Autonomous Robots manuscript No.
(will be inserted by the editor)

Computationally Efficient Solutions for Tracking People with a Mobile Robot: an Experimental Evaluation of Bayesian Filters

Nicola Bellotto · Huosheng Hu

Received: date / Accepted: date

Abstract Modern service robots will soon become an essential part of modern society. As they have to move and act in human environments, it is essential for them to be provided with a fast and reliable tracking system that localizes people in the neighbourhood. It is therefore important to select the most appropriate filter to estimate the position of these persons. This paper presents three efficient implementations of multisensor-human tracking based on different Bayesian estimators: Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Sampling Importance Resampling (SIR) particle filter. The system implemented on a mobile robot is explained, introducing the methods used to detect and estimate the position of multiple people. Then, the solutions based on the three filters are discussed in detail. Several real experiments are conducted to evaluate their performance, which is compared in terms of accuracy, robustness and execution time of the estimation. The results show that a solution based on the UKF can perform as good as particle filters and can be often a better choice when computational efficiency is a key issue.

Keywords People Tracking · Mobile Robot · Kalman Filter · Particle Filter · Multisensor Fusion

1 Introduction

In the last decade, several mobile robots have been employed in exhibitions and public places to entertain visitors, interacting with them and providing useful information. The tour-guide robot in (Burgard et al. 2002), for example, has been working in a museum to accompany visitors and provide them with information about the different exhibits. The robot was equipped with a laser-based tracking to create maps of the environments discarding human occlusions, and to adapt its velocity to the visitors' motion. Another case was the interactive mobile robot described in (Bellotto and Hu 2005), which integrated laser and visual data to detect human legs and faces, moving towards visitors to interact with them by use of synthesized speech and a touch-screen interface.

Another field of application for people tracking is automatic or remote surveillance with mobile security robots, which can be used to monitor wide areas of interest otherwise difficult to cover with fixed sensors. These robots should be able to detect and track people in restricted zones, signaling, for examples, the presence of intruders to the security personnel. Such a task was accomplished by an internet-based mobile robot in (Liu et al. 2005), which used a PTZ camera to detect and recognize human faces. The security robot in (Treptow et al. 2005), instead, combined thermal vision, to detect and track people, with a normal camera, to track and recognize faces. Human tracking is also very important in the new research area of socially assistive robotics (Tapus et al. 2007) to maintain an appropriate spatial distance between people and robots, when these are engaged in social interactions.

To achieve full autonomy in mobile robotics, no external sensors or computers should be used, otherwise the system performance is completely dependent on the working environment. With these constraints, people tracking is particularly challenging and becomes even more difficult if the hardware resources are limited. Therefore, the computational efficiency of the tracking system has also to be carefully considered during software design.

N. Bellotto
School of Computer Science, University of Lincoln
Brayford Pool, Lincoln LN6 7TS, United Kingdom
Tel.: +44-(0)522-886080
E-mail: nbellotto@lincoln.ac.uk

H. Hu
E-mail: hhu@essex.ac.uk

The main contribution of this paper is an experimental comparison of different Bayesian estimators, which is important to select the most appropriate solution for tracking people with a fully autonomous mobile robot. Although some previous work already analyzed the performance of Kalman and particle filters (Merwe et al. 2000), results have been obtained only in simulation with synthetic models, or from batch estimations on limited set of data. These situations are significantly different from the case here considered, which deals instead with the difficult problem of real-time target tracking under computational constraints. The performance evaluation of these Bayesian estimators, considering also hardware and software limitations, is of fundamental importance for practical applications of modern service robots.

Three classic approaches are examined: Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF) and Sampling Importance Resampling (SIR) particle filter. While the first one is a well known technique developed long time ago (Kalman 1960), the last two solutions have been proposed more recently and extensively used only in the last decade (Julier and Uhlmann 1997; Gordon et al. 1993). The choice of these particular filters is due to the fact that all of them have been already applied, somehow, to people tracking with mobile robots. In this context, the performance of each individual technique has been already described, but not yet compared, in previous robotics literature.

The EKF has been implemented for tracking humans with mobile robots in the works of (Beymer and Konolige 2001) and (Bobruk and Austin 2004), using visual or laser data respectively. Both the devices have been used in (Bellotto and Hu 2009) applying sensor fusion techniques and UKF estimation to perform people tracking in typical office environments. Several other approaches have been proposed using particle filters with laser data and/or vision (Schulz et al. 2003a; Chakravarty and Jarvis 2006).

To evaluate and compare the effectiveness of each technique, a common framework has to be set up, on which quantitative and qualitative experiments can be conducted. Therefore, in the following sections, a general probabilistic approach for tracking people with a mobile robot is introduced. This solution integrates legs and face detections, obtained from robot's laser and camera respectively, which are fused using a sequential Bayesian filter. Since the comparison focuses on multi-target (people) tracking, the same data association algorithm is applied to all the filtering techniques under consideration.

The choice of the best estimator to use for human tracking 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. Whose familiar with the subject already know that Kalman filters are the most computational efficient, while particle filters are the most accurate. The challenge however lies on the design of meaningful experiments so that known facts can be proved on the base of solid quantitative data. In this paper, accuracy, robustness and execution time of the three Bayesian filters are analyzed, showing that a solution based on the UKF not only performs better than the EKF, but can also be a valid alternative to particle filters when used for tracking people with a mobile platform.

The remainder of the paper is organized as follows. Section 2 introduces the system designed to track people with a mobile robot. Sections 3, 4 and 5 describe respectively the implementation of the EKF, UKF and SIR particle filters. Several experiments are illustrated in Section 6 to compare the performance of the different solutions in real scenarios. Finally, conclusions and future work are discussed in Section 7.

2 People Tracking with a Mobile Robot

In general, tracking is a problem of estimating the position of a target from noisy sensor measurements. In the presence of multiple targets, which is the case for people tracking, each measurement has also to be assigned to the proper track. This section introduces the solutions adopted for human detection, tracking and data association with a mobile robot. The system used was a Pioneer platform, shown in Fig. 1, equipped with a SICK laser and a PTZ camera, which provided data respectively at 5Hz and 10fps. The on-board PC of the robot was a Pentium III 850MHz with 128MB of RAM, running Linux OS.

2.1 Human Detection

Two kind of sensors, cameras and laser range finders, are the most commonly used for tracking people with a mobile platform (Beymer and Konolige 2001; Schulz et al. 2003a; Chakravarty and Jarvis 2006). The robot employed in the current research makes use of both the sensors to recognize human legs and faces. The detection algorithms, and the advantage of combining laser and visual information, are described in detail in (Bellotto and Hu 2009).

The legs detection algorithm is able to recognize different legs postures on a 180° laser horizontal scan, with a resolution of 0.5°, returning their direction and distance. The algorithm starts with a smoothing process of the laser readings, and then detects all the radial edges on the directions of the laser beam. Groups of adjacent edges, possibly generated by human legs, are extracted using simple geometric relations and spatial constraints. The mid-points of these groups, corresponding to the 2D location of the legs, are finally computed.

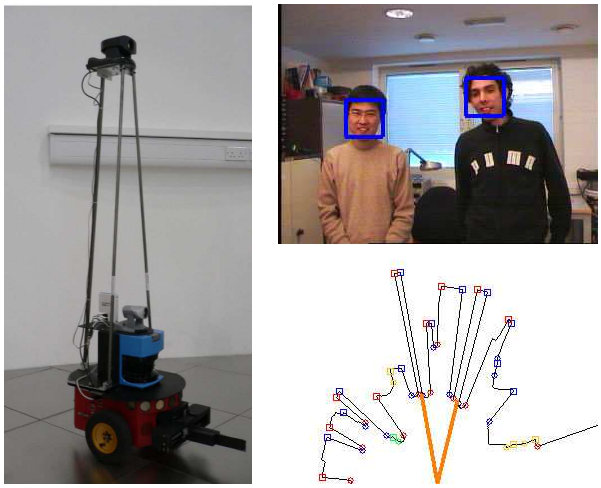


Fig. 1 Robot with laser and camera used for legs and face detection.

