

A FastSLAM algorithm based on the Unscented Filtering with adaptive selective resampling

Manuel Cugliari, Francesco Martinelli

► **To cite this version:**

Manuel Cugliari, Francesco Martinelli. A FastSLAM algorithm based on the Unscented Filtering with adaptive selective resampling. 6th International Conference on Field and Service Robotics - FSR 2007, Jul 2007, Chamonix, France. Springer, 42, 2007, Springer Tracts in Advanced Robotics. <inria-00258750>

HAL Id: inria-00258750

<https://hal.inria.fr/inria-00258750>

Submitted on 25 Feb 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A FastSLAM algorithm based on the Unscented Filtering with adaptive selective resampling^{*}

Manuel Cugliari and Francesco Martinelli

DISP, Università di Roma Tor Vergata, via del Politecnico, I-00133 Roma, Italy
mcugliari@tiscali.it, martinelli@disp.uniroma2.it

Summary. A FastSLAM approach to the SLAM problem is considered in this paper. An improvement to the classical FastSLAM algorithm has been obtained by replacing the Extended Kalman Filters used in the prediction step and in the feature update with Unscented Kalman Filters and by introducing an adaptive selective resampling. The simulations confirm the effectiveness of the proposed modifications.

1 Introduction

The Simultaneous Localization and Map Building (SLAM) problem has recently received a large attention (see, e.g., [4, 13]). The SLAM problem consists in a mobile robot moving in an unknown environment, which attempts to estimate its own position and to realize a spatial map. The environment is described through a set of natural landmarks extracted by the robot from the surrounding.

The SLAM can be faced as a state estimation problem, where the state includes the robot and the landmark coordinates, and can be approached with the well-known Extended Kalman Filter (EKF) [1]. EKF-based SLAM algorithms have two limitations: they present a high computational complexity and are prone to the data association problem when, like in this paper, the correspondence between measurements and landmarks is assumed unknown. An Unscented Kalman Filter (UKF) based solution to the SLAM problem [10] has been recently proposed. The UKF has been introduced in the context of non-linear system filtering by [7, 8]. Even if more performant, the resulting approach is similar to the EKF-SLAM since the conditional independence among the features position estimates given the robot path (introduced by the FastSLAM approach) is not exploited. The FastSLAM approach [11, 12] exploits this independence: it uses a modified particle filter to estimate the robot pose and each particle is equipped with N EKF to estimate the position of the N landmarks. The resulting algorithm is an application of the Rao-Blackwellized particle filter [5]. The FastSLAM has a lower complexity w.r.t. the EKF-SLAM and is more

^{*} This work has been partially supported by MIUR under grant PRIN 2005092439. The authors would like to thank F. Romanelli and L. Spinello for their valuable initial support.

robust regarding the data association problem since each particle performs its own data association. Nevertheless, the robot pose estimated by the classical FastSLAM is based on the EKF and may incur into divergence due to the approximations introduced by linearization. The idea of replacing EKF with UKF to improve the proposal distribution of a particle filter has been introduced in [15] in a general framework. In addition, the use of UKF to cope with nonlinearities associated with a stereo camera observation model was exploited in [2], where a FastSLAM algorithm was developed in an experimental context. In our paper, the idea of using the Unscented Transformation (UT) both in the observation and in the motion model has been combined with an adaptive selective resampling, based on an adaptive threshold, which significantly improves the selective resampling described in [5, 6], where a fixed threshold was considered. With these modifications, the robot pose error is drastically reduced (around one order of magnitude in our simulation results) and a more consistent map is obtained with respect to the classical FastSLAM 2.0 algorithm described in [12].

2 SLAM and FastSLAM Background

The SLAM problem can be correctly described through a Markov chain or, more in general, through a dynamic Bayesian network. The robot pose s_t changes according to a probabilistic law $p(s_t|u_t, s_{t-1})$ referred to as *motion model*, where $u_t \in \mathbb{R}^{n_u}$ is the control at time t . Following a common notation, the superscript t denotes a set of variables from time 1 up to time t , the subscript t indicates the value of the variable at time t . In planar SLAM $s_t = [x_t, y_t, \alpha_t]^T$, where x, y are the coordinates of the robot and α is its orientation. The environment is represented through N fixed landmarks ϑ_n , $n = 1, \dots, N$, described by their coordinates relative to the global coordinate frame. Proprioceptive data (like encoders) and exteroceptive data (like, e.g., lasers) are available to the robot which can sense landmarks and know their distance and bearing, relative to its local coordinate frame. z_t denotes a landmark measurement at time t and for mathematical convenience it is assumed that a single feature at a time is observed. These landmark measurements are noisy and a *measurement model* $p(z_t|s_t, \vartheta_{n_t}, n_t)$ is considered for them, with ϑ_{n_t} , $n_t \in 1, \dots, N$ being the particular landmark observed at time t . We assume in our formulation that data association is unknown. Both the *motion* and the *measurement models* are in general governed by non-linear deterministic functions, respectively h and g , affected by noise [14]:

