PEOPLE TRACKING WITH A MOBILE ROBOT: A COMPARISON OF KALMAN AND PARTICLE FILTERS

Nicola Bellotto and Huosheng Hu
Department of Computer Science, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, UK
email: {nbello,hhu}@essex.ac.uk

ABSTRACT

People tracking is an essential part for modern service robots. In this paper we compare three different Bayesian estimators to perform such task: Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Sampling Importance Resampling (SIR) Particle Filter. We give a brief explanation of each technique and describe the system implemented to perform people tracking with a mobile robot using sensor fusion. Finally, we report several experiments where the three filters are compared in terms of accuracy and robustness. In particular we show that, for this kind of applications, the UKF can perform as well as a particle filter but at a much lower computational cost.

KEY WORDS

People Tracking, Mobile Robot, Kalman Filter, Particle Filter, Multisensor Fusion.

1 Introduction

People tracking has become an essential skill modern service robots have to be provided with. Since they are supposed to operate in human environments and share the navigation space with normal people, robots have to be able to detect and track them in a fast and reliable way.

This is the case for service robots used in exhibitions and public places to entertain visitors and provide them with useful information. An example is the tour-guide robot of [1], which makes use of a laser-based tracking system to interact with people and create maps of the environment discarding human occlusions. Another important field of application is automatic or remote surveillance with security robots [2], which can be used to monitor wide areas of interest otherwise difficult to cover with fixed sensors. These robots have to able to detect and track people in restricted zones, for example to signal the presence of intruders to the security personnel.

People tracking is particularly challenging if we want to achieve the full autonomy of a robot without the use of external sensors or computers, in order to be independent from the current working place. This becomes even more difficult if the hardware resources are limited, which often happens, for example, in robots for surveillance applications. Hence the software design of the tracking system has to consider carefully the computational efficiency.

In this paper we compare different Bayesian estimators to perform people tracking using a fully autonomous mobile robot operating in a typical office environment. We consider three classic approaches: Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Sampling Importance Resampling (SIR) particle filter. All of them have been somehow applied for people tracking with robots.

In the work of [3] the EKF has been used to track a person with a mobile robot using stereo vision. Laser and visual data, instead, have been integrated in [4] using the UKF and sensor fusion techniques to perform people tracking. Other approaches are based instead on particle filters, sometimes using only one sensor [2], and other times combining more [5].

The choice of the best filter to use depends on several factors, among which the following important ones: linearity/non-linearity of the system, probability distribution of the uncertainty and, last but not least, computational efficiency. In the following sections we present a solution to the tracking problem that integrates legs and face detection, which are obtained respectively from the laser and the camera of a mobile robot. Analysing accuracy and robustness of the people tracking, we show that the UKF not only performs better than the EKF, but also that, for this task, it can be a valid alternative to particle filters, in particular when the hardware resources are limited and the computational efficiency is a key issue.

The paper is organized as follows. Section 2 introduces Bayes estimators and describes briefly EKF, UKF and SIR particle filter. An implementation of people tracking for a mobile robot is then illustrated in Section 3. Several experiments are described in Section 4 to compare the performance of the different estimators considered. Finally, conclusions and future work are reported in Section 5.

2 Recursive Bayesian Estimation

The most popular methods for dynamic state estimation belong to the family of recursive Bayesian estimators, which include Kalman filters [6, 7] and sequential Monte Carlo estimators [8], also known as particle filters.

In general, for a tracking application, the target state evolves according to the following discrete-time model:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \qquad (1)$$

where $\mathbf{x}_k$ is the state vector at the current time step k and $\mathbf{w}_{k-1}$ is white noise. The relative observations are normally described by another model as follows:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \tag{2}$$

where $\mathbf{z}_k$ is the observation vector and $\mathbf{v}_k$ is also white noise, mutually independent from $\mathbf{w}_{k-1}$. In general, f and h are non-linear functions.

Given a set of observations $\mathbf{Z}_k \triangleq \{\mathbf{z}_i, i = 1, \ldots, k\}$, the prior probability density $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$ can be expressed as follows:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})d\mathbf{x}_{k-1} \tag{3}$$

where the transitional density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is defined by (1). From (3), using the Bayes' rule, the posterior density can be calculated with the following equation:

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \tag{4}$$

where the denominator is a normalization factor:

$$p(\mathbf{z}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})d\mathbf{x}_k \tag{5}$$

and the likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ depends on (2).

