Multisensor-Based Human Detection and Tracking for Mobile Service Robots

Nicola Bellotto, *Student Member, IEEE*, and Huosheng Hu, *Senior Member, IEEE*

***Abstract*—One of fundamental issues for service robots is human–robot interaction. In order to perform such a task and provide the desired services, these robots need to detect and track people in the surroundings. In this paper, we propose a solution for human tracking with a mobile robot that implements multisensor data fusion techniques. The system utilizes a new algorithm for laser-based leg detection using the onboard laser range finder (LRF). The approach is based on the recognition of typical leg patterns extracted from laser scans, which are shown to also be very discriminative in cluttered environments. These patterns can be used to localize both static and walking persons, even when the robot moves. Furthermore, faces are detected using the robot's camera, and the information is fused to the legs' position using a sequential implementation of unscented Kalman filter. The proposed solution is feasible for service robots with a similar device configuration and has been successfully implemented on two different mobile platforms. Several experiments illustrate the effectiveness of our approach, showing that robust human tracking can be performed within complex indoor environments.**

***Index Terms*—Leg detection, people tracking, sensor fusion, service robotics, unscented Kalman filter (UKF).**

I. Introduction

IN RECENT years, the social aspect of service robots has made it clear that these are not only expected to navigate within the environment they have been placed in but they also have to interact with people to provide useful services and show good communication skills. The study of the so-called human-centered robotics and human–robot interaction aims to achieve such tasks and are currently some of the most fascinating research fields in mobile robotics. In general, a service robot has to focus its attention on humans and be aware of their presence. It is necessary, therefore, to have a tracking system that returns the current position, with respect to the robot, of the adjacent persons. This is a very challenging task, as people's behaviors are often completely unpredictable. Researchers have been using different methods to deal with this problem, in many cases, with solutions subjected to strong limitations, such as tracking in rather simple situations with a static robot or using some additional distributed sensors in the environment.

Manuscript received January 12, 2008; revised May 1, 2008 and July 17, 2008. First published December 9, 2008; current version published January 15, 2009. This paper was recommended by Associate Editor J. Su.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This includes five video files showing results of the implemented system. This material is 36.2 MB in size.

The authors are with the Human Centred Robotics Group, Department of Computing and Electronic Systems, University of Essex, CO4 3SQ Colchester, U.K. (e-mail: n.bellotto@ieee.org; hhu@essex.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2008.2004050

Service robotics has become one of the research areas attracting major attention for over a decade. The necessity of fast and reliable systems for tracking people with mobile robots is evidenced in the literature by the growing number of real-world applications. Human tracking can help service robots plan and adapt their movements according to the motion of the adjacent people or follow an instructor across different areas of a building. For example, the tour-guide robot of Burgard *et al.* [1] adopts laser-based people tracking both for interacting with users and for mapping the environment, discarding human occlusions. Another field of application is automatic or remote surveillance with security robots, which can be used to monitor wide areas of interest that are otherwise difficult to cover with fixed sensors. An example is the system implemented by Liu *et al.* [2], where a mobile robot tracks possible intruders in a restricted area and signals their presence to the security personnel.

The most common devices used for people tracking are laser sensors and cameras. For instance, Lindström and Eklundh [3] propose a laser-based approach to track a walking person with a mobile robot. The system detects only moving objects, keeping track of them with a heuristic algorithm, and needs the robot to be static or move very slowly. Zajdel *et al.* [4] illustrate a vision-based tracking and identification system which makes use of a dynamic Bayesian network for handling multiple targets. Even in this case, targets can be detected only when moving. Moreover, the working range is limited by the camera's angle of view; hence, it is difficult to track more than two subjects at the same time. The robot described by Luo *et al.* [5] uses a tilting laser to extract body features, which are fused then with the face detected by a camera. The solution is useful for pursuing a person in front of the robot; however, the complexity of feature extraction limits its application to multiple people tracking. Other implementations also use a combination of face detection and laser-based leg detection [6], [7]. In all these cases, however, since they do not rely on any motion model, situations in which a person is temporarily occluded are difficult to handle. Schulz *et al.* [8] describe a robot equipped with two laser range sensors that can track several people, using a combination of particle filters and probabilistic data association. Many other recent approaches make also use of particle filters, sometimes combining visual and laser data [9] to track from a fixed position or using other kind of devices such as thermal cameras [10].