$$p(s_t|u_t, s_{t-1}) = h(s_{t-1}, u_t + \delta_t) \quad (1)$$

$$p(z_t|s_t, \vartheta_{n_t}, n_t) = g(\vartheta_{n_t}, s_t) + \varepsilon_t \quad (2)$$

The noises δ_t and ε_t are assumed normally distributed with zero mean and covariance given, respectively, by P_t and R_t . With respect to [14], a direct effect of δ_t on u_t has been assumed. This is the case in several kinematic models and in particular for the one considered in Section 5. Formally stated, the SLAM problem consists in the determination of the location of all the landmarks ϑ_n and of the robot pose s_t only using the information of the controls $u^t = \{u_1, \dots, u_t\}$ and the measurements $z^t = \{z_1, \dots, z_t\}$. That is, it consists in the determination of the posterior

$$p(s_t, \Theta | z^t, u^t, n^t) \quad (3)$$

where $\Theta = \{\vartheta_1, \dots, \vartheta_N\}$ is the set of all the landmarks. The FastSLAM approach heavily relies on the conditional independence present in the SLAM problem which allows to factorize the posterior over the robot path as follows [14]:

$$p(s^t, \Theta | z^t, u^t, n^t) = p(s^t | z^t, u^t, n^t) \prod_n p(\vartheta_n | s^t, z^t, n^t) \quad (4)$$

A particle filter, similar to the *Monte Carlo Localization* (MCL) algorithm [3], estimates the robot path, maintaining a set S_t of M particles $s_t^{[m]}$, $m = 1, \dots, M$, which approximates the posterior $p(s^t | z^t, u^t, n^t)$. A temporary set of particles is generated at time t according to S_{t-1} , using the measurements z_t and the controls u_t . Since the set S_{t-1} corresponds to $p(s^{t-1} | z^{t-1}, u^{t-1}, n^{t-1})$, the new set is distributed according to $p(s^t | s^{t-1}, z^t, u^t, n^t)$. This distribution is often referred to as the *proposal distribution* of the particle filter. The new set is generated by sampling (with replacement) M particles proportionally to an *importance factor* $w_t^{[m]}$:

$$w_t^{[m]} = \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{p(s^t | z^t, u^t, n^t)}{p(s^t | s^{t-1}, z^t, u^t, n^t)} \quad (5)$$

This step, usually known as resampling process, accounts for the difference between the target and the proposal distribution. Next, the FastSLAM updates the posterior $p(\vartheta_n | s^t, z^t, n^t)$ conditioning it on the robot path. So, the full posterior over the robot path and the landmark position is approximated through the set:

$$S_t = \{s_t^{[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \dots, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]}\}_{m=1, \dots, M} \quad (6)$$

where $\mu_{n,t}^{[m]}$ and $\Sigma_{n,t}^{[m]}$ denote the mean and the covariance of the normal distribution representing the estimate provided by the m -th particle about the n -th landmark position. The FastSLAM performs the position estimate update of landmark n_t through the EKF while the position estimate of the other landmarks remains unchanged. Therefore there are $N \cdot M$ low dimensional EKF (one for each landmark relative to each particle). The FastSLAM samples the pose $s_t^{[m]}$ from the posterior:

$$s_t^{[m]} \sim p(s_t | s^{t-1, [m]}, z^t, u^t, n^t) \quad (7)$$

The posterior in (7) is approximated through the following expression [14], which allows to perform the sampling procedure:

$$s_t^{[m]} \sim p(s_t | s^{t-1, [m]}, z^t, u^t, n^t) \stackrel{\text{Bayes, Markov}}{=} \eta^{[m]} \int \left[p(z_t | \vartheta_{n_t}, s_t, n_t) p(\vartheta_{n_t} | s^{t-1, [m]}, z^{t-1}, n^{t-1}) \right] d\vartheta_{n_t} p(s_t | s_{t-1}^{[m]}, u_t) \quad (8)$$

In the remaining of the paper only the modified steps of the FastSLAM 2.0 algorithm will be discussed. The other steps can be found in [12] and [14].