Equations (3) and (4) are, respectively, the prediction and correction (or update) stages of a recursive Bayesian estimator. At each time step, the estimation usually consists in calculating the *Minimum Mean-Square Error* (MMSE) estimate $\hat{\mathbf{x}}_{k|k}^{\text{MMSE}} \triangleq \mathbb{E}\{\mathbf{x}_k|\mathbf{Z}_k\}$.

2.1 Extended Kalman Filter

Although originally not formulated as such, the Kalman filter belongs to the general class of Bayesian estimators and is known to be optimal for a restricted class of linear systems with Gaussian noises. However, good performances can be still obtained for a non-linear system, if this is sufficiently described by a linear approximation, using the EKF.

First of all, at each time step, the partial derivative elements of the Jacobians $\mathbf{F}_k$, $\mathbf{W}_k$ and $\mathbf{H}_k$ have to be calculated as follows [6]:

$$\mathbf{F}_k^{(i,j)} = \frac{\partial f_i}{\partial x_j}(\hat{\mathbf{x}}_{k-1}, 0) \quad \mathbf{W}_k^{(i,j)} = \frac{\partial f_i}{\partial w_j}(\hat{\mathbf{x}}_{k-1}, 0)$$
$$\mathbf{H}_k^{(l,j)} = \frac{\partial h_l}{\partial x_j}(\hat{\mathbf{x}}_k^-) \tag{6}$$

The prediction is done calculating the *a priori* estimate $\hat{\mathbf{x}}_k^-$ and the error covariance $\mathbf{P}_k^-$:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, 0) \tag{7}$$

$$\mathbf{P}_k^- = \mathbf{F}_k\mathbf{P}_{k-1}\mathbf{F}_k^T + \mathbf{W}_k\mathbf{Q}_{k-1}\mathbf{W}_k^T \tag{8}$$

where $\mathbf{Q}_{k-1}$ is the covariance matrix of the state noise.

Given also the covariance matrix $\mathbf{R}_k$ of the observation noise, the correction procedure then computes and uses the Kalman gain $\mathbf{K}_k$ to calculate the *a posteriori* estimate $\hat{\mathbf{x}}_k$ and the error covariance $\mathbf{P}_k$ as follows:

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{9}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k) \tag{10}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \tag{11}$$

where $\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k$ and the term $(\mathbf{z}_k - \hat{\mathbf{z}}_k)$, with $\hat{\mathbf{z}}_k = h(\hat{\mathbf{x}}_k^-)$, is the difference between real and predicted measurement, also called *innovation*.

2.2 Unscented Kalman Filter

In the UKF, the first-order linearization of the EKF is substituted with an Unscented Transformation (UT), which captures mean and covariance of the probability distributions with carefully chosen weighted points (*sigma points*).

Given the state estimate $\hat{\mathbf{x}}$, of size n, and the error covariance $\mathbf{P}$, the $2n + 1$ sigma points $\boldsymbol{\mathcal{X}}_i$ and associated weights W_i of the UT are calculated as follows [7]:

$$
\begin{aligned}
\boldsymbol{\mathcal{X}}_0 &= \hat{\mathbf{x}} & W_0 &= \beta/(n+\beta) \\
\boldsymbol{\mathcal{X}}_i &= \hat{\mathbf{x}} + \left(\sqrt{(n+\beta)\mathbf{P}}\right)_i & W_i &= [2(n+\beta)]^{-1} \\
\boldsymbol{\mathcal{X}}_{i+n} &= \hat{\mathbf{x}} - \left(\sqrt{(n+\beta)\mathbf{P}}\right)_i & W_{i+n} &= [2(n+\beta)]^{-1}
\end{aligned} \tag{12}
$$

where $i = 1, \ldots, n$ and β is a parameter for tuning the higher order moments of the approximation ($n + \beta = 3$ for Gaussian distributions). The term $\left(\sqrt{(n+\beta)\mathbf{P}}\right)_i$ is the ith column or row of the matrix square root of $\mathbf{P}$.