Since legs are detected in real-time from a single laser scan, the algorithm does not need to compensate for the dynamics of the robot, as other motion-based techniques do. The method is also quite robust to cluttered environments and showed to perform well compared to other laser-based detection techniques (Bellotto and Hu 2009). An example of detection is illustrated in Fig. 1, which shows a typical laser scan from the robot with two lines pointing to the human legs mid-points.

When in proximity of a person, vision improves human tracking thanks to face detection. This is based on a popular algorithm (Viola and Jones 2001) available on the OpenCV library (Bradski et al. 2005). The solution works in real-time on a single camera's frame, 320×240, and is color-independent, which makes it more robust to lighting variations.

The method is based on a cascade of (weak) classifiers using particular visual features. Each classifier is trained to detect faces, from sub-regions of the image, with a high hit rate. A sub-region can be rejected by the current classifier or passed to the following one. For a certain number of trained classifiers, the final false alarm will be therefore very low, yet keeping a total hit rate close to 100%. Using a pin-hole camera model, the direction of the face is finally calculated and used for human tracking, as discussed in Section 2.4.

2.2 Bayesian Estimation

The most popular methods for dynamic state estimation belong to the family of recursive Bayesian estimators, which include Kalman filters (Welch and Bishop 2004; Julier and Uhlmann 1997) and sequential Monte Carlo estimators (Arulampalam et al. 2002), also known as particle filters. These estimate the target position recursively, combining the expected state information with the current observations from the sensors.

In the discrete-time domain, for a general tracking application, the evolution of the target state can be described by the following general model:

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

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 generally described by another model with additive noise:

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

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

If $\mathbf{Z}_k = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$ is the set of observations up to time k, the prior probability density $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$ can be expressed as follows:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_{k-1}) &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}_{k-1})\; p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\; d\mathbf{x}_{k-1} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\; p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})\; d\mathbf{x}_{k-1} \end{aligned} \tag{3}$$

where the transitional $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is determined by the Markovian prediction model in (1). Then, applying Bayes' rule, the posterior density is given by the following equation:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_k) &= p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{Z}_{k-1}) \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}_{k-1})\; p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)\; p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \end{aligned} \tag{4}$$

Note that, in the numerator term, $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}_{k-1}) = p(\mathbf{z}_k|\mathbf{x}_k)$ because $\mathbf{z}_k$ is completely described by the observation model in (2), which depends only on the current state $\mathbf{x}_k$ and the noise $\mathbf{v}_k$. The denominator is just a normalization factor calculated as follows:

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

Equations (3) and (4) are called, respectively, prediction and update, or correction, of the recursive Bayesian estimation. The desired estimate is usually obtained, at the end of every predict-update iteration, by the minimum mean-square error value, i.e. the conditional mean $\hat{\mathbf{x}}_k \triangleq \mathbb{E}[\mathbf{x}_k|\mathbf{Z}_k]$.

2.3 Prediction

A common solution to approximate human motion, while walking at a normal speed, is the constant velocity model. The version here considered is an extension of the latter, already introduced in (Bellotto and Hu 2006), which includes a state vector formed by the position (x_k, y_k) and the height z_k of the human subject, plus the relative orientation ϕ_k and velocity v_k. The equations of the model are the following:

$$\begin{cases} x_k = x_{k-1} + v_{k-1}\delta_k \cos\phi_{k-1} \\ y_k = y_{k-1} + v_{k-1}\delta_k \sin\phi_{k-1} \\ z_k = z_{k-1} + n^z_{k-1} \\ \phi_k = \phi_{k-1} + n^\phi_{k-1} \\ v_k = |v_{k-1}| + n^v_{k-1} \end{cases} \tag{6}$$

where $\delta_k = t_k - t_{k-1}$ is the time interval, while n^z_{k-1}, n^ϕ_{k-1} and n^v_{k-1} are noises. These latter are assumed to be zero-mean Gaussians with $\sigma_z = 0.01$m, $\sigma_\phi = \frac{\pi}{6}$rad and $\sigma_v = 0.1$m/s respectively. The motion model in (6) is used for the prediction step of the Bayesian filter, as illustrated in Fig. 3

2.4 Sequential Update

The observation models described next take into account the 2D location and orientation of the robot given by the odometry. Its cumulative error is not an issue in the current application, since the objective of the system is to track humans relatively to the current robot's position. The odometry error between two consecutive estimations is also very small, and can be safely included in the noise of the observation models.

Given the location $(x^{\mathrm{R}}_k, y^{\mathrm{R}}_k)$ and heading ϕ^{R}_k of the robot, the absolute position $(x^{\mathrm{L}}_k, y^{\mathrm{L}}_k)$ and orientation ϕ^{L}_k of the laser are calculated as follows:

$$\begin{aligned} x^{\mathrm{L}}_k &= x^{\mathrm{R}}_k + L_x \cos\phi^{\mathrm{R}}_k \\ y^{\mathrm{L}}_k &= y^{\mathrm{R}}_k + L_x \sin\phi^{\mathrm{R}}_k \\ \phi^{\mathrm{L}}_k &= \phi^{\mathrm{R}}_k \end{aligned} \tag{7}$$

where the constant L_x is the horizontal distance of the laser from the robot's centre (L_y is zero). Using the quantities in (7), the observation model for the bearing b_k and the distance r_k of the detected legs can be written as follows:

$$\begin{cases} b_k = \tan^{-1}\left(\dfrac{y_k - y^{\mathrm{L}}_k}{x_k - x^{\mathrm{L}}_k}\right) - \phi^{\mathrm{L}}_k + n^b_k \\ r_k = \sqrt{(x_k - x^{\mathrm{L}}_k)^2 + (y_k - y^{\mathrm{L}}_k)^2} + n^r_k \end{cases} \tag{8}$$

where the noises n^b_k and n^r_k are zero-mean Gaussians with standard deviations $\sigma_b = \frac{\pi}{60}$rad and $\sigma_r = 0.1$m.

Similarly, the absolute position $(x^{\mathrm{C}}_k, y^{\mathrm{C}}_k, z^{\mathrm{C}}_k)$ and orientation $(\phi^{\mathrm{C}}_k, \theta^{\mathrm{C}}_k)$ of the camera take into account the horizontal distance C_x from the robot's centre (C_y is zero), the height

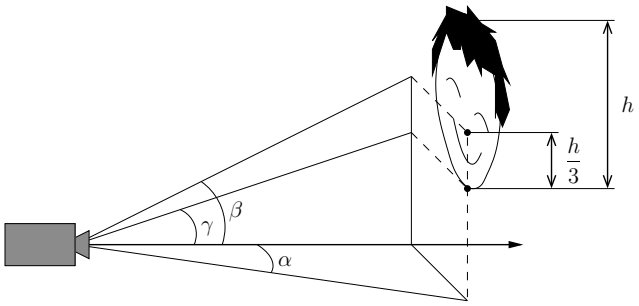


Fig. 2 Face observation angles, including chin, measured from the camera.

C_z, the pan C_ϕ and the tilt C_θ. Combining the odometry information, these can be calculated as follows:

$$\begin{aligned} x^{\mathrm{C}}_k &= x^{\mathrm{R}}_k + C_x \cos\phi^{\mathrm{R}}_k \\ y^{\mathrm{C}}_k &= y^{\mathrm{R}}_k + C_x \sin\phi^{\mathrm{R}}_k \\ z^{\mathrm{C}}_k &= C_z \\ \phi^{\mathrm{C}}_k &= \phi^{\mathrm{R}}_k + C_\phi \\ \theta^{\mathrm{C}}_k &= C_\theta \end{aligned} \tag{9}$$

The next observation model is relative to the bearing α_k and the elevation β_k of the face's centre, plus the elevation γ_k of its chin. The latter is relative to the size of the face and is useful to discriminate false positives or facilitate data association in case of multiple faces. The equations of the model are the following:

$$\begin{cases} \alpha_k = \tan^{-1}\left(\dfrac{y_k - y^{\mathrm{C}}_k}{x_k - x^{\mathrm{C}}_k}\right) - \phi^{\mathrm{C}}_k + n^\alpha_k \\ \beta_k = -\tan^{-1}\left[\dfrac{z_k - z^{\mathrm{C}}_k}{\sqrt{(x_k - x^{\mathrm{C}}_k)^2 + (y_k - y^{\mathrm{C}}_k)^2}}\right] - \theta^{\mathrm{C}} + n^\beta_k \\ \gamma_k = -\tan^{-1}\left[\dfrac{\mu\, z_k - z^{\mathrm{C}}_k}{\sqrt{(x_k - x^{\mathrm{C}}_k)^2 + (y_k - y^{\mathrm{C}}_k)^2}}\right] - \theta^{\mathrm{C}} + n^\gamma_k \end{cases} \tag{10}$$

The noises n^α_k, n^β_k and n^γ_k are zero-mean Gaussians with $\sigma_\alpha = \sigma_\beta = \frac{\pi}{45}$rad and $\sigma_\gamma = \frac{\pi}{30}$rad. Note that, in the third member of (10), the constant μ is chosen so that the product $\mu\, z_k$ corresponds to the height of the lower face's bound, i.e. approximately the chin. For the latter, the "canon of proportion" of the human figure, as described in (Vitruvius 1914), has been adopted. This considers the average height of a person being 8 times his head, and the distance from the chin to the nose is $1/3$ of the head's length h, as illustrated in Fig. 2. Since the face detection is centred on the nose, a value $\mu \simeq 0.955$ can be easily derived.