The solution presented in this paper adopts multisensor data fusion techniques for tracking people from a mobile robot using a laser scanner and a monocular camera. A new detection algorithm has been implemented to find human legs by using laser scans, which works either in large empty environments

Fig. 1. Pioneer robot with laser and camera for legs and face detection.

or small cluttered rooms. Different from other approaches, our system is also able to distinguish among different leg postures, improving the discrimination of false positives. Vision is then used for face detection, and robust human tracking is performed with a sequential unscented Kalman filter (UKF) fusing the two different sensor data. The solution is feasible for several applications of mobile service robots with a similar device configuration, although the computational efficiency makes it particularly indicated for robots with limited processing power. The system has been tested in cluttered indoor environments, where human detection is made difficult by the presence of furniture or by the small size of a room, and proved to be robust enough to track people constantly even while the robot moves at a normal walking speed.

This paper is organized as follows. Section II explains, in detail, the algorithm for leg detection and also introduces the face-detection module. The tracking system, including UKF, sensor fusion, and data association, is described in Section III. Then, Section IV presents several experiments and analyzes the results. Finally, conclusions and future work are illustrated in Section V.

II. Detection

The human-tracking algorithm adopts multisensor data fusion techniques to integrate the following two different sources of information: the first one is leg detection, based on the laser scans of a Sick LRF, and the other one is face detection, which uses a monocular camera. The devices and their location are shown in Fig. 1 for one of our mobile robots. Next, we describe, in detail, the principles underlying these two detection algorithms.

A. Leg Detection

In the literature, there are several systems using laser scans to detect human legs. However, most of them are simply based on the search of local minima [6], [11], which, in general, works well only for rather simple environments such as empty rooms

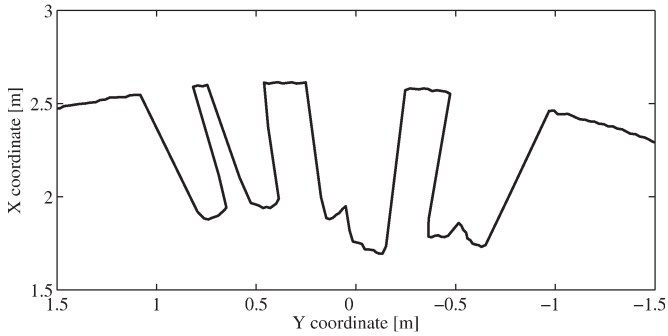

Fig. 2. Leg patterns extracted from a laser scan. Most of the time, (left) LA and (center) FS patterns can be easily distinguished from other objects; however, (right) SL patterns can be very ambiguous in some environments.

and corridors. In [12], it is reported that at least a single leg (SL) must always be well distinguishable, whereas attempts of using more general solutions showed to be not very robust in cluttered environments [13]. Many other approaches can only detect moving persons [3], [9], normally searching for differences between two or more scans. Aside from the problem of missing static humans, these methods are often unreliable for mobile robots due to the difficulty of compensating for the ego motion.

The algorithm for leg detection presented here extracts the necessary features from a single laser scan, independently then from the human or robot motion. However, in contrast with the local minima approach, we identify typical patterns (relative to particular leg postures) that, in most of the cases, are well distinguishable from the other objects in the environment. These patterns are shown with an example in Fig. 2 and correspond to the following three typical situations: two legs apart (LA), forward straddle (FS), and two legs together or SL. The first pattern is usually very common in case the person is standing in front of the robot. The second, however, is most likely to happen when the person is walking. The last pattern covers most of the remaining postures; often, however, it can also be generated by other objects in the environment.