The estimation with the UKF can be done as follows. First of all, the state is augmented to include the process noise, then the relative sigma points are generated, from the previous estimate $\hat{\mathbf{x}}_{k-1}$, using (12). The *a priori* mean $\hat{\mathbf{x}}_k^-$ and covariance $\mathbf{P}_k^-$ are predicted with the UT:

$$\boldsymbol{\mathcal{X}}_{i\ k}^- = f(\boldsymbol{\mathcal{X}}_{ik-1}) \quad \text{for } i = 0, \ldots, 2n \tag{13}$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2n} W_i \boldsymbol{\mathcal{X}}_{i\ k}^- \tag{14}$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2n} W_i \left[\boldsymbol{\mathcal{X}}_{i\ k}^- - \hat{\mathbf{x}}_k^-\right]\left[\boldsymbol{\mathcal{X}}_{i\ k}^- - \hat{\mathbf{x}}_k^-\right]^T \tag{15}$$

Using the observation model and the new sigma points in (13), the expected measurement is also calculated:

$$\boldsymbol{\mathcal{Z}}_{ik} = h(\boldsymbol{\mathcal{X}}_{i\ k}^-) \quad \text{for } i = 0, \ldots, 2n \tag{16}$$

$$\hat{\mathbf{z}}_k = \sum_{i=0}^{2n} W_i \boldsymbol{\mathcal{Z}}_{ik} \tag{17}$$

The innovation covariance $\mathbf{S}_k$ and the cross-correlation $\mathbf{C}_k$ are then computed as follows:

$$\mathbf{S}_k = \mathbf{R}_k + \sum_{i=0}^{2n} W_i \left[\boldsymbol{\mathcal{Z}}_{ik} - \hat{\mathbf{z}}_k\right]\left[\boldsymbol{\mathcal{Z}}_{ik} - \hat{\mathbf{z}}_k\right]^T \tag{18}$$

$$\mathbf{C}_k = \sum_{i=0}^{2n} W_i \left[\boldsymbol{\mathcal{X}}_{i\ k}^- - \hat{\mathbf{x}}_k^- \right] \left[\boldsymbol{\mathcal{Z}}_{ik} - \hat{\mathbf{z}}_k \right]^T \qquad (19)$$

The *a posteriori* estimate $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$ are finally given by the same (10) and (11) of the EKF using the gain calculated as follows:

$$\mathbf{K}_k = \mathbf{C}_k \mathbf{S}_k^{-1} \qquad (20)$$

2.3 SIR Particle Filter

Particle filters are practical implementations of recursive Bayesian estimators using Monte Carlo simulations [8]. The major advantage of such filters is that they can be applied both to linear and non-linear systems with any probability distribution.

In a general particle filter, the posterior of the state (4) can be approximated by the following weighted sum:

$$p(\mathbf{x}_k|\mathbf{Z}_k) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \qquad (21)$$

where the samples $\mathbf{x}_k^i$ are drawn from the *importance density* $q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$ and the weights are calculated as follows:

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)} \qquad (22)$$

For $N \to \infty$, the approximation (21) tends to the true posterior $p(\mathbf{x}_k|\mathbf{Z}_k)$.

The most popular implementation of particle filter is the SIR (or Boostrap) filter [8]. The chosen importance density, in this case, is the transitional prior:

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) \qquad (23)$$

which depends on the state model. The weights are simply given by the current measurement likelihood:

$$w_k^i \propto p(\mathbf{z}_k|\mathbf{x}_k^i) \qquad (24)$$

At the end of each iteration, the SIR algorithm performs a resample step that eliminates the particles with very small weights and then generates new ones, equally weighted, from the remaining samples.

The prediction of the SIR filter consists in generating new particles, from the previous ones, using (1) and samples drawn from the pdf of the state noise. Then, as soon as a new measurement is available, the update is performed calculating the weights (24), where the likelihood depends on the observation model (2), and deriving the approximated posterior (21). The particles are finally resampled for the next iteration. See [8] for a detailed explanation.

3 People Tracking

3.1 Human Detection

Most of the mobile robots performing people tracking are equipped with cameras or laser range sensors to detect humans [4, 5, 9]. We use both the devices and, as in other similar approaches, we detect legs with the laser and faces with the camera.

For the legs, instead of looking simply for local minima in the laser scans or applying motion detection techniques [9, 5], we use the detection algorithm described in [10]. This is based on the recognition of typical legs patterns, corresponding to three possible postures: legs apart, forward straddle and two legs together (or single leg). The method is quite robust, even in cluttered environments, and computationally inexpensive.