The independent measurements provided by legs and face detection are finally used for a sequential update of the estimation (Bar-Shalom and Li 1995). As shown by the diagram in Fig. 3, which illustrates a single iteration of the filter, legs measurements are the first to be considered, since more accurate, and then faces. If any of the two observations is missing, the estimate is updated only by one sensor.

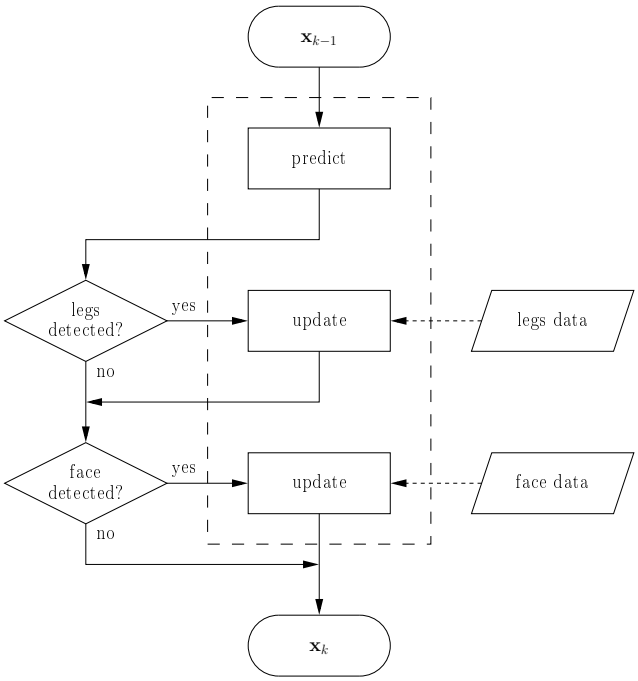


Fig. 3 Sensor data fusion with sequential estimation.

2.5 Data Association

In the current system, a gating procedure is first applied using a validation region for each predicted observation $\hat{\mathbf{z}}_i$ (Bar-Shalom and Li 1995), relative to the i^{th} target, so that a real measurement $\mathbf{z}_j$ is accepted only if it satisfies the following condition:

$$(\hat{\mathbf{z}}_i - \mathbf{z}_j)^T \mathbf{S}_{ij}^{-1} (\hat{\mathbf{z}}_i - \mathbf{z}_j) < \lambda^2 \tag{11}$$

where $\mathbf{S}_{ij}$ is the covariance matrix of the difference $\hat{\mathbf{z}}_i - \mathbf{z}_j$. The constant λ is chosen from tables of the chi-square distribution for a probability P_G of the correct measurements to fall within the validation region. This value depends on the size of the observation vector and is set to 3.03 for legs detections and 3.37 for faces.

Instead of solutions like JPDA and MHT, powerful but computationally expensive, an efficient algorithm based on nearest-neighbour data association is adopted (Bar-Shalom and Li 1995). This showed to be a good compromise between performances and computational cost in case the set of subjects to track is not too dense (Montemerlo et al. 2002; Bellotto and Hu 2006), in particular for autonomous robots with limited processing power. At every time step, two association matrices are created, one for the laser and another for the camera information. The elements of these matrices contain the following similarity measure (Uhlmann 2001):

$$d_{ij} = \frac{1}{\sqrt{(2\pi)^n |\mathbf{S}_{ij}|}} \exp\left[-\frac{1}{2}(\hat{\mathbf{z}}_i - \mathbf{z}_j)^T \mathbf{S}_{ij}^{-1} (\hat{\mathbf{z}}_i - \mathbf{z}_j)\right] \tag{12}$$

where n is the size of the observation vector, i.e. 2 for legs detection and 3 for faces.

2.6 Creating and Removing Tracks

New tracks are created from the sensor readings discarded during the validation gate procedure or the data association. Initially, a candidate track is generated by a sequence of measurements falling inside a certain region, calculated according to the maximum distance a person can cover at the maximum speed of 1.5m/s. The candidate is promoted to human track if there are at least 3 readings falling within this region, each one of which must be received not later than 0.5s from the previous one, otherwise the candidate is removed. Tracks are eventually deleted from the database if not updated for more than 2s or if the uncertainty of their 2D position is too big, i.e. the sum of the variances in x and y is greater than 2m^2.

Note that the procedure for tracks creation is independent from the particular Bayesian estimator used, therefore its parameters, equally set for EKF, UKF or SIR filter, do not influence the experimental comparison. The deletion criteria, instead, is based on time but also on the estimated covariance of the track, which might therefore be different depending on the filter used. In practice however, the uncertainty's threshold works only as a precaution, and tracks are usually removed because they exceed the time condition, which is the same for all the estimators.

3 EKF Implementation

The Kalman filter was initially proposed in (Kalman 1960) and, although originally not formulated as such, it has been later shown to belong to the more general class of Bayesian estimators (Barker et al. 1994). It was also proved to be optimal in case of linear systems with Gaussian noises, for which the posterior in (4) becomes the following:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_k) &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{P}_k) \\ &= |2\pi\mathbf{P}_k|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{P}_k^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k)\right] \end{aligned} \tag{13}$$

where $\hat{\mathbf{x}}_k$ and $\mathbf{P}_k$ are, respectively, the estimated mean and covariance of $\mathbf{x}_k$.

In case of non-linearities, the EKF provides an approximated solution, applying the same equations to linearized system models. This can give good results if the linearization is sufficiently accurate to describe the system, but fails badly if it is not.

Given the state vector $\mathbf{x}_k = [x_k, y_k, z_k, \phi_k, v_k]^T$ and the relative noise $\mathbf{w}_k = [0, 0, n_k^z, n_k^\phi, n_k^v]^T$, the components

of which have already been defined in Section 2.3, the linearized version of the prediction model in (6) has the form:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{14}$$

The Jacobian $\mathbf{F}_k$ is calculated as follows:

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & 0 & -v_{k-1}\Delta t_k \sin\phi_{k-1} & \Delta t_k \cos\phi_{k-1} \\ 0 & 1 & 0 & v_{k-1}\Delta t_k \cos\phi_{k-1} & \Delta t_k \sin\phi_{k-1} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \mathrm{sgn}\,(v_{k-1}) \end{bmatrix} \tag{15}$$

where $\Delta t_k = t_k - t_{k-1}$ is the time interval and $\mathrm{sgn}(v_{k-1})$ is the algebraic sign of v_{k-1}.

