A Comparative Analysis of Particle Filter Based Localization Methods

Luca Marchetti, Giorgio Grisetti, and Luca Iocchi

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113 00198 Rome Italy
{firstname.lastname}@dis.uniroma1.it

Abstract. Self-localization is a deeply investigated field in mobile robotics, and many effective solutions have been proposed. In this context, Monte Carlo Localization (MCL) is one of the most popular approaches, and represents a good tradeoff between robustness and accuracy. The basic underlying principle of this family of approaches is using a Particle Filter for tracking a probability distribution of the possible robot poses.

Whereas the general particle filter framework specifies the sequence of operations that should be performed, it leaves open several choices including the observation and the motion model and it does not directly address the problem of robot kidnapping.

The goal of this paper is to provide a systematic analysis of Particle Filter Localization methods, considering the different observation models which can be used in the RoboCup soccer environments. Moreover, we investigate the use of two different particle filtering strategies: the well known Sample Importance Resampling (SIR) filter, and the Auxiliary Variable Particle filter (APF).

1 Introduction

The knowledge of the pose and the orientation of a mobile robot in its operating environment is of utmost importance for an autonomous robot. Therefore self-localization is a well known problem in mobile robotics, and many effective solutions have been proposed. The presence of an initial position guess about robot position determines a distinction between the *position tracking* and the *global localization* problems. The prototype of the algorithms for the position tracking problem is Kalman Filter localization [6], while global positioning encloses common frameworks like Multi Hypotheses Localization, Histogram Filters and Particle Filters [1]. In the last years Particle Filter Localization, also known as Monte Carlo Localization (MCL), became one of the most popular approaches for solving the global localization problem.

Whereas the implementation of a particle filter for localization is straightforward, its performance is strongly affected by the modeling of the process to estimate. Namely the user has to specify the system *motion model*, that is the

G. Lakemeyer et al. (Eds.): RoboCup 2006, LNAI 4434, pp. 442–449, 2007.

probability distribution of successor states conditioned to the odometry readings, and the *observation model* that describes the likelihood of a given observation given the current robot position.

In the RoboCup Four-legged league the localization problem becomes a challenging task, because of the following reasons: i) the only sensor that can be used for acquiring measures of the environment is a low resolution and low quality camera; ii) the robot motion is affected by a considerable amount of noise due to both the presence of opponents in the field of play and to the poor accuracy of the odometry; iii) the computational power available for localization is rather limited.

The dynamic environment strongly violates the Markov assumption which underlies most of the approaches proposed in literature. In order to cope with such violations, several extensions have been proposed to the original Bayes formulation of the localization problem. To this end the most popular technique is known as *sensor resetting* [5]. It consists in bootstrapping the estimator with hypotheses based on the raw observations.

The goal of this paper is to present a systematic analysis performed on the Particle Filter localization, when considering the different observation models which can be used in the RoboCup Four Legged league contexts. Moreover, we investigate the use of two different particle filtering strategies: the well known Sample Importance Resampling (SIR) filter [2], and the Auxiliary Variable Particle filter (APF) [8].

Localization based on APF has been previously proposed by Vlassis *et al.* [9] for solving the vision based localization problem, together with a nonparametric estimate of the likelihood function. The main focus of their work is on how to compute a satisfactory nonparametric estimate of the direct observation model $p(x|z)$, expressing the probability of being in a given location x given the observed panoramic camera image z. Such a distribution is expressed through a Gaussian mixture learned from the data.

In contrast to [9], the contribution of our work is to investigate possible variants of the particle based localization algorithms, for parametric (feature based) observation models, treating APF as an additional degree of freedom.

2 Particle Filtering

One of the most popular algorithms used in localization is the so-called Monte Carlo Localization introduced by Dellaert *et al.* [1]. The core idea of the algorithm is to estimate a robot pose distribution using a particle filter. The system state is represented through a set of samples in the robot pose space $\left\{x^{(i)} = (\mathbf{x}, \mathbf{y}, \theta)^{(i)}\right\} \in \Re^2 \times [0 \ldots 2\pi)$.

$$p(x) \simeq \frac{1}{N} \sum_{i=0}^{N} \delta_{x^{(i)}}(x)$$

here $\delta_{x^{(i)}}(\cdot)$ is the impulse function centered in the sample $x^{(i)}$. The denser are the samples in a state space region the higher the probability that the robot is in that region.

Ideally one wants to sample from the posterior distribution

$$x_t^{(i)} \sim p(x_t|z_{0:t}, u_{0:t})$$

but this is not possible in the general case because such a distribution is not available in closed form. However b.