The face detection is based on the real-time solution of [11], which uses a cascade of classifiers to extract simple but critical visual features. This algorithm is color independent and quite robust to different light conditions. From the position inside the image, the direction of the face is calculated with simple geometric transformations [10].

3.2 State and Observation Models

For the prediction of the human motion, we adopt an extension of the constant velocity model, where the state includes position (x_k, y_k), height z_k, orientation ϕ_k and velocity v_k of the human target [4]:

$$\begin{cases} x_k = x_{k-1} + v_{k-1}\delta_k \cos\phi_{k-1} \\ y_k = y_{k-1} + v_{k-1}\delta_k \sin\phi_{k-1} \\ z_k = z_{k-1} + n_{k-1}^z \\ \phi_k = \phi_{k-1} + n_{k-1}^\phi \\ v_k = |v_{k-1}| + n_{k-1}^v \end{cases} \qquad (25)$$

with $\delta_k = t_k - t_{k-1}$. The noises n_{k-1}^z, n_{k-1}^ϕ and n_{k-1}^v are zero-mean Gaussians with $\sigma_z = 0.01$m, $\sigma_\phi = \frac{\pi}{6}$rad and $\sigma_v = 0.1$m/s.

The observation model of the legs detection takes into account the current position of the robot (x_k^R, y_k^R, ϕ_k^R), given by the odometry, and includes the bearing b_k and the distance r_k of the detected legs:

$$\begin{cases} b_k = \tan^{-1}\left(\dfrac{y_k - y_k^l}{x_k - x_k^l}\right) - \phi_k^R + n_k^b \\ r_k = \sqrt{\left(x_k - x_k^l\right)^2 + \left(y_k - y_k^l\right)^2} + n_k^r \end{cases} \qquad (26)$$

where (x_k^l, y_k^l) is the absolute position of the laser device (also depending on the odometry). The noises n_k^b and n_k^r are zero-mean Gaussian with $\sigma_b = \frac{\pi}{60}$rad and $\sigma_r = 0.1$m.

The model of the face detection takes into account the current pan ψ_k^c, tilt θ_k^c and absolute position (x_k^c, y_k^c, z_k^c) of the camera, in order to get bearing α_k and elevation β_k of the face, plus elevation γ_k of the chin:

$$\begin{cases} \alpha_k = \tan^{-1}\left(\dfrac{y_k - y_k^c}{x_k - x_k^c}\right) - \phi_k^R - \psi_k^c + n_k^\alpha \\ \beta_k = \tan^{-1}\left(\dfrac{z_k^c - z_k}{\sqrt{(x_k - x_k^c)^2 + (y_k - y_k^c)^2}}\right) - \theta_k^c + n_k^\beta \\ \gamma_k = \tan^{-1}\left(\dfrac{z_k^c - \mu z_k}{\sqrt{(x_k - x_k^c)^2 + (y_k - y_k^c)^2}}\right) - \theta_k^c + n_k^\gamma \end{cases} \qquad (27)$$

The costant μ in the third member of (27) is chosen so that the product μz_k corresponds approximately to the height of the the chin, which is also given by the face detection. Again, the noises n_k^α, n_k^β and n_k^γ are zero-mean Gaussians with $\sigma_\alpha = \sigma_\beta = \frac{\pi}{45}$ rad and $\sigma_\gamma = \frac{\pi}{30}$ rad.

3.3 Data Association and Tracks Creation

We adopt Nearest Neighbour (NN) data association to handle multiple targets [12]. This is a reasonable compromise between performance and computational cost, which gives good results in most of the cases where the set of entities to track is not too dense [4, 13].

False readings are excluded using a validation gate [12] and the following similarity measure is used for the creation of the association matrices [14]:

$$d = \frac{1}{(2\pi)^{\frac{m}{2}} \sqrt{|\mathbf{S}|}} \exp\left(-\frac{d_M}{2}\right) \quad (28)$$

where m is the size of the observation vector, $\mathbf{S}$ is the covariance matrix of the innovation and d_M is the Mahalanobis distance between real and predicted measurement.

The sensor readings, when discarded by the gating or the assignment procedure, are used to create new candidate tracks. We mainly consider sequences of adjacent legs detections, since faces do not provide distance information. However, the latter can contribute at the creation of new tracks if the current legs pattern is not reliable enough (e.g. single leg case). The procedure is better described in [10]. A track is eventually deleted from the database if not updated for more than a certain time or if the uncertainty of its position is too high.