As also shown in the schematic representation of Fig. 3, the algorithm is divided in three main parts:

1) data preprocessing;
2) detection of vertical edges;
3) extraction of leg patterns.

Suppose the angular step between two consecutive laser scans is constant and that the readings are stored in an array $\mathcal{S} = [r_1, \ldots, r_i, \ldots, r_M]$, where r_i is the range measured on the direction θ_i and M is the total number of readings. Initially, the laser data are preprocessed by applying a local minimization operator [3], in order to remove possible spikes due to reflections on sloped surfaces, and a local maximization operator, in order to discard thin objects such as table legs.

From the resulting array $\hat{\mathcal{S}}$ of preprocessed data, the recognition of the three different leg patterns can be done efficiently using the following method based on vertical edges features. If we represent $\hat{\mathcal{S}}$ on a Cartesian graph, with the angle (indexed by i) on the abscissa and the range on the ordinate, we can identify a sequence of vertical edges defined as follows. The

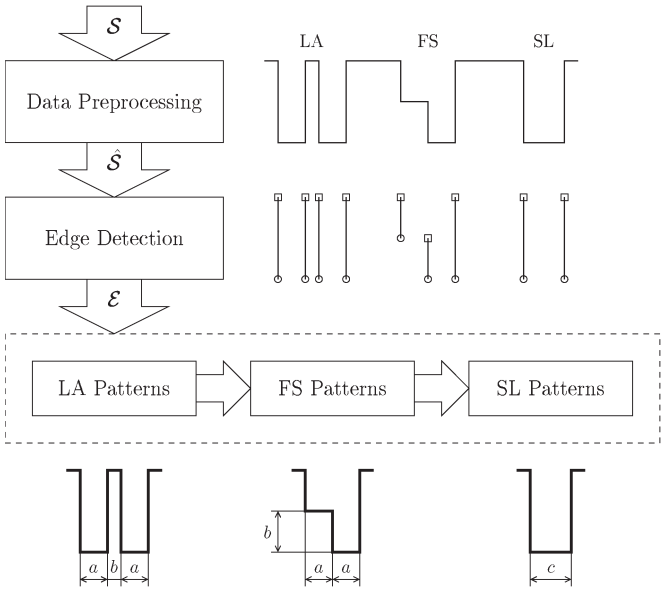

Fig. 3. Schematic representation of the leg-detection algorithm. The laser scan $\mathcal{S}$ is preprocessed to reduce measurement noise. From the resulting data $\hat{\mathcal{S}}$, the set of vertical edges $\mathcal{E}$ is then extracted. Finally, the algorithm detects, in order, the LA, FS, and SL patterns, based also on some constraints on the measures a, b, and c.

doublet $\{\hat{r}_i, \hat{r}_{i+1}\}$ can be considered an almost vertical edge if the distance $|\hat{r}_{i+1} - \hat{r}_i|$ is greater than a given threshold. Moreover, we can distinguish a left edge, when $\hat{r}_i > \hat{r}_{i+1}$, from a right edge, when $\hat{r}_i < \hat{r}_{i+1}$, and refer to them as L_i and R_i, respectively (hereafter, for simplicity, we omit the index i). The resulting vertical edges are initially queued into a list $\mathcal{E} = \{e_1, \ldots, e_n, \ldots\}$, where each element e_n can be either an L or R edge. If they are very close and almost aligned, adjacent edges of the same type are connected to form a longer one.

After that, from the updated list of connected edges, we extract all the subsets that might belong to one of the three leg patterns described before. In particular, according to some constraints and spatial relations between edges, including maximum distance between legs and limits on their size, the ordered sequences of left/right edges we look for are the following.

