could be compared directly with the RatSLAM algorithm also under investigation in our laboratory.

References

[Dellaert *et al.*, 1999]Dellaert, Fox, Burgard and Thrun. Monte Carlo localization for mobile robots. *Proceedings of International Conference on Robotics and Automation. vol.2 10-15 May 1999 Detroit, MI, USA [IEEE Robotics & Autom. Soc]*. 1999 Dept. of Comput. Sci. Carnegie Mellon Univ. Pittsburgh PA USA.

[Dissanayake *et al.*, 2001]Dissanayake, Newman, Clark, Durrant-Whyte and Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE-Transactions-on-Robotics-and-Automation. June 2001; 17(3): 229-41*. USA, 2001 Mech. & Mechatronic Eng. Sydney Univ. NSW Australia.

[Doucet *et al.*, 2001]Doucet, De Freitas and Gordon, Doucet, De Freitas and Gordon. *Sequential Monte Carlo Methods in Practice*. Springer.

[Hahnel *et al.*, 2003]Hahnel, Burgard, Fox and Thrun. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2003

[Kortenkamp *et al.*, 1998]Kortenkamp, Bonasso and Murphy. AI-based Mobile Robots: Case studies of successful robot systems. Cambridge, MA, MIT Press.

[Liu and Thrun, 2003]Liu and Thrun. Results for outdoor-SLAM using sparse extended information filters. *Proceedings of IEEE International Conference on Robotics and Automation.* Taipei, 2003

[Masson *et al*., 2002]Masson, Guivant and Nebot. Hybrid architecture for simultaneous localization and map building in large outdoor areas. *IROS 2002: IEEE/RSJ International Conference on Intelligent Robots and Systems. vol.1 30 Sept.-5 Oct. 2002 Lausanne, Switzerland [IEEE Robotics & Autom. Soc.; IEEE Ind. Electron. Soc.; Robotics Soc. Japan; Soc. Instrum. & Control Eng.; INRIA Rhone-Alpes Grenoble; EPFL Lausanne]*. 2002

[Metropolis *et al*., 1953]Metropolis, Rosenbluth, Rosenbluth, Teller and Teller. "Equations of state calculations by fast computing machine." *Chemical Physics* 21(1953

[Milford *et al*., 2004]Milford, Wyeth and Prasser. RatSLAM: a hippocampal model for simultaneous localization and mapping. *Proceedings. ICRA '04. 2004 IEEE International Conference on Robotics and Automation, 2004*. New Orleans, LA, USA, 2004

[Montemerlo and Thrun, 2003]Montemerlo and Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. *Proceedings of IEEE International Conference on Robotics and Automation*. Taipei, 2003

[Montemerlo *et al*., 2002]Montemerlo, Thrun, Koller and Wegbreit. "FastSLAM: A factored solution to the simultaneous localization and mapping problem." *AAAI* 2002

[Murphy, 1999]Murphy. "Bayesian map learning in dynamic environments." *NIPS* 1999

[Murphy and Russell, 2001]Murphy and Russell. Rao-blackwellized particle filtering for dynamic bayesian networks. Sequential monte carlo methods in practice. Springer.

[Prasser and Wyeth, 2003]Prasser and Wyeth. Probabilistic visual recognition of artificial landmarks for simultaneous localization and mapping. . *Proceedings. ICRA '03. IEEE International Conference on Robotics and Automation, 2003*. Taipei, 2003

[Smith and Cheeseman, 1986]Smith and Cheeseman. "On the Representation and Estimation of Spatial Uncertainty." *Int. J. Robotics Research, 5(4)* 1986

[Thrun, 2000]Thrun. Probabilistic algorithms in robotics. 2000 Dept. of Comput. Sci. & Robotics Carnegie Mellon Univ. Pittsburgh PA USA.