4 Experimental Results

The performances of the three different filters used for people tracking have been tested with several experiments in a typical office environment using a Pioneer 2 robot (see Fig. 1). This is equipped with a SICK laser and a PTZ camera mounted on a special support. The embedded PC is a Pentium III 850MHz 128MB, running Linux OS. The software is written in C++ making use of the optimized Bayes++ library[1] for the estimation part. Images from the camera are provided at 10fps, while the frequency of the laser scans is 5Hz. The latter sets also the maximum update speed of the tracking software when running in real-time on the robot. This value can decrease depending also on the execution time of the particular estimator adopted.

The following paragraphs describes the comparison of EKF, UKF and SIR filter. The latter has been tested with 500 samples, which we found to be the minimum number of particles needed for a correct human tracking, and 1000 samples, closer to the quantity normally used by other researchers [5, 9]. The experiments show accuracy and robustness of the different methods.

[1]http://bayesclasses.sourceforge.net/Bayes++.html
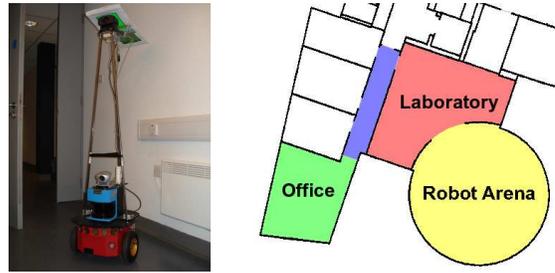
(a) Mobile robot. (b) Floor plan.

Figure 1. Robot and experiments location.

Table 1. Tracking Error for One Person

	EKF	UKF	SIR$_{(500)}$	SIR$_{(1000)}$
RMSE [m]	0.31	0.24	0.23	0.23
Mean [m]	0.24	0.20	0.19	0.19
SD [m]	0.20	0.13	0.13	0.13
Max [m]	1.49	0.97	1.13	1.10

4.1 Tracking Accuracy

To measure the error of the estimators, we utilized the absolute tracking system of our robot arena, which makes use of a camera mounted on the ceiling and calibrated to get the ground truth position of the robot and the people. All the data, including those from the robot's sensors, have been recorded to perform an off-line comparison of the filters. Fig. 2 shows an example of people being tracked from both the ceiling camera and the robot.

In the first experiment one person was walking randomly around the robot for about 60s, while this was moving approximately at 0.4m/s following a square path. The accuracy of the tracking is shown in Table 1, which reports RMSE, mean, standard deviation and maximum error using different estimators. As can be seen, the performances of the two SIR filters, which are of course better than the EKF, are almost identical, despite the higher number of particles. But what is more important is the fact that, basically, the UKF performs as well as the particle filters for this task.

Similar results have been obtained also in some other experiments with three persons being tracked by the robot when stationary or moving. One of these is illustrated in Fig. 3 with the paths of the robot and the people walking around it. The relative tracking errors are reported in Table 2 and show again that the accuracy of the UKF, when applied to people tracking, is similar to the SIR filter.

Please note that, while the EKF and UKF were very fast, the SIR filters could not run in real-time on the robot because, with just 500 particles and three people, the estimation time was close to the period of a laser scan (200ms). Since other tasks had to be executed (e.g. image processing), the SIR could not process the sensor information as fast as it should and was not be able to track properly.
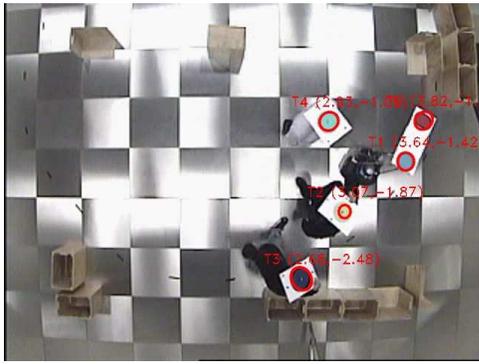
(a) Ceiling camera view. Each target has a color marker (one more for the robot to get its orientation).
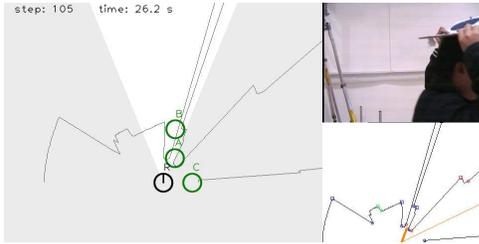
(b) Case of occlusion, as seen by the robot. Target A is passing in front of target B while the robot R is turning.