1) The LA pattern is a quadruplet $\{L, R, L, R\}$.
2) The FS pattern is a triplet $\{L, L, R\}$ or $\{L, R, R\}$.
3) The SL pattern is a doublet $\{L, R\}$.

Every edge is removed from $\mathcal{E}$ as soon as it contributes to form one of the aforementioned sequences. Therefore, all the LA patterns, which are normally the most reliable, are extracted first, while the SL patterns, which are the easiest to misinterpret, are left at the end.

With reference to Fig. 3, some dimensional constraints are fixed for the measures a, b, and c, which are, respectively, the leg's width, the maximum step length, and the width of two legs together. These are used by the algorithm's procedures to recognize LA, FS, and SL patterns, described in detail in Table I (to simplify the pseudocode and make it more readable, some checks on the indexes are omitted). Finally, the distance and direction of the detected legs are calculated from the midpoint of each pattern. Fig. 4 shows the leg detection of three persons in different postures.

TABLE I
ROUTINES FOR THE DETECTION OF LEG PATTERNS

{Extract the set $\mathcal{P}_{LA}$ of LA patterns from $\mathcal{E}$}
$\mathcal{P}_{LA} := \oslash$
$e_m :=$ first L edge of $\mathcal{E}$
while $e_{m+3} \neq$ null **do**
  if $(\{e_m, e_{m+1}\} = \{L, R\}) \wedge$ (leg) **then**
    $e_n := e_{m+2}$
    found $:= false$
    while (found $= false$) $\wedge$ ($e_{n+1} \neq$ null) **do**
      if $(\{e_m, e_{m+1}, e_n, e_{n+1}\} = \{L, R, L, R\}) \wedge$ (legs apart) **then**
        $\mathcal{P}_{LA} \leftarrow \{e_m, e_{m+1}, e_n, e_{n+1}\}$
        found $:= true$
        remove $\{e_m, e_{m+1}, e_n, e_{n+1}\}$ from $\mathcal{E}$
      else
        $e_n :=$ next L edge of $\mathcal{E}$
      end if
    end while
  end if
  $e_m :=$ next L of $\mathcal{E}$
end while

{Extract the set $\mathcal{P}_{FS}$ of FS patterns from $\mathcal{E}$}
$\mathcal{P}_{FS} := \oslash$
$e_m :=$ first L edge of $\mathcal{E}$
while $e_{m+2} \neq$ null **do**
  if $(\{e_m, e_{m+1}, e_{m+2}\} = \{L, L, R\}$ or $\{L, R, R\}) \wedge$ (forward straddle) **then**
    $\mathcal{P}_{FS} \leftarrow \{e_m, e_{m+1}, e_{m+2}\}$
    remove $\{e_m, e_{m+1}, e_{m+2}\}$ from $\mathcal{E}$
  else
    $e_m :=$ next L edge of $\mathcal{E}$
  end if
end while

{Extract the set $\mathcal{P}_{SL}$ of SL patterns from $\mathcal{E}$}
$\mathcal{P}_{SL} := \oslash$
$e_m :=$ first L edge of $\mathcal{E}$
while $e_{m+1} \neq$ null **do**
  if $(\{e_m, e_{m+1}\} = \{L, R\}) \wedge$ (two legs together or single leg) **then**
    $\mathcal{P}_{SL} \leftarrow \{e_m, e_{m+1}\}$
    remove $\{e_m, e_{m+1}\}$ from $\mathcal{E}$
  else
    $e_m :=$ next L edge of $\mathcal{E}$
  end if
end while

B. Face Detection

In order to improve the human-detection performance when in proximity of people, the robot is provided with a camera for face localization. One of the classic methods to accomplish this task in real time is based on the color segmentation of skin regions, as in the case reported by Fritsch *et al.* [6]. However, this kind of solution is prone to many errors due to light variations and shadows, as well as limitations in the detection of different skin tones.