By representing the distribution through a set of weighted samples $< w^{(i)}, x^{(i)} >$, it is possible to estimate $p(x)$. The samples are drawn from a proposal distribution $q(x_t|z_{0:t}, u_{0:t})$, while the weights for each sample are computed according to the Importance Sampling Principle

$$w_t^{(i)} = \frac{p(x_t^{(i)}|z_{0:t}, u_{0:t})}{q(x_t^{(i)}|z_{0:t}, u_{0:t})} \quad (1)$$

By choosing the motion model $p(x_t|x_{t-1}^{(i)}, u_t)$ as proposal distribution, we can recursively compute the weights as

$$w_t^{(i)} \propto p(z_t|x_t^{(i)})\, w_{t-1}^{(i)}.$$

Two particle filtering techniques used in self-localization are the Sampling Importance Resampling (SIR) [2] and the Auxiliary Variable Particle Filter (APF) [8,9].

One of the main problems of the SIR algorithm is the *degeneracy problem* [8]. If the ratio between the variance of the proposal distribution and the observation model is high, it can happen that only a few samples generated in the sampling steps have a meaningful weight. The subsequent application of the resampling operation results in the suppression of most of the samples generated, because only those with greater weight are replicated, replacing the low-weighted particles. This fact reduces the chances that the filter achieves to a correct convergence because it impoverishes the set diversity.

The APF has been introduced to lessen the degeneracy problem. The key idea of this algorithm is to select the samples that will be propagated in the subsequent updated estimate. Such a selection is performed by evolving the current filter state using a reduced noise motion model, and evaluating the sample weights according to the Importance Sampling (IS) principle. Then a set of indices is sampled from the weights distribution, and only the surviving particles will be used for computing the updated particle generation, by means of the original motion model. This ensures an increased particle diversity, since the resampling is performed before evolving the filter.

More in detail the APF exploits the following factorization over the joint posterior of particle indices and robot poses

$$p(x_t, i|z_{0:t}, u_{0:t}) \propto p(z_t|\mu_t^{(i)})p(x_t|x_{t-1}^{(i)}, u_t).$$

Here μ_t is a mean, a mode or some other indicator of the predicted distribution, designed so that

$$p(i|z_{0:t}, u_{0:t}) \propto p(z_t|\mu_t^{(i)}).$$

Under the above hypotheses, we can sample from $p(x_t, i|z_{0:t}, u_{0:t})$ by sampling an index j with probability $\lambda^{(j)} = p(z_t|\mu_t^{(i)})$. We denote with j^i that the value of the j^{th} sampled index is the original index i. Subsequently, we can sample from the motion model $x_t^{(j)} \sim p(x_t|x_{t-1}^{(j^i)})$, according to the value of the state referred to by the drawn index.

According to the IS principle, the resulting weights will be:

$$w^{(j)} = \frac{p(z_t|x_t^{(j)})}{\lambda^{(j)}}$$

3 Algorithm Implementation

Given a filtering method, there are a number of implementation issues to be considered and several parameters to be tuned. The motion model depends on the robot kinematics and on the characteristics of the environment (i.e., the friction with the surface and the presence of collisions). The observation model depends on the characteristics of both the landmarks being observed and the sensor.

Motion Model. When the robot moves, its pose estimate should be updated according to the motion model, incorporating the relative movement u_t, estimated from the odometry. In the Sony AIBO, such a displacement can be obtained by taking into account the joints measures returned from the internal motor encoders. Inaccuracies in the model, as well as environment random phenomena (such as slipping and collisions), affect reliability and precision of measuring such a displacement.

A simple motion model can take into account such a noise by adding an amount of random Gaussian noise to odometry update:

$$x_t \sim x_{t-1} \oplus (u_t + \mathbf{e}_t)$$

where $\mathbf{e}_t$ is the random variable representing the noise affecting the odometry measure u_t, and $\oplus$ is the standard motion composition operator defined as in [7].

A more complex motion model can also consider noise depending on the relative motion of the robot

$$x_t \sim x_{t-1} \oplus (u_t + \alpha(u_t)\mathbf{e}_t)$$

where $\alpha(u_t)$ is a matrix of functions of the odometry motion, and $\mathbf{e}_t$ represents the noise affecting the movement for each time a unity distance is traveled. If α is a constant, the variance of the odometry error grows linearly with the distance traveled by the robot.

Finally, we can extend the previous model considering the presence of random collisions. When a robot hits an object or another robot, it is likely to stack, although the odometry measures a non zero displacement. We can take into account this phenomenon by including a prior of hitting an obstacle and marginalizing it out, as follows.

$$x_t \sim \mathbf{h}x_{t-1} + \neg\mathbf{h}(x_{t-1} \oplus (u_t + \alpha(u_t)\mathbf{e}_t))$$

where $\mathbf{h}$ is a binary random variable that takes into account the probability of hitting an obstacle.