The prediction stage consists in calculating the *a-priori* estimate $\hat{\mathbf{x}}_k^-$ and the covariance matrix $\mathbf{P}_k^-$ of its error:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, 0) \tag{16}$$

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q} \tag{17}$$

where $\mathbf{Q}$ is the covariance of the (additive) process noise:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_v^2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 0 & \frac{\pi^2}{81} & 0 \\ 0 & 0 & 0 & 0 & 10^{-2} \end{bmatrix} \tag{18}$$

The observation models described in Section 2.4 are linearized as follows:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{19}$$

where $\mathbf{H}_k$ is the Jacobian of the laser or camera observation, with relative noise vectors $\mathbf{v}_k \equiv [n_k^b, n_k^r]^T$ or $\mathbf{v}_k \equiv [n_k^\alpha, n_k^\beta, n_k^\gamma]^T$. In the first case, given the observation vector $[b_k, r_k]^T$ and the quantities defined in (8), $\mathbf{H}_k$ is defined as follows:

$$\mathbf{H}_k \equiv \mathbf{H}_k^{\mathrm{L}} = \begin{bmatrix} -\dfrac{y_k - y_k^{\mathrm{L}}}{d_k^2} & \dfrac{x_k - x_k^{\mathrm{L}}}{d_k^2} & 0 & 0 & 0 \\ \dfrac{x_k - x_k^{\mathrm{L}}}{d_k} & \dfrac{y_k - y_k^{\mathrm{L}}}{d_k} & 0 & 0 & 0 \end{bmatrix} \tag{20}$$

$$\text{with } d_k^2 = (x_k - x_k^{\mathrm{L}})^2 + (y_k - y_k^{\mathrm{L}})^2$$

For the second one, given the vector $[\alpha_k, \beta_k, \gamma_k]^T$, the Jacobian matrix of the model in (10) is the following:

$$\mathbf{H}_k \equiv \mathbf{H}_k^{\mathrm{C}} = \begin{bmatrix} -\dfrac{y_k - y_k^{\mathrm{C}}}{d_k^2} & \dfrac{x_k - x_k^{\mathrm{C}}}{d_k^2} & 0 & 0\,0 \\ \dfrac{(x_k - x_k^{\mathrm{C}})\,(z_k - z_k^{\mathrm{C}})}{r_k^2\, d_k} & \dfrac{(y_k - y_k^{\mathrm{C}})\,(z_k - z_k^{\mathrm{C}})}{r_k^2\, d_k} & -\dfrac{d_k}{r_k^2} & 0\,0 \\ \dfrac{(x_k - x_k^{\mathrm{C}})\,(\mu\, z_k - z_k^{\mathrm{C}})}{l_k^2\, d_k} & \dfrac{(y_k - y_k^{\mathrm{C}})\,(\mu\, z_k - z_k^{\mathrm{C}})}{l_k^2\, d_k} & -\dfrac{d_k}{l_k^2} & 0\,0 \end{bmatrix} \tag{21}$$

$$\begin{aligned} \text{with } \; & d_k^2 = (x_k - x_k^{\mathrm{C}})^2 + (y_k - y_k^{\mathrm{C}})^2 \\ & r_k^2 = d_k^2 + (z_k - z_k^{\mathrm{C}})^2 \\ & l_k^2 = d_k^2 + (\mu\, z_k - z_k^{\mathrm{C}})^2 \end{aligned}$$

The update part includes the calculation of the following Kalman gain $\mathbf{K}_k$:

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R} \tag{22}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{23}$$

The quantity $\mathbf{S}_k$ in (22) is the innovation covariance and $\mathbf{R}$ is the covariance matrix of the observation noise $\mathbf{v}_k$. In case of laser readings, the latter is set as follows:

$$\mathbf{R} \equiv \mathbf{R}^{\mathrm{L}} = \begin{bmatrix} \sigma_b^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} = \begin{bmatrix} \frac{\pi^2}{3600} & 0 \\ 0 & 10^{-2} \end{bmatrix} \tag{24}$$

instead for the camera the following matrix is used:

$$\mathbf{R} \equiv \mathbf{R}^{\mathrm{C}} = \begin{bmatrix} \sigma_\alpha^2 & 0 & 0 \\ 0 & \sigma_\beta^2 & 0 \\ 0 & 0 & \sigma_\gamma^2 \end{bmatrix} = \begin{bmatrix} \frac{\pi^2}{2025} & 0 & 0 \\ 0 & \frac{\pi^2}{2025} & 0 \\ 0 & 0 & \frac{\pi^2}{900} \end{bmatrix} \tag{25}$$

Finally, the *a-posteriori* estimate $\hat{\mathbf{x}}_k$ and the relative error covariance $\mathbf{P}_k$ are computed as follows:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\,(\mathbf{z}_k - \hat{\mathbf{z}}_k) \tag{26}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \tag{27}$$

where the term $(\mathbf{z}_k - \hat{\mathbf{z}}_k)$, with $\hat{\mathbf{z}}_k = \mathbf{h}(\hat{\mathbf{x}}_k^-)$, is the difference between real and predicted measurements, also called *innovation*.

4 UKF Implementation

To overcome the problem of the linearization, which could introduce large errors and require the computation of big Jacobian matrices, the UKF makes use of another approximation, called the Unscented Transformation (UT). This is based on the idea that it is generally easier and more accurate to approximate probability distributions than non-linear functions. The UT captures mean and covariance of a probability distribution with carefully chosen weighted points, called *sigma points*. These differ from the points of particles filters in that they are not randomly sampled and do not have to lie in the interval $[0, 1]$.

From the state $\mathbf{x}$ of size n, and its error covariance $\mathbf{P}$, the $2n + 1$ sigma points $\boldsymbol{\mathcal{X}}_i$ and associated weights W_i of the UT are calculated using the following equations (Julier and Uhlmann 1997):

$$\begin{aligned} \boldsymbol{\mathcal{X}}_0 &= \mathbf{x} & W_0 &= \rho/(n+\rho) \\ \boldsymbol{\mathcal{X}}_i &= \mathbf{x} + \left[\sqrt{(n+\rho)\,\mathbf{P}}\right]_i & W_i &= [2\,(n+\rho)]^{-1} \\ \boldsymbol{\mathcal{X}}_{i+n} &= \mathbf{x} - \left[\sqrt{(n+\rho)\,\mathbf{P}}\right]_i & W_{i+n} &= [2\,(n+\rho)]^{-1} \end{aligned} \tag{28}$$

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

Mean and covariance of a generic non-linear transformation $\mathbf{y} = \mathbf{g}(\mathbf{x})$ are calculated using the sigma points as follows:

$$\boldsymbol{\mathcal{Y}}_i = \mathbf{g}\left(\boldsymbol{\mathcal{X}}_i\right) \tag{29}$$

$$\mathbf{y} = \sum_{i=0}^{2n} W_i \boldsymbol{\mathcal{Y}}_i \tag{30}$$

$$\mathbf{P}_{\mathbf{yy}} = \sum_{i=0}^{2n} W_i \left[\boldsymbol{\mathcal{Y}}_i - \mathbf{y}\right]\left[\boldsymbol{\mathcal{Y}}_i - \mathbf{y}\right]^T \tag{31}$$

These equations yield to a projected mean and covariance that are correct up to the second order, giving better results than the EKF's linearization, yet keeping the same computational complexity.

Given the state vector $\mathbf{x}_k = [x_k, y_k, z_k, \phi_k, v_k]^T$ of size $n = 5$, the estimation procedure of the UKF consists initially in an UT. This takes the last estimate $\hat{\mathbf{x}}_{k-1}$ and its relative covariance $\mathbf{P}_{k-1}$ to generate, using (28), the $2n + 1 = 11$ sigma points $\boldsymbol{\mathcal{X}}_{i_{k-1}}$. Note that, in this case, the tuning parameter assumes a negative value $\rho = 3 - n = -2$. In (Julier et al. 2000), it is shown that $\rho < 0$ can lead to a non-positive semidefinite matrix when the state covariance is calculated with (31). In order to solve this problem, the authors suggest to simply add a term $\left[\boldsymbol{\mathcal{Y}}_0 - \hat{\mathbf{y}}\right]\left[\boldsymbol{\mathcal{Y}}_0 - \hat{\mathbf{y}}\right]^T$ to the sum in (31).