Figure 2. People tracking from ceiling camera and robot.

4.2 Tracking Robustness

Besides accuracy, another important characteristic of a tracking system is its robustness. To evaluate this, we considered the following parameters: total number of created tracks, sum of all their durations and total errors. The values are given in Table 3 for the previous case of three persons being tracked (the one-person case is omitted since always correct and basically identical for all the estimators).

As expected, the EKF performs worse than the other filters. The non-linearity of the system, indeed, was the cause of several track losses, from which derives the largest

Table 2. Tracking Error for Three Persons

		EKF	UKF	$SIR_{(500)}$	$SIR_{(1000)}$
	A	0.50	0.21	0.22	0.20
RMSE [m]	B	0.34	0.25	0.24	0.23
	C	0.29	0.26	0.25	0.25
	A	0.33	0.19	0.20	0.18
Mean [m]	B	0.26	0.21	0.20	0.19
	C	0.26	0.24	0.23	0.23
	A	0.38	0.09	0.08	0.09
SD [m]	B	0.22	0.15	0.12	0.12
	C	0.14	0.11	0.12	0.11
	A	1.98	0.70	0.59	0.65
Max [m]	B	1.04	0.87	0.62	0.57
	C	0.49	0.46	0.55	0.45
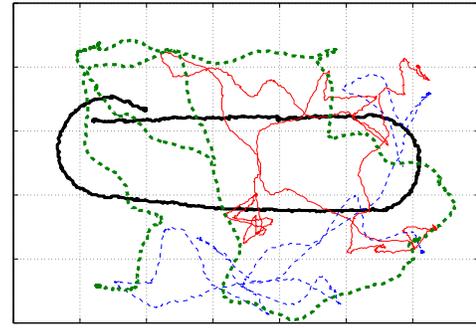
Figure 3. Paths of robot (thick line) and three persons: A (thin line), B (thin dashed line) and C (thick dashed line).

Table 3. Tracking Robustness for Three Persons

	EKF	UKF	$SIR_{(500)}$	$SIR_{(1000)}$
Total Tracks	11	7	7	7
Total Time [s]	60.23	65.03	64.72	64.58
Total Errors	2	0	0	0

number of generated tracks with the lowest tracking time.

Also, two of these tracks were erroneously associated to other targets in proximity of the original ones. With the UKF and the SIR filters, instead, the estimates were always correct, even in case of short occlusions. For example, Fig. 2 shows a situation where the robot was turning and a person, walking in front of it, was occluding another one. Even in this case, where the EKF failed, the UKF performed correctly, behaving exactly as the SIR filters.

Additional experiments have been carried out in more challenging situations, where the robot had to track one or more persons between different rooms. We report here a case were the robot followed a person starting from the office in Fig. 1(b), moving along a corridor at normal walking speed, crossing our laboratory and finally reaching the arena used for the previous experiments. The length of the path was approximately 20m, covered in about one minute, and included door passages and sharp turns.

Although almost always detected by the robot sensors, the EKF failed the tracking three times, as reported in Fig. 4. Instead, the UKF kept continuously the track until the final destination, behaving as good as the particle filters. This is shown for the same three instants in Fig. 5.

5 Conclusions and Future Work

This paper introduced a solution to people tracking with a mobile robot fusing the information provided by a camera and a laser device. We focused in particular on the comparison of three classic Bayesian estimators, EKF, UKF and SIR particle filter, using several experiments to test the performance of each one.