3 The proposed unscented based improvements

We focus our attention on the rightmost term in (8) because an accurate approximation of this term drastically reduces the ambiguity in the data association.

Let Ξ_t be the set representing the evolution of the particles belonging to the set S_{t-1} at time $t - 1$, according to noisy u_t . Ξ_t has the following structure:

$$\Xi_t = \{\langle \xi_t^{[1]}, Q_t^{[1]} \rangle, \dots, \langle \xi_t^{[M]}, Q_t^{[M]} \rangle\} \quad (9)$$

where $\xi_t^{[m]}$ is the new m -th particle pose and $Q_t^{[m]}$ is the covariance matrix relative to the m -th particle. FastSLAM 2.0 computes Ξ_t by putting each particle $s_{t-1}^{[m]}$ into the motion model and taking the obtained value as the mean of a Gaussian distribution approximating the particle pose. That is²:

$${}^s\xi_t^{[m]} = h(s_{t-1}^{[m]}, u_t), \quad {}^sQ_t^{[m]} = G_{s_{t-1}} Q_{t-1}^{[m]} G_{s_{t-1}}^T + G_{u_t} P_t G_{u_t}^T \quad (10)$$

where $G_{s_{t-1}}$ and G_{u_t} are the Jacobians of h with respect to s_{t-1} and u_t . This linearization-based approach, which neglects the second and higher order terms of the Taylor expansion of $s_{t-1}^{[m]}$, sometimes produces uncorrect *clouds* of particles for the motion-only evolution. We attempt to provide a more accurate set Ξ_t exploiting the UT. First, the pose vector is augmented with the control noise vector $\delta_t \in \mathbb{R}^{n_u}$ to give an $n_a = 3 + n_u$ dimensional random vector $s_{t-1}^{a,[m]} := [s_{t-1}^{[m]}; \delta_t] \in \mathbb{R}^{n_a}$, with augmented covariance matrix:

$$Q_{t-1}^{[m]'} = \begin{bmatrix} Q_{t-1}^{[m]} & 0 \\ 0 & P_t \end{bmatrix} \quad (11)$$

The cross terms in $Q_{t-1}^{[m]'}$ are 0 since the pose $s_{t-1}^{[m]}$ and the noise δ_t are independent. Then, a set of (*sigma points*) $\{\mathcal{X}^{i,[m]}\}$, each $\mathcal{X}^{i,[m]} \in \mathbb{R}^{n_a}$, and corresponding weights $\{W^{i,[m]}\}$, $i = 0, 1, \dots, p$, are generated according to [7]: the weights $W^{i,[m]}$ can be positive or negative but, to provide an unbiased estimate, must sum to one. In addition, to match the mean and the covariance, it must be:

$$\sum_{i=0}^p W^{i,[m]} \mathcal{X}^{i,[m]} = \bar{s}_{t-1}^{a,[m]}, \quad \sum_{i=0}^p W^{i,[m]} (\mathcal{X}^{i,[m]} - \bar{s}_{t-1}^{a,[m]})(\mathcal{X}^{i,[m]} - \bar{s}_{t-1}^{a,[m]})^T = Q_{t-1}^{[m]'} \quad (12)$$

where $\bar{s}_{t-1}^{a,[m]} = \begin{bmatrix} s_{t-1}^{[m]} \\ 0 \end{bmatrix}$.