The approach adopted in our system, instead, is based on the work of Viola and Jones [14] and is a further extension of [15]. The detection algorithm uses a set of simple but important visual features, the prototypes of which are shown in Fig. 5. The value of a feature is calculated subtracting the weighted sum of the pixels within the white rectangles from the weighted sum of the pixels within the black rectangles. The prototypes can be scaled independently in a horizontal or vertical direction, thus to generate an overcomplete set of features $\mathcal{F}$ far larger than the number of pixel in the considered subimage. For example, the

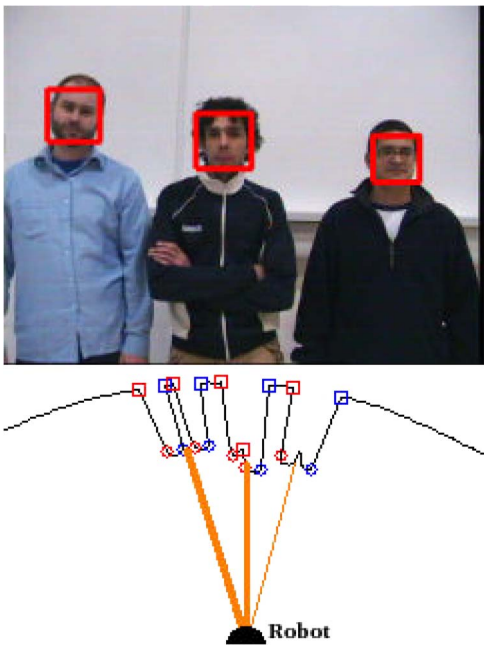

Fig. 4. Example of face and leg detections. From left to right, all the three leg patterns, LA, FS, and SL, are detected.

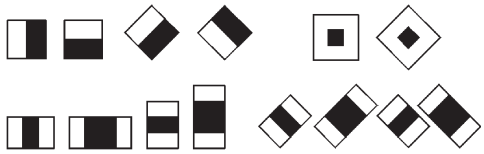

Fig. 5. Set of features used for face detection [15].

total number of features for a window size of 24 × 24 pixels is 117 941 [15].

Thus, the detection system consists of a cascade of weak classifiers, each one relative to a particular feature of $\mathcal{F}$. A modified version of the Adaboost algorithm [16] is used both to determine a small number of salient features and to train the relative classifiers. Every weak classifier is trained to detect faces, with a high hit rate, from subregions of the image. A pattern can be rejected by the current classifier or passed to the following one, as shown in Fig. 6. For a certain number of trained classifiers, the final false alarm will be very low yet keeping a total hit rate close to 100%. For example, if the number of classifiers is $N = 10$, with each one trained so that the hit rate is 99.8% and the false alarm is 50%, the resulting cascade will still have a high hit rate of $0.998^{10} \simeq 0.98$ but with a very small false alarm of $0.5^{10} \simeq 0.001$.

Aside from being very fast, an important characteristic of this face detection is that it is color independent and, therefore, not constrained by the skin tone of a person. As reported by Viola and Jones [14] and as also tested with our robot under different conditions, this algorithm is quite robust to varying illumination and to facial details such as beard or glasses. An example is shown in Fig. 4.

The position of the face on the image can be used to calculate its bearing and elevation with respect to the camera's location and orientation. For our purpose, a simple pinhole camera

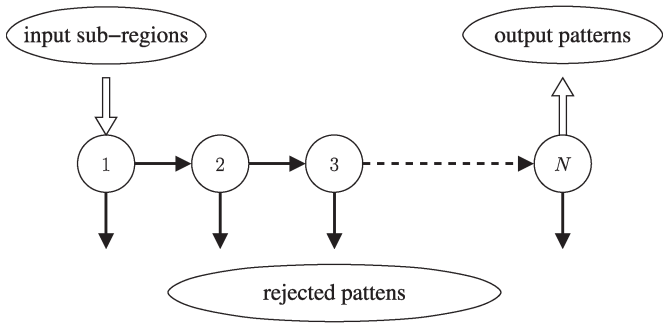

Fig. 6. Cascade of classifiers for face detection.