Using the prediction model $\mathbf{f}(\mathbf{x}_{k-1})$ defined in (6), the a-priori estimate $\hat{\mathbf{x}}_k^-$ and covariance $\mathbf{P}_k^-$ are computed as follows:

$$\hat{\mathbf{x}}_{k-1} \xrightarrow{\text{UT}} \left\{\boldsymbol{\mathcal{X}}_{i_{k-1}}\right\}_{i=0}^{10} \tag{32}$$

$$\boldsymbol{\mathcal{X}}_{i_k}^- = \mathbf{f}\left(\boldsymbol{\mathcal{X}}_{i_{k-1}}\right) \text{ for } i = 0, \ldots, 10 \tag{33}$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{10} W_i \boldsymbol{\mathcal{X}}_{i_k}^- \tag{34}$$

$$\begin{aligned}\mathbf{P}_k^- \;=\; & \sum_{i=0}^{10} W_i \left[\boldsymbol{\mathcal{X}}_{i_k}^- - \hat{\mathbf{x}}_k^-\right]\left[\boldsymbol{\mathcal{X}}_{i_k}^- - \hat{\mathbf{x}}_k^-\right]^T + \\ & \left[\boldsymbol{\mathcal{X}}_{0_k}^- - \hat{\mathbf{x}}_k^-\right]\left[\boldsymbol{\mathcal{X}}_{0_k}^- - \hat{\mathbf{x}}_k^-\right]^T + \mathbf{Q}\end{aligned} \tag{35}$$

where $\mathbf{Q}$ is the covariance of the process noise defined in (18).

The expected observations for the legs and face detections are generated using the observation model $\mathbf{h}(\mathbf{x}_k)$, defined respectively in (8) and (10), applied to the sigma points in (33) as follows:

$$\boldsymbol{\mathcal{Z}}_{i_k} = \mathbf{h}\left(\boldsymbol{\mathcal{X}}_{i_k}^-\right) \text{ for } i = 0, \ldots, 10 \tag{36}$$

$$\hat{\mathbf{z}}_k = \sum_{i=0}^{10} W_i \boldsymbol{\mathcal{Z}}_{i_k} \tag{37}$$

$$\begin{aligned}\mathbf{S}_k \;=\; & \sum_{i=0}^{10} W_i \left[\boldsymbol{\mathcal{Z}}_{i_k} - \hat{\mathbf{z}}_k\right]\left[\boldsymbol{\mathcal{Z}}_{i_k} - \hat{\mathbf{z}}_k\right]^T + \\ & \left[\boldsymbol{\mathcal{Z}}_{0_k} - \hat{\mathbf{z}}_k\right]\left[\boldsymbol{\mathcal{Z}}_{0_k} - \hat{\mathbf{z}}_k\right]^T + \mathbf{R}\end{aligned} \tag{38}$$

where $\hat{\mathbf{z}}_k$ is the predicted observation, $\mathbf{S}_k$ is the innovation covariance and $\mathbf{R}$ is the covariance of the observation noise, defined in (24) for the laser and in (25) for the camera.

The cross-correlation $\mathbf{C}_k$ and the gain $\mathbf{K}_k$ are computed using the following formulas:

$$\mathbf{C}_k = \sum_{i=0}^{10} W_i \left[\boldsymbol{\mathcal{X}}_{i_k}^- - \hat{\mathbf{x}}_k^-\right]\left[\boldsymbol{\mathcal{Z}}_{i_k} - \hat{\mathbf{z}}_k\right]^T \tag{39}$$

$$\mathbf{K}_k = \mathbf{C}_k \mathbf{S}_k^{-1} \tag{40}$$

Finally, the a-posteriori estimate $\hat{\mathbf{x}}_k$ and relative covariance $\mathbf{P}_k$ are determined applying the same equations (26) and (27) previously used for the EKF.

5 SIR Implementation

Particle filters are recursive Bayesian estimators that make use of Monte Carlo methods to approximate and transform probability distributions (Doucet et al. 2001; Arulampalam et al. 2002; Ristic et al. 2004). The major advantages of such filters are their independence from the non-linearities of a system and capability to approximate any kind of probability distribution, including multimodal cases. The drawback is that a large number of particles is normally required for a good estimation, with a consequent negative effect on the computational cost.

In particle filters, the posterior of the state, introduced in (4), is approximated by the weighted sum of N samples $\mathbf{x}_k^i$:

$$p(\mathbf{x}_k|\mathbf{Z}_k) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{41}$$

where $\delta(\cdot)$ is the Dirac delta measure. The samples $\mathbf{x}_k^i$ are drawn from a known *importance density* $q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$, and their weights are calculated recursively as follows:

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)} \tag{42}$$

It can be proved that for $N \to \infty$ the approximation in (41) tends to the true posterior $p(\mathbf{x}_k|\mathbf{Z}_k)$.

There are many different implementations of particle filters, however the SIR algorithm is probably the most popular, due to its simplicity. This estimator, originally proposed

in (Gordon et al. 1993) with the name of "bootstrap" filter, makes use of the transitional prior as importance density:

$$q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i) \tag{43}$$

Like for the previous ones, the SIR estimation has an iterative predict-update sequence. The prediction part generates new particles, from the previous ones, using (1) and samples drawn from the probability distribution of the state noise. In the current implementation, the number of samples used was 1000, similar to other existing solutions (Chakravarty and Jarvis 2006; Schulz et al. 2003a), and also 500, which reduces the computational burden but is still sufficient to track humans correctly. The prior distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is a Gaussian $\mathcal{N}[\mathbf{x}_k; \mathbf{f}(\mathbf{x}_{k-1}), \mathbf{Q}]$, where $\mathbf{f}(\mathbf{x}_{k-1})$ is the prediction model defined in (6) and $\mathbf{Q}$ is the same covariance matrix in (18).

Then, as soon as a new measurement is available, the update is performed calculating the new weights of the samples. The choice of the importance density in (43) simplifies the calculus of the weights, which are given by the following formula:

$$w_k^i \propto w_{k-1}^i \, p(\mathbf{z}_k|\mathbf{x}_k^i) \tag{44}$$

The likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ is a Gaussian $\mathcal{N}[\mathbf{z}_k; \mathbf{h}(\mathbf{x}_k), \mathbf{R}]$ that depends on the observation models $\mathbf{h}(\mathbf{x}_k)$ defined in (8) and (10), for laser and camera respectively, and on the relative noise covariance $\mathbf{R}$ illustrated in (24) and (25).

Weighted samples are finally used to calculate an approximated posterior with (41). At the end of each iteration, the SIR algorithm performs also a resample step that eliminates all the particles with very small weights and, from the remaining ones, generates new samples equally weighted. A detailed explanation of SIR and other particle filters is given in (Arulampalam et al. 2002; Ristic et al. 2004).

6 Experimental Results

The effectiveness of the tracking system has been tested, with several experiments in a real environment, comparing the three solutions based on EKF, UKF and SIR filters. To achieve maximum performances, the code has been written in C/C++ making use of highly optimized libraries for image processing[1] and estimation[2]. When running in real-time on the robot, the maximum update frequency of the program was approximately 4Hz, but it could decrease in case particle filters were used. The test scenario was the indoor environment illustrated in Fig. 4, which includes several offices, connected by a corridor to a laboratory and a robot arena. Data have been collected tracking 7 different subjects who were moving in this environment. The results have been compared in terms of accuracy, robustness and computational efficiency.

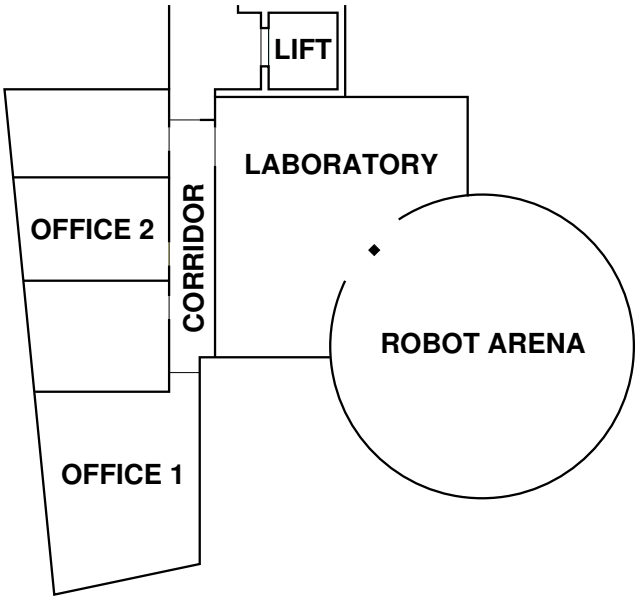


Fig. 4 Floor plan of the environment used for the experiments.