Next, the non-linear motion model is applied, in turn, to each point ($i = 0, \dots, p$) for the m -th particle to yield a cloud of transformed points $\mathcal{Z}^{i,[m]} \in \mathbb{R}^3$:

$$\mathcal{Z}^{i,[m]} = h(\mathcal{X}_{1:3}^{i,[m]}, u_t + \mathcal{X}_{4:n_a}^{i,[m]}) \quad (13)$$

² We denote the standard computation of $\xi_t^{[m]}$ by ${}^s\xi_t^{[m]}$, while we denote by ${}^u\xi_t^{[m]}$ the computation performed through the unscented approach. A similar notation applies to the covariance matrix.

where $\mathcal{X}_{r:j}^{i,[m]}$ denotes a vector comprising the components from r to j of $\mathcal{X}^{i,[m]}$. Now, according to [7], the predicted particle pose $\xi_t^{[m]}$ is given by the weighted average of the transformed points:

$${}^u\xi_t^{[m]} = \sum_{i=0}^p W^{i,[m]} \mathcal{Z}^{i,[m]} \quad (14)$$

and the covariance is the weighted outer product of the transformed points:

$${}^uQ_t = \sum_{i=0}^p W^{i,[m]} (\mathcal{Z}^{i,[m]} - {}^u\xi_t^{[m]})(\mathcal{Z}^{i,[m]} - {}^u\xi_t^{[m]})^T \quad (15)$$

After estimating the particles pose, FastSLAM 2.0 carries out the landmark location estimation by using a set of EKF. We rather provide an unscented-based feature update (UKF). The UT enables one to avoid the linearization of the perceptual model and the common problems that this solution involves. Details are omitted for space reasons. The update of the landmark location is capital for the correct data association and consequently for the estimated pose corrections.

4 The proposed adaptive selective resampling

Resampling is the step that has a major influence on the performance of the particle filter and it consists in replacing particles with low importance factor $w_t^{[m]}$ by particles with a higher weight. This process may erroneously eliminate good particles, causing the phenomenon known as particles impoverishment or *depletion*. To overcome this problem, it is important to determine *when* and *how* the resampling must be performed. Liu [9] introduced the effective number of particles

$$N_{eff} = \frac{1}{\sum_{m=1}^M (w^{[m]})^2} \quad (16)$$

which reflects the particles dispersion and consequently is also a measure of the approximation quality of the true posterior: it would be equal to the real number M of particles if the samples were drawn from the true posterior. Our approach extends the one reported in [5, 6] where the resampling is performed when the effective number N_{eff} of particles goes below a fixed threshold. A fixed threshold however does not take into account of the past evolution of N_{eff} , evolution which contains important information on the quality of the posterior approximation. For example, if the effective number of particles is not very high but it slightly oscillates in the time around a constant value, this probably corresponds to a good posterior approximation. In this case a resampling performed when one of the small oscillations of N_{eff} slightly goes below the threshold would be undermining. It would be better to decrease the threshold, avoiding the resampling, at least until this effective number remains above a minimum acceptable value. Also the average number of effective particles in a time

interval plays an important role in the characterization of the quality of the approximation over that interval. Therefore, instead of a static threshold, in this paper a dynamic threshold has been introduced, whose value depends on the past history of N_{eff} . The idea above can be formally stated as follows. We take into account of the

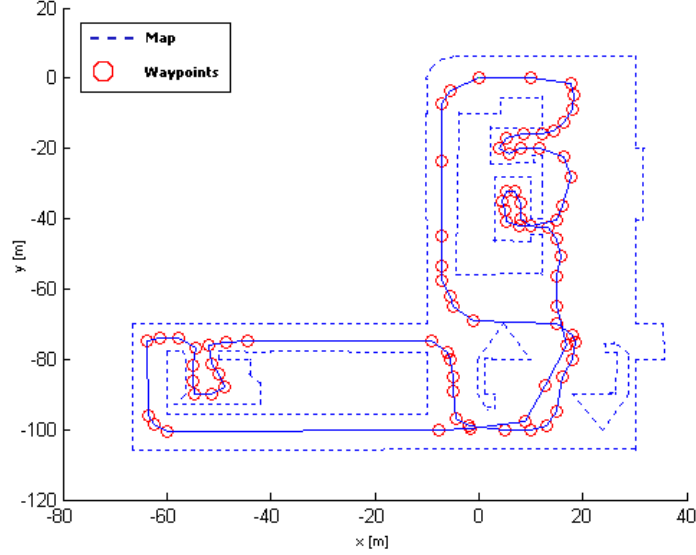


Fig. 1. Map (dashed black line) and waypoints (red circle). The robot controls are computed according to the waypoints. The blue solid line simply links the waypoints.