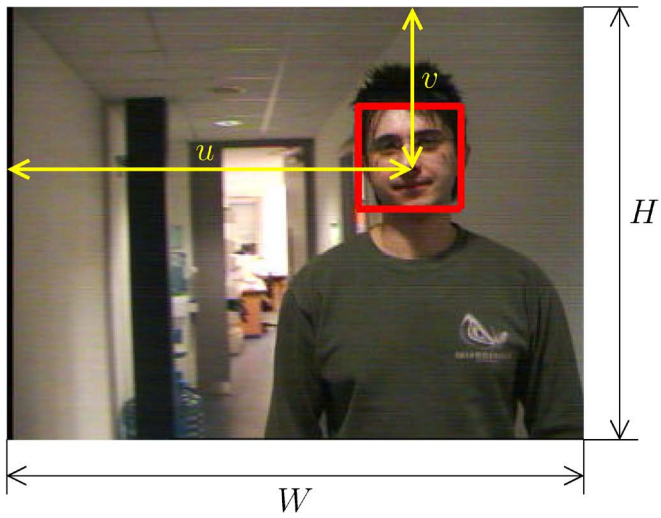

Fig. 7. Face-detection algorithm calculates bearing and elevation of the face center based on its position on the image plane and the focal length of the camera. (a) Face detection from camera image. The values u and v give the position, in pixels, of the face center. The coordinates of the bounding box around the face are also available. (b) Bearing α and elevation β of a point P (face center) captured by the camera with focal length f.

model can be adopted, and the distortion introduced by the camera lens can be ignored. We then derive the following transformations [17]:

$$\alpha = \tan^{-1}\left(\frac{W/2 - u}{f}\right) \qquad \beta = \tan^{-1}\left(\frac{v - H/2}{f}\right) \quad (1)$$

where (u, v) is the face's center on an image $W \times H$ and f is the focal length in pixel units, as also shown in Fig. 7.

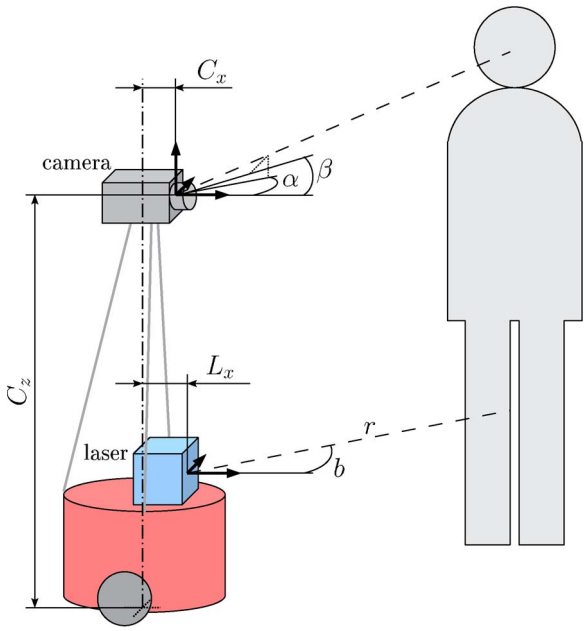

Fig. 8. Location of the robot sensors and detection measurements.

Furthermore, considering that the bounding box of the detected face is also available, we can get an additional measure from the elevation of the lower (or upper) bound. This can be useful, for example, for distinguishing different faces, considering that their size on the image varies with the distance from the camera.

III. Tracking

Tracking a walking person is a very challenging task. There are many factors which may influence the human trajectories, such as the environment, the number of people, or the interactions among them. The unpredictability of such behaviors have been modeled by some researchers as Brownian motion [18], although a constant velocity model seems to be a better choice in order to deal with clutters [8], [19]. This section describes the methodology used in our tracking system.