The results show that, for this specific task, an ap-
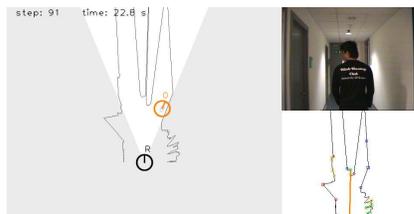
(a) First tracking error of the EKF in the corridor.
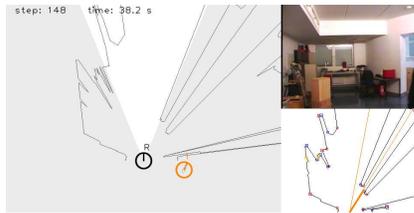
(b) Second error while entering the laboratory.
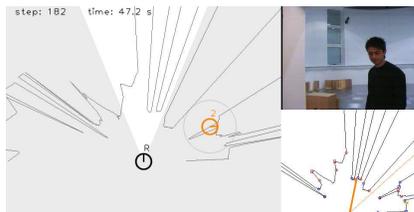
(c) Third error moving towards the arena.

Figure 4. Three error cases with a person followed and being tracked using the EKF.

(a) Following along the corridor.
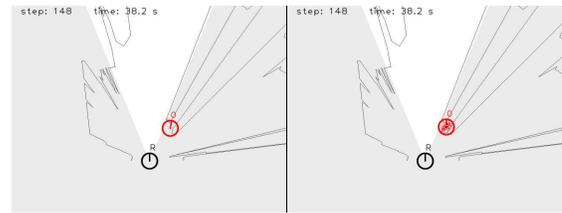
(b) Entering the laboratory.
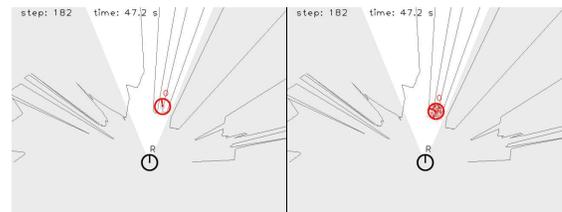
(c) Moving towards the arena.

Figure 5. Same situations as in Fig. 4, but this time with the UKF, on the left, and the $SIR_{(500)}$, on the right.

proach based on the UKF can work as well as a particle filter in terms of accuracy and robustness. The UKF could be therefore a better alternative in case the computational resources are limited, which often happens with autonomous mobile robots, but also in case the estimation needs to be fast enough to allow the execution of other tasks.

In the future, we would like to the extend the comparison including more recent and efficient particle filters. Finally, we plan to implement a real-time solution for simultaneous people tracking and recognition with a mobile robot, integrating also navigation and communication skills to perform human-robot interactions.

References

[1] W. Burgard, P. Trahanias, D. Hähnel, M. Moors, D. Schulz, H. Baltzakis, and Argyros A. TOURBOT and WebFAIR: Web-Operated Mobile Robots for Tele-Presence in Populated Exhibitions. In *Proc. IROS 2002 Workshop on Robots in Exhibitions*, 2002.

[2] A. Treptow, G. Cielniak, and T. Duckett. Active people recognition using thermal and grey images on a mobile security robot. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2103–2108, Canada, 2005.

[3] D. Beymer and K. Konolige. Tracking people from a mobile platform. In *IJCAI-2001 Workshop on Reasoning with Uncertainty in Robotics*, Seattle, WA, USA, 2001.

[4] N. Bellotto and H. Hu. Vision and laser data fusion for tracking people with a mobile robot. In *Proc. IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, pp. 7–12, China, 2006.

[5] P. Chakravarty and R. Jarvis. Panoramic vision and laser range finder fusion for multiple person tracking. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2949–2954, Beijing, China, 2006.

[6] G. Welch and G. Bishop. An introduction to the kalman filter. Tech. Rep. 95-041, University of North Carolina, 2004.

[7] S. J. Julier and J. K. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. In *Proc. SPIE AeroSense Symposium*, pp. 182–193, FL, USA, 1997.

[8] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Processing*, 50 (2):174–188, 2002.

[9] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. People tracking with mobile robots using sample-based joint probabilistic data association filters. *Int. J. of Robotics Research*, 22(2):99–116, 2003.

[10] N. Bellotto and H. Hu. Multisensor based human tracking for an indoor service robot. Submitted to IJRR, 2007.

[11] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. of Computer Vision*, 57(2):137–154, 2004.

[12] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. Y. Bar-Shalom, 1995.

[13] M. Bennewitz, G. Cielniak, and W. Burgard. Utilizing learned motion patterns to robustly track persons. In *Proc. IEEE Int. W. on VS-PETS*, pp. 102–109, France, 2003.

[14] D. L. Hall and J. Llinas, editors. *Handbook of multisensor data fusion*. CRC Press, 2001.