recent past of N_{eff} considering a window of observation comprising k time steps. First, denote with $v_{\tau-1}$ the value of the threshold in the window $\tau - 1$ and with N_{min} and N_{max} , respectively, the minimum and the maximum value that the threshold may assume. Our heuristic is to set $N_{min} = 0.6M$ and $N_{max} = 0.9M$. Observe that N_{max} must be taken below M since a threshold too near to M would imply a too large resampling frequency. The new v_{τ} is taken according to how rapidly N_{eff} changes and to how many effective particles are present in window $\tau - 1$. If k denotes the number of time steps in the observation window and $N_{eff}(t)$ is the effective number of particles at the t -th time step in the window, a measure α of the mean (normalized) number of particles in window $\tau - 1$ and a measure β of the (normalized) variation of N_{eff} can be expressed by:

$$\alpha = \frac{1}{Mk} \sum_{t=1}^{k-1} \frac{N_{eff}(t+1) + N_{eff}(t)}{2}, \quad \beta = \frac{N_{eff}(k) - N_{eff}(1)}{(k-1)(M - N_{min})} \quad (17)$$

Then, the following two quantities associated with α and β are computed:

$$N_\alpha = \alpha(N_{max} - N_{min}) + N_{min} \quad (18)$$

$$N_\beta = (N_{max} - N_{min})|\beta|^{\frac{1}{N_{min}}} + N_{min} \quad (19)$$

The new threshold v_τ can be obtained as a proper weighted average of N_α and N_β :

$$v_\tau = aN_\alpha + bN_\beta \quad (20)$$

with $a, b > 0$, $a + b = 1$, two suitable parameters. Notice that α near to one denotes a large average value of N_{eff} and the threshold can be increased. This is in fact obtained, since, when α approaches one, N_α moves linearly to N_{max} and the threshold is increased, according to (20). The non-linear function N_β takes into account of the variation of N_{eff} . Now, when β is near to zero, the variation of N_{eff} in the window is small. A low variation of N_{eff} , even if N_{eff} is not large, means an agreement among particles with respect to the approximation of the true posterior and the resampling should be avoided. This is in fact obtained since the threshold, according to (20), with a rate dependent on the exponent of $|\beta|$ in (19), is decreased. We choose a function like the one in (19) for its high variability near to zero. On the other hand, a $|\beta|$ quite greater than 0 corresponds to a large variation of N_{eff} in the window and the threshold is increased, making a resampling more likely to occur.

5 Simulation results

The robot considered in the simulations is characterized by a car-like kinematic model (e.g. [4]) and is equipped with a range-bearing laser with a maximum range of 20 meters and a 180 degrees frontal field-of-view. The indoor environment and the waypoints defining its target path are graphically depicted in Fig. 1. The robot controls (velocity and steering inputs) are generated in order to follow, from the initial pose $s_0 = \langle 0, 0, 0 \rangle$, the blue solid line in Fig. 1. The total traveled distance is about 700 m. Range and bearing readings, as well as the controls, are perturbed by Gaussian noises with zero mean and diagonal covariance matrices, respectively, R_t and P_t . If $u_t = [u_{t,v}; u_{t,s}]$ is the control vector, with $u_{t,v}$ and $u_{t,s}$ the velocity and the steering inputs, and $\delta_t = [\delta_{t,v}; \delta_{t,s}]$ the corresponding control noise vector, then $P_t = E[\delta_t \delta_t^T] = [\sigma_v^2, 0; 0, \sigma_s^2]$. In the simulations reported below, $\sigma_v = 0.01m/s$ and $\sigma_s = 0.017rad$. We have assumed, for the noise on the range and the bearing of each laser reading, a standard deviation respectively of $0.01m$ and of $0.017rad$.

The FastSLAM algorithm, with the modifications proposed in this paper (with parameters $a = 0.7$, $b = 0.3$, $k = 150$ and $p = 2n_a + 1 = 11$, where a and b are needed in (20), k is the number of steps in each observation window and p the number of Sigma points) was tested extensively in various conditions (high and low noises for the odometry and for the laser measurements, different parameters for the data association and, also, different environments). Our methods reduce the error between the real path and the estimated path (at each time step the particle with the greater weight is considered as the best estimate of the robot pose).