3.1 Observation Model

The environment of RoboCup Four-Legged League includes a set of features that are normally used in localization: unique colored beacons (or markers), unique colored goals, and white lines on the green carpet.

Since the beacons are of limited size they are usually entirely contained in an image. The vision system on the robot can estimate the position of these beacons in the robot reference frame. With these preconditions, developing a localization algorithm should be straightforward, however the noise affecting both the robot sensing and the robot motion, as well as the dynamic environment turns this task into a challenging one.

In particular, the low resolution of the camera, combined with motion blurring effects caused by the robot movements, affects the reliability and precision of feature detection. This is particularly evident when a landmark located quite far away from the robot is perceived. In this case a beacon occupy only a few pixels in the image (as few as 10), and the estimation process is likely to fail.

An adequate observation model for a generic landmarks takes into account these phenomena is

$$p(z|x) = p(z_e) + \sum_i p(z_i|x)$$

here, $p(z|x)$ is the probability of the reading z given the robot pose x. It can be expressed as the conjunction of the following disjoint events:

- the reading is generated by a spurious reading with probability $p(z_e)$ or,
- it is due the landmark γ_i, with probability $p(z_i|x)$.

More in detail, $p(z_i|x)$ can be expressed as

$$p(z_i|x) = p(z|\gamma_i, x)p(\gamma_i|x)$$

here, $p(z|\gamma_i, x)$ is the probability of making the observation z given that it originates from landmark γ_i and the robot is in x, and $p(\gamma_i|x)$ is the prior of perceiving the landmark γ_i from the location x.

The measurement probability is a function of the angular (δ_α) and linear (δ_ρ) distance between the expected and the measured landmark locations. For every type of landmark (beacon b_i, goal post p_i and line l_i) we use these equations:

$$p(z|b_i, x) \propto e^{(\delta_\alpha^2/\sigma_\alpha^2)} e^{(\delta_\rho^2/\sigma_\rho^2)}$$
$$p(z|p_i, x) \propto e^{(\delta_\alpha^2/\sigma_\alpha^2)}$$
$$p(z|l_i, x) \propto e^{(\delta_\theta^2/\sigma_\theta^2)} e^{(\delta_\rho^2/\sigma_\rho^2)}.$$

In the previous equations we use α denote angular measurements, and ρ indicates the distance. For lines across the fields (and corners, that are intersection of lines), we assume that they are expressed in polar coordinates, so ρ and θ indicates the hough parameters of a single detected line.

For a single beacon observation b_i we use both angle and distance, while for single goal post p_i we only use the measured angle. This is because goal posts are not recognized exactly and they are less reliable. Each σ inside equation is specific to type of observation and are note the same.

The probability for the single observation models are then combined to form definitive probabilities

$$p_{particles} = \prod p(z|b_i, x) \prod p(z|p_i, x) \prod p(z|l_i, x)$$

4 Experiments

In this section we present the results of localization experiments for the methods described in the previous section.

The experiments have been performed using standard Four-Legged soccer field as used in RoboCup 2005 and RoboCup 2006. The scenario is more challenging for localization task with respct to RoboCup 2004 setting (having a smaller field and six beacons instead of four), where experiments in [3] have been performed. We also use two different strategies to track real robot position: external camera to track real position of robot and ground truth made using some measured spot and a smoothing technique. For acquiring a smooth ground truth of robot position we process off-line log data captured from the robot, as follows:

- we take a log containing images and internal information from the robot;
- we manually mark every feature on the images in the log;[1]
- we use the true perceptions to generate a log with reliable and more precise measures about distances and angles for objects (these represent the real ground truth of perception);
- we iterate the localization task several times on same input: first iteration the localization is performed in normal way. The second iteration starts using the last set of particles and use log backward from last to first frame. Running localization many times we obtain a smoothed ground truth.

The same log (with ground truth on perception) is then used to run localization methods (with different parameters) several times and to compare different methods on the same input. Moreover, by using the external camera view and the ground-truth of robot path, we can measure absolute errors in localization.

The subsequent path graphs use these conventions: green is used for field lines, cyan for ground truth, red for SIR and blue for APF.

[1] These data sets have been used also for evaluating color segmentation and image processing and are available from `www.dis.uniroma1.it/~spqr`

Random walk without kidnapping. In the first experiment we consider the behavior of SIR and APF using 100 particles, when the robot is in a random walk, without obstacles. In Fig. 1, the left graph shows the result with sensor resetting disabled, while the right one shows the behavior of sensor resetting. While the behaviour of the two filters is similar in the first case, the SIR particle filter converges faster and recover more precisely the position with sensor resetting enabled, while the APF is less robust and more spikes are present in the calculated path.