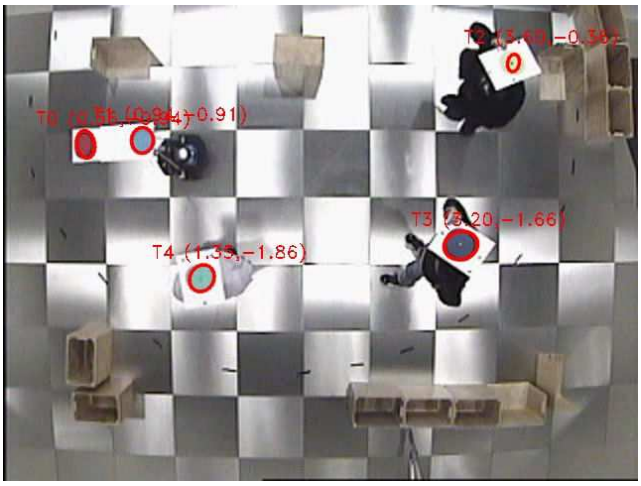
6.1 Tracking Accuracy

The accuracy of the estimations has been determined using the ground-truth position measured in the robot arena. This is equipped with a marker-based tracking system using a camera mounted on the ceiling, calibrated to provide the ground-truth position of the robot and the people around it. A bird-eye view from the ceiling camera and the relative observation from the robot are shown in Fig. 5.

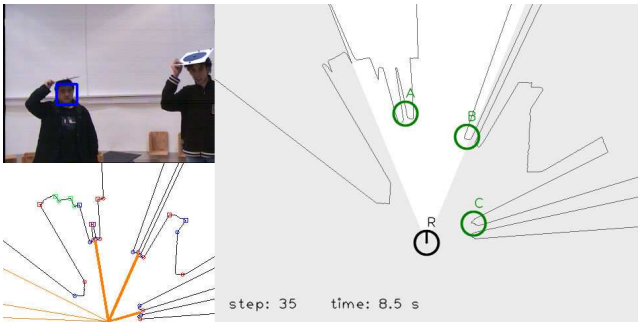
Experimental data have been recorded during four similar trials, for a total length of approximately 5 minutes (1200 time steps). These covered various cases in which a single person or multiple people were tracked, either with the robot static or in motion. An example is represented in Fig. 6, which illustrates the trajectory of the robot, moving at 0.4m/s, and the random paths of three people wandering around it. The data collected from the robot and from the global tracking system have been used for an off-line comparison of the accuracy, where the tracking error was given by the Euclidean distance between the estimated human position, $(\hat{x}_k, \hat{y}_k)$, and the relative ground-truth, (x_k^*, y_k^*). The latter was obtained tracking the robot with the ceiling camera, together with the human targets. Then, at every time step, their absolute position was transformed to the robot's frame of reference.

The results of the experiments are summarized in Table 1, which reports the root mean square (RMS) of the 2D position error e_k, calculated over all the M tracking steps,

[1] Intel IPP – http://developer.intel.com/software/products/ipp
[2] Bayes++ – http://bayesclasses.sourceforge.net



(a) Bird-eye view from the ceiling camera of the robot arena. Each target has a color marker (one more for the robot to get its orientation).



(b) Same situation as observed by the robot. Face and legs detection are shown on the left. The robot R and the (true) position of the humans, A,B, and C, are shown on the right.

Fig. 5 Example of people tracked in the robot arena.

and the relative mean $\bar{e}$, the standard deviation (SD) and the maximum value. The position error is defined as follows:

$$e_k = \sqrt{(\hat{x}_k - x_k^*)^2 + (\hat{y}_k - y_k^*)^2} \tag{45}$$

The number M is the sum of the duration, in time steps, of all the human tracks created during these experiments. The RMS, the mean and the SD were calculated as follows:

$$\text{RMS} = \sqrt{\frac{\sum_{k=1}^{M} e_k^2}{M}} \tag{46}$$

$$\bar{e} = \frac{\sum_{k=1}^{M} e_k}{M} \tag{47}$$

$$\text{SD} = \sqrt{\frac{1}{M-1}\sum_{k=1}^{M}(e_k - \bar{e})^2} \tag{48}$$

The results in Table 1 show that the performances of the two SIR filters were almost identical, despite the different number of particles used. Note also that the UKF's tracking

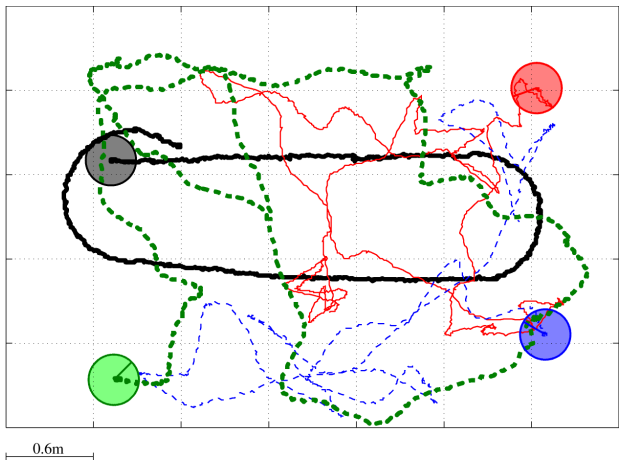


Fig. 6 Paths of robot (thick line) and three persons: A (thin line), B (thin dashed line) and C (thick dashed line).

Table 1 Tracking error

	EKF	UKF	$\text{SIR}_{(500)}$	$\text{SIR}_{(1000)}$
RMS [m]	0.439	0.317	0.285	0.280
Mean [m]	0.325	0.261	0.248	0.244
SD [m]	0.296	0.180	0.141	0.138
Max [m]	2.084	1.680	1.291	1.267

accuracy, besides being better than the EKF, was also very close to that one obtained with particle filters.

This is also confirmed by the graphs of the cumulative distribution function (CDF) for the RMS and the SD of the error, shown respectively in Fig. 7 and Fig. 8. Using the solution proposed in Colegrove et al. (2003), which defines a practical method to evaluate the performance of tracking systems from real data, RMS and SD values are adopted as comparison metrics. These are calculated for the whole length M_t of each human track t created during the experiments. The relative RMS_t and SD_t are reported in the abscissa of the graphs, each one corresponding to an increment $1/M_t$ of the probability in the ordinate. Since the lower the metric value the better the performance, the CDFs in Fig. 7 and Fig. 8 show that the tracking system based on UKF is better than the EKF's one, and is comparable to the SIR solutions.

6.2 Tracking Robustness

When comparing different tracking solutions, another essential factor to be considered is robustness. To evaluate this, two important parameters are considered here: the number of tracking errors and the total amount of tracks generated by the different systems. In addition to the previous data recorded in the robot arena, several other experiments have

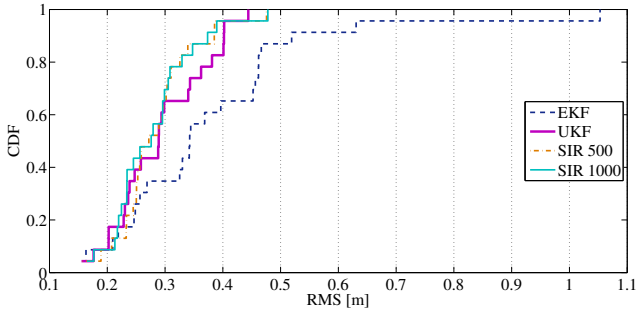


Fig. 7 Cumulative distribution function of the root mean square error.

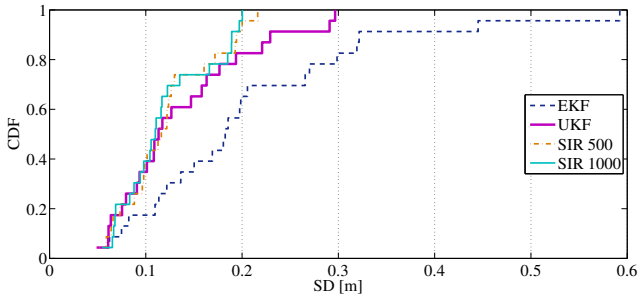


Fig. 8 Cumulative distribution function of the error standard deviation.