In Fig. 2 a comparison between the FastSLAM with and without modifications is illustrated for the robot coordinates $\langle x, y, \alpha \rangle$ error (Frames (a) and (b)) and for the

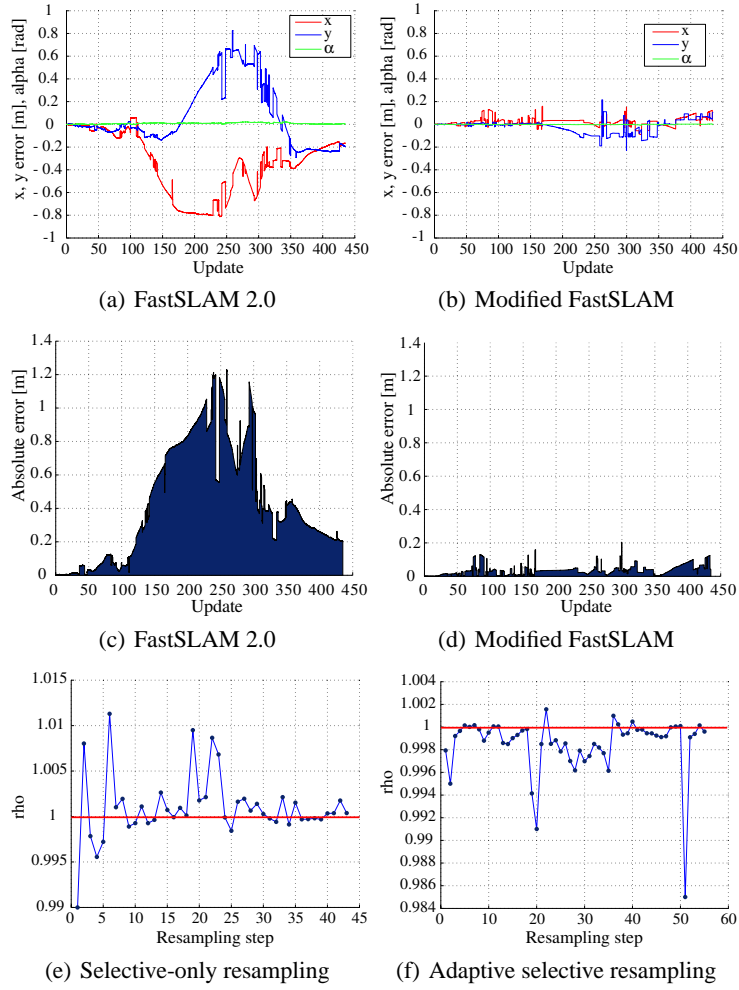


Fig. 2. Comparison of the standard FastSLAM and the FastSLAM with our modifications for the $\langle x, y, \alpha \rangle$ error ((a) and (b)) and for the pose error ((c) and (d)); (e) and (f) show the ratio ρ after the selective-only resampling step (e) and the adaptive selective resampling step (f).

absolute robot position error (Frames (c) and (d)). The robot, running the FastSLAM algorithm with the proposed modifications and with a number of particles $M = 60$, incurs in an average mean absolute error (MAE) (average computed over 50 simulation runs, with different noise realizations) of 0.03 m with standard deviation 0.007 m. With the standard FastSLAM (with the same number of particles and simulation runs) the average mean absolute error obtained is 0.62 m with standard deviation 0.02 m. The Frames (e) and (f) of Fig. 2 show the effectiveness of the adaptive selective resampling (Frame (f)) w.r.t. the selective resampling (Frame (e)) by reporting

the ratio ρ between the absolute position error when a resampling occurs and the absolute position error in the previous time step. That is:

$$\rho = \frac{e_{tr}}{e_{tr-1}} \quad (21)$$

where e_{tr} is the pose error after the resampling step and e_{tr-1} is the same quantity in the previous time step. With the adaptive selective resampling this ratio is almost always less than one, i.e. the resampling almost always reduces the pose error. Fig. 3 shows the maps generated using the FastSLAM modified and the classical FastSLAM algorithms. In the first case the real path (green) and the estimated path (black) are almost always overlapping and the robot closes the loop with a position error of only 0.0414 m; the classical FastSLAM closes the loop with a position error of 0.2040 m and there are pronounced gaps between the real path and the estimated path; both the algorithms generate a consistent map, but with the proposed modifications the quality of the map is so high that it can be magnified up to 3 cm of resolution without an observable significant error; on the other hand, with the classical FastSLAM algorithm, a map reconstruction error can be highlighted with a smaller magnification level (1 m is enough).