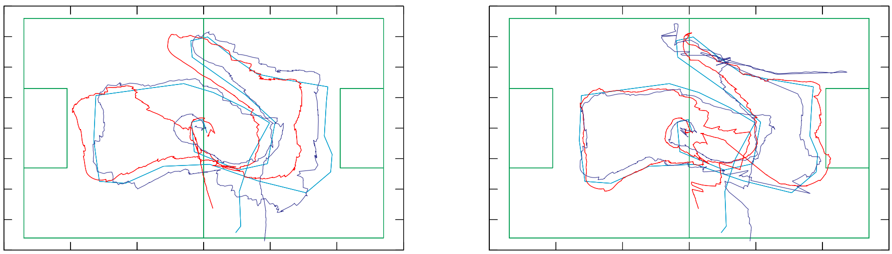


Fig. 1. Random walk with sensor resetting disabled (left) and enabled (right)

Random walk with kidnapping. The second experiment considers another random walk but with a teleporting of robot to simulate the kidnapping problem. The kidnapping was performed from (a) to (b), shown in Fig. 2. Again the SIR filter using sensor resetting recovers from erroneous positions quickly, while the APF changes more slowly.

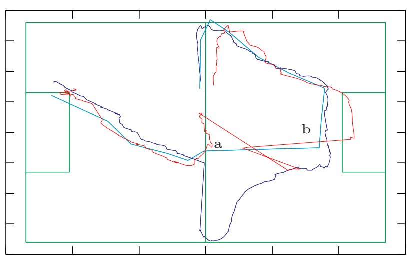
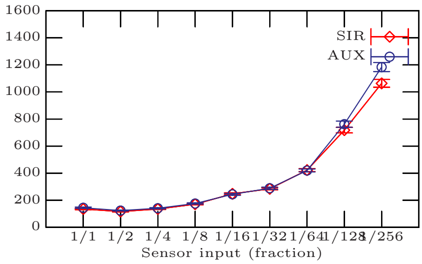


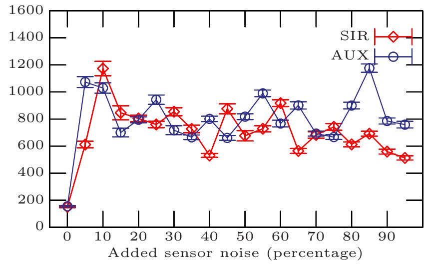
Fig. 2. Second experiment: random walkwith kidnapping from a to b

Playing mode. The third experiment has been done in a typical playing sequence. When robot is in playing state, it interacts with other robot, competing for the ball, occasionally kicking the ball itself. In such situations the odometry give very noisy information to robot. Therefore, sensor resetting needs to be enabled.

Following [3], we performed experiments varying the observation frequency (up to a fraction of 1/256) and adding Gaussian noise (approximately 15% of the real distance and 10 degrees for the angle). As shown in Figure 3 the difference between the two methods are negligible and both degrades its performance in a similar way as in [3].



a)



b)

Fig. 3. Performance of SIR and AUX compared a) when using only a fraction of observations and b) when additional gaussian noise is present

5 Conclusion

In this paper we have illustrated the performance of SIR-based particle filter and Auxiliary Particle Filter, to accomplish a localization task based on visual landmarks with AIBO robots. The results show that both the method are suitable for this task, and that SIR is slightly better in combination with sensor resetting techniques that are needed in the RoboCup soccer scenario.

Additional results, including the use of datasets used in [3] and [4], are provided in a technical report available from `www.dis.uniroma1.it/~spqr`

References

1. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: Efficient position estimation for mobile robots. In: Proc. of the 16th National Conference on Artificial Intelligence (AAAI99) (1999)
2. Gordon, N., Salmond, D., Ewing, C.: A novel approach to nonlinear nongaussian bayesian estimation. In: Proceedings F., pp. 107–113 (1993)
3. Gutmann, J.S., Fox, D.: An experimental comparison of localization methods continued. In: In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2002)
4. Jensfelt, P., Kristensen, S.: An experimental comparison of localizationmethod, the mhl sessions. In: IROS (2003)
5. Lenser, S., Veloso, M.: Sensor resetting localization for poorly modelled mobile robots. In: Proceedings of International Conference on Robotics and Automation (ICRA'00) (2000)
6. Leonard, J.J, Durrant-White, H.F.: Mobile robot localization by tracking geometric beacons. Transactions on Robotics and Automation 7, 376–382 (1991)
7. Lu, F., Milios, E.: Robot pose estimation in unknown environments by matching 2D range scans. Journal of Intelligent and Robotic Systems 18, 249–275 (1997)
8. Pitt, M., Shephard, N.: Filtering via simulation: Auxiliary particle filters. Journal of the American Statistical Association 94(446), 590–599 (1999)
9. Vlassis, N., Terwijn, B., Kröse, B.: Auxiliary particle filter robot localization from high-dimensional sensor observations. In: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA) (2002)