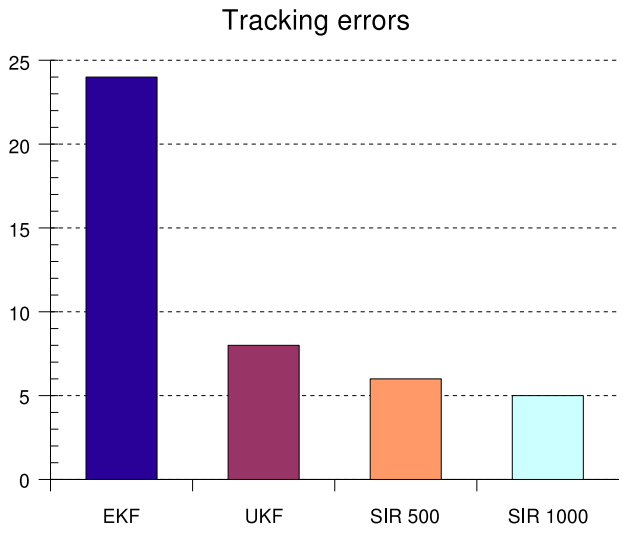


Fig. 11 Number of tracking errors with different filters.

been carried out tracking people who were moving between the rooms in Fig. 4. Totally, more than 10 minutes of data have been collected, and all the generated tracks have been manually labeled.

The number of tracking errors was evaluated considering only the 2D position for sake of simplicity. Each one of the following situations was counted as an error: a) the track deviates from the correct trajectory of the human target and is eventually deleted by the system; b) the track "jumps" to a static object, adjacent to the path of the person, due to a false positive (gating error); c) the track switches to another person close to the original one (data association error). All these cases are strictly related to the estimate of the filter and to the distribution of its uncertainty.

Although this work does not include an exhaustive evaluation of the tracking performance under varying sensing conditions (false positives, occlusions, etc.), intuitively these will be better handled by the UKF and the SIR particle filter, rather than the EKF, due to their ability to better model the propagated probability functions. This is shown, for example, in Fig. 9 and Fig. 10, where a couple of EKF's errors occurred while the robot was following some persons between different rooms. The correct path of the UKF track (identical to the SIR case) and the wrong one generated by the EKF are shown on the left of the figures, together with the robot's trajectory. A moment of the wrong tracking with the EKF is shown in the middle, while the correct estimation of the UKF is illustrated on the right. In Fig. 9, the tracking error in the office was caused by the curvilinear trajectory of the human and the simultaneous motion of the robot, as shown also in Video 1. In Fig. 10 and relative Video 2, the EKF failed between the laboratory and the arena as a consequence of a false positive on the legs detection, generated by a column. These situations were correctly handled instead by the UKF and the SIR tracking systems.

The chart in Fig. 11, showing the total amount of tracking errors, illustrates clearly that the results obtained with the UKF and the particle filters were much better than the EKF-based tracking. The non-linearity of the system, indeed, made the EKF fail in several occasions, in particular when both the robot and the person being tracked were moving. The performance of the UKF was generally similar to the SIR tracking in terms of the number of errors, but differed on the type. Despite occasional errors due to some false positives, the major accuracy of particle filters in representing the probability distribution of the estimate seemed to be an advantage for data association. However, as will be shown in Section 6.3, a solution based on particle filters was not feasible for real-time tracking with the current robot platform.

The previous results were also confirmed by the total number of tracks generated for each system implementation, as reported in Fig. 12. Indeed, the more robust and stable is the estimation, the less likely is the tracking to fail, and consequently the smaller is the number of tracks generated by the system. Although the whole tracking length was about the same for all the solutions (approximately 2750 estimation steps), the chart shows that the number of tracks was higher using the EKF. Instead, even in this case, the values relative to UKF and SIR particle filters were very close.

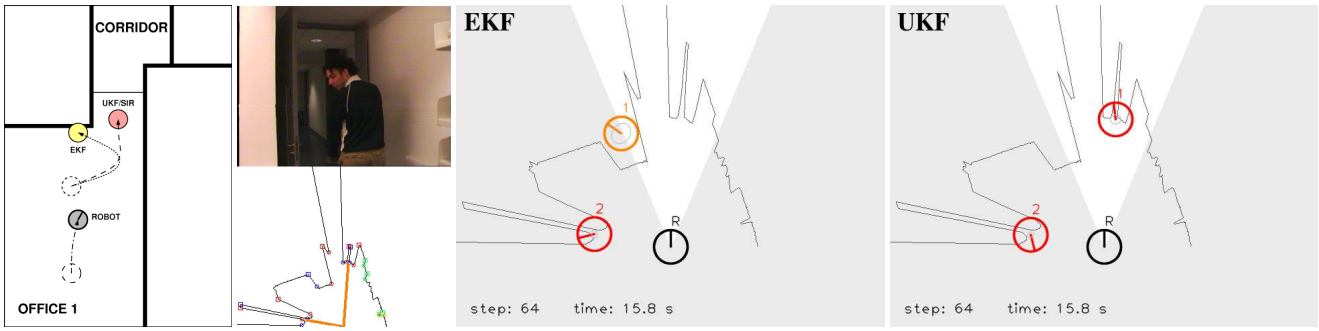


Fig. 9 Human tracking in Office 1.

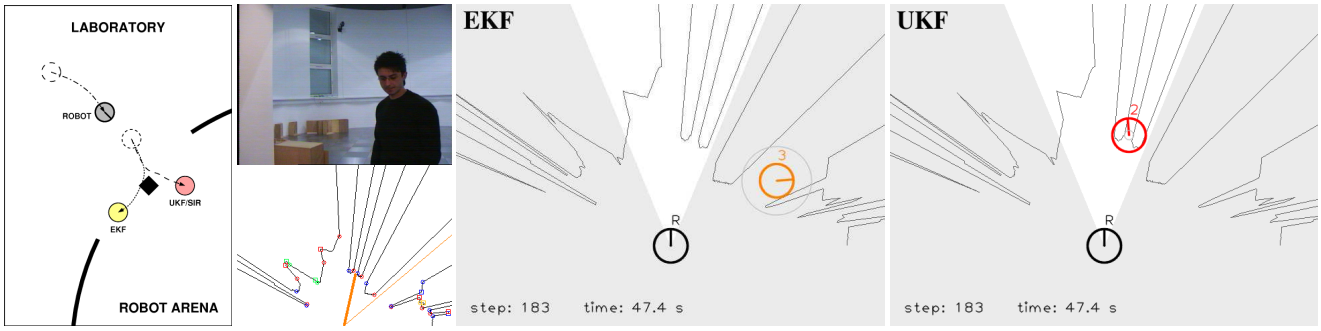


Fig. 10 Human tracking in the laboratory.

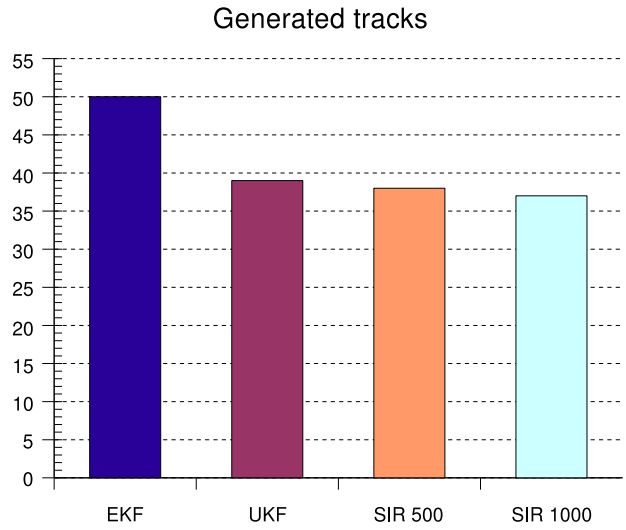


Fig. 12 Total amount of tracks generated during the experiments.