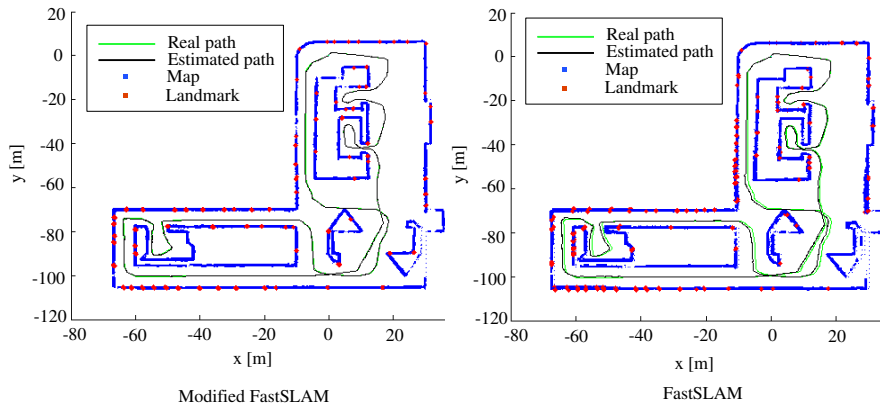


Fig. 3. Final 2-D map generated with FastSLAM and our modification of the FastSLAM

6 Conclusions

This work provides some modifications to the FastSLAM 2.0 algorithm. Unscented Transformations replace the EKF (avoiding linearizations) generating more accurate sets of particles S_t and improving the proposal distribution from which the set of particles is drawn. The paper has emphasized the importance of providing accurate *clouds* of particles for motion-only evolution. This allows to generate a more

consistent proposal and consequently a better set S_t . In addition an adaptive selective resampling has been introduced in the resampling step of the particle filter to significantly reduce w.r.t. the classical FastSLAM the risk of ill-timed resampling operations. Finally the set of EKF used in the classical FastSLAM approach for the landmark position estimate has been replaced by a set of UKF. The proposed modifications substantially improve the performance of the classical FastSLAM algorithm decreasing of about one order of magnitude the robot pose estimation error, as shown through an extensive campaign of simulations. The effectiveness of the adaptive selective resampling has been also illustrated in the paper.

References

1. J. Andrade-Cetto, A. Sanfeliu (2006) Environment Learning for Indoor Mobile Robots. Springer.
2. T.D. Barfoot (2005) Online Visual Motion Estimation using FastSLAM with SIFT Features. Proceedings of Intelligent Robots and Systems (IROS '05).
3. F. Dellaert, D. Fox, W. Burgard, and S. Thrun (1998) Monte Carlo localization for mobile robots. In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA).
4. M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba (2001) A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans. on Robotics and Automation, 17(3):229-241.
5. A. Doucet, N. de Freitas, K. Murphy, and S. Russell (2000) Rao-Blackwellised particle filtering for dynamic Bayesian networks. UAI '00 (Uncertainty in AI).
6. G. Grisetti, C. Stachniss, and W. Burgard, (2005) Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling, in IEEE International Conference on Robotics & Automation (ICRA).
7. S. J. Julier and J. K. Uhlmann (2004) Unscented filtering and non-linear estimation, Proceedings of the IEEE, vol. 92, no. 3, pp. 401-422.
8. S. J. Julier (2003) The spherical simplex Unscented Transformation, in Proc. of the American Control Conference (ACC), Denver.
9. J.S. Liu (1996) Metropolized independent sampling with comparisons to rejection sampling and importance sampling. Statistics and Computing, 6:113-119.
10. R. Martinez-Cantin, J.A. Castellanos (2005) Unscented SLAM for Large-Scale Outdoor Environments, in Intelligent Robots and Systems (IROS '05).
11. M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit (2002) FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. AAAI-02.
12. M. Montemerlo, S. Thrun D. Koller, B. Wegbreit (2003) FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, in Proc. of Int. Conf. on Artificial Intelligence (IJCAI).
13. P. Newman (2000) On the Structure and Solution of the Simultaneous Localisation and Map Building Problem. PhD thesis, Univ. of Sydney.
14. S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot (2004) FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association, Journal of Machine Learning Research.
15. R. van der Merwe, A. Doucet, N. de Freitas and E. Wan (2000) The Unscented Particle Filter. Advances in Neural Information Processing Systems (NIPS13). T.K. Leen, T.G. Dietterich and V. Tresp editors.