6.3 Computational Efficiency

It is known that, in general, particle filters are computationally much more demanding than Kalman filters, and that the time needed for the estimation increases with the number of samples used. For many applications, this does not represent a problem, because the number of actual estimations is limited (e.g. single target tracking) or simply because the hardware is powerful enough. It might pose a serious constraint, however, in case of frequent estimations and limited computing resources, e.g. for the system currently studied.

In this experiment, the execution time needed by each filter to perform an estimation (i.e. single iteration of the process in Fig. 3) was compared while tracking one or more persons. For simplicity, only legs detections were used to update the track estimates. The detections have been simulated with static laser data hard-coded in the software, generating from one to four pairs of legs observable at the same time. This permitted to have the same inputs constantly available to the tracking system for all the estimators under comparison.

The graph in Fig. 13 shows the average times needed for an update iteration run on the Pioneer robot, which includes only the prediction and one filter update, i.e. the first two blocks in the diagram of Fig. 3. The time spent for legs detection and additional routines (tracks handling, data association, logging, etc.) has not been counted. The results have been obtained averaging the total estimation time on 100 consecutive time steps, tracking up to 4 target simultaneously. The estimation processing used approximately 70% of the CPU and, as expected, had different time durations, depending on the filter adopted for tracking. The graph shows that the time increased almost linearly with the number of tracked persons and, for the SIR filter, with the number of particles.

It is important to note that, while the EKF and UKF solutions were very fast, the ones based on particle filters were much slower and, in some case, their tracking performance

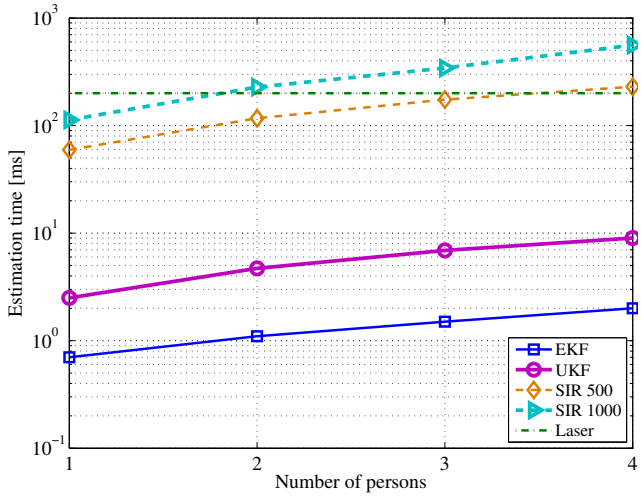


Fig. 13 Estimation time, in logarithmic scale, as function of the number of persons being tracked. The laser scans period (200ms) is shown for reference.

was drastically limited. For example, during the tracking of three or more people, the execution time of the SIR with 500 particles was close to 200ms (i.e. the period of a laser scan). Since normally there are other tasks to be executed in addition to the estimation, the SIR tracking system cannot process the sensor information as fast as it should and, very often, it is not able to work properly. Indeed, in this experiment, the measurement of the execution time was possible only because the targets were static, otherwise the SIR-based tracking would have failed because of the low update frequency (i.e. it could not work in real-time). Same considerations can be done for the SIR with 1000 particles, or considering a larger number of people. Although other particle filters computationally more efficient (Schulz et al. 2003b; Kwok et al. 2004) should be considered in future comparisons, the UKF remains generally a faster solution.

7 Conclusions and Future Work

This paper presented an experimental comparison of people tracking systems based on three different Bayesian estimators, namely EKF, UKF and SIR particle filter. These solution makes use of probabilistic sensor fusion techniques to integrate laser and visual data. Their implementation on a mobile robot have been described in detail. With several experiments in real situations, the systems have been compared in terms of accuracy, robustness and computational efficiency.

On the specific task of real-time people tracking with mobile robots, the results showed that a UKF solution could perform as good as particle filters. Furthermore, analyzing the estimation time, the UKF proved to be a better choice for the current application, in particular when hardware resources are limited. An approach based on this filter could be generally preferred for autonomous robots with low processing power, for which the computational efficiency is a key issue.

In the future, it would be interesting to extend this comparison to include more recent and efficient particle filters, possibly using different mobile platforms as well. Their performance could also be evaluated using different data association algorithms to see how these influence people tracking. The results of this research are also important for the authors' implementation of an interactive robot performing simultaneous people tracking and recognition, for which an accurate and robust real-time estimation is a fundamental requirement.

References

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

Bar-Shalom, Y. and Li, X. R. (1995). *Multitarget-Multisensor Tracking: Principles and Techniques*. Y. Bar-Shalom.

Barker, A. L., Brown, D. E., and Martin, W. N. (1994). Bayesian estimation and the kalman filter. Technical Report IPC-TR-94-002, Institute of Parallel Computing, School of Engineering and Applied Science, University of Virginia.

Bellotto, N. and Hu, H. (2005). Multisensor integration for human-robot interaction. *The IEEE Journal of Intelligent Cybernetic Systems*, 1.

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

Bellotto, N. and Hu, H. (2009). Multisensor-based human detection and tracking for mobile service robots. *IEEE Trans. on Systems, Man, and Cybernetics – Part B*, 39(1):167–181.

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

Bobruk, J. and Austin, D. (2004). Laser motion detection and hypothesis tracking from a mobile platform. In *Proc. of the 2004 Australian Conference on Robotics & Automation*, Canberra, Australia.

Bradski, G., Kaehler, A., and Pisarevsky, V. (2005). Learning-based computer vision with Intel's open source computer vision library. *Intel Technology Journal*, 09(02):119–130.

Burgard, W., Trahanias, P., Hähnel, D., Moors, M., Schulz, D., Baltzakis, H., and A., A. (2002). TOURBOT

and WebFAIR: Web-Operated Mobile Robots for Tele-Presence in Populated Exhibitions. In *Proc. of the IROS 2002 Workshop on Robots in Exhibitions*.

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

Colegrove, S., Cheung, B., and Davey, S. (2003). Tracking system performance assessment. In *Proc. of the 6th Int. Conf. on Information Fusion*, pages 926–933, Cairns, Australia.

Doucet, A., de Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo Methods in Practice*. Springer.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. of Radar and Signal Processing*, 140(2):107–113.

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

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. on Automatic Control*, 45(3):477–482.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Trans of the ASME - Journal of Basic Eng.*, 82:35–45.

Kwok, C., Fox, D., and Meilă, M. (2004). Real-time particle filters. *Proc. of the IEEE*, 92(3):469–484.

Liu, J. N. K., Wang, M., and Feng, B. (2005). iBotGuard: an internet-based intelligent robot security system using invariant face recognition against intruder. *IEEE Trans. on Systems, Man, and Cybernetics (Part C)*, 35(1):97–105.

Merwe, R. V. D., Doucet, A., Freitas, N. D., and Wan, E. (2000). The unscented particle filter. CUED/F-INFENG TR 380, Cambridge University Engineering Department.

Montemerlo, M., Whittaker, W., and Thrun, S. (2002). Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 695–701, Washington DC, USA.

Ristic, B., Arulampalam, S., and Gordon, N. (2004). *Beyond the Kalman filter: particle filters for tracking applications*. Artech House.

Schulz, D., Burgard, W., Fox, D., and Cremers, A. B. (2003a). People Tracking with Mobile Robots Using Sample-based Joint Probabilistic Data Association Filters. *Int. Journal of Robotics Research*, 22(2):99–116.

Schulz, D., Fox, D., and Hightower, J. (2003b). People Tracking with Anonymous and ID-Sensors Using Rao-Blackwellised Particle Filters. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 921–926, Acapulco, Mexico.

Tapus, A., Mataric, M. J., and Scasselati, B. (2007). Socially assistive robotics. *IEEE Robotics and Automation Magazine*, 14(1):35–42.

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

Uhlmann, J. K. (2001). Introductions to the algorithmics of data association in multiple-target tracking. In Hall, D. L. and Llinas, J., editors, *Handbook of multisensor data fusion*. CRC Press.

Viola, P. and Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 511–518, Kauai, HI, USA.

Vitruvius (1914). *Ten Books on Architecture*. Project Gutenberg. English translation by M. H. Morgan.

Welch, G. and Bishop, G. (2004). An Introduction to the Kalman Filter. Technical Report 95-041, University of North Carolina.