A. State and Observation Models

We adopt the state model as in [19], which is an extension of the constant velocity model. The equations are as follows:

$$\begin{cases} x_k = x_{k-1} + v_{k-1}\Delta t_k \cos\phi_{k-1} \\ y_k = y_{k-1} + v_{k-1}\Delta t_k \sin\phi_{k-1} \\ z_k = z_{k-1} + n^z_{k-1} \\ \phi_k = \phi_{k-1} + n^\phi_{k-1} \\ v_k = |v_{k-1}| + n^v_{k-1} \end{cases} \tag{2}$$

where $\Delta t_k = t_k - t_{k-1}$ and the state is given by the position (x_k, y_k), the height z_k, the orientation ϕ_k, and the velocity v_k of the human target. The noises n^z_{k-1}, n^ϕ_{k-1}, and n^v_{k-1} are zero-mean Gaussians with standard deviations $\sigma_z = 0.01$ m, $\sigma_\phi = \pi/9$ rad, and $\sigma_v = 0.1$ m/s.

With reference to Fig. 8, the absolute 2-D position $(x^{\mathrm{L}}_k, y^{\mathrm{L}}_k)$ and orientation ϕ^{L}_k of the laser depend on the current location $(x^{\mathrm{R}}_k, y^{\mathrm{R}}_k)$ and heading ϕ^{R}_k of the robot, as given by the odometry, and are calculated as follows:

$$\begin{aligned} x^{\mathrm{L}}_k &= x^{\mathrm{R}}_k + L_x \cos\phi^{\mathrm{R}}_k \\ y^{\mathrm{L}}_k &= y^{\mathrm{R}}_k + L_x \sin\phi^{\mathrm{R}}_k \\ \phi^{\mathrm{L}}_k &= \phi^{\mathrm{R}}_k \end{aligned} \tag{3}$$

where the constant L_x is the horizontal distance of the laser from the robot's center (L_y is null). The observation model for the laser, which includes the bearing b_k and distance r_k of the detected legs, can therefore be written as follows:

$$\begin{cases} b_k = \tan^{-1}\left(\frac{y_k - y^{\mathrm{L}}_k}{x_k - x^{\mathrm{L}}_k}\right) - \phi^{\mathrm{L}}_k + n^b_k \\ r_k = \sqrt{\left(x_k - x^{\mathrm{L}}_k\right)^2 + \left(y_k - y^{\mathrm{L}}_k\right)^2} + n^r_k \end{cases} \tag{4}$$

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

Aside from the odometry, the absolute 3-D 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 center (C_y is null), the height C_z, the pan C_ϕ, and the tilt C_θ

$$\begin{aligned} x^{\mathrm{C}}_k &= x^{\mathrm{R}}_k + C_x \cos\phi^{\mathrm{R}}_k \\ y^{\mathrm{C}}_k &= y^{\mathrm{R}}_k + C_x \sin\phi^{\mathrm{R}}_k \\ z^{\mathrm{C}}_k &= C_z \\ \phi^{\mathrm{C}}_k &= \phi^{\mathrm{R}}_k + C_\phi \\ \theta^{\mathrm{C}}_k &= C_\theta. \end{aligned} \tag{5}$$

The following observation model for the face detection calculates the bearing α_k and elevation β_k of the face's center, as well as the elevation γ_k of the chin:

$$\begin{cases} \alpha_k = \tan^{-1}\left(\frac{y_k - y^{\mathrm{C}}_k}{x_k - x^{\mathrm{C}}_k}\right) - \phi^{\mathrm{C}}_k + n^\alpha_k \\ \beta_k = -\tan^{-1}\left[\frac{z_k - z^{\mathrm{C}}_k}{\sqrt{\left(x_k - x^{\mathrm{C}}_k\right)^2 + \left(y_k - y^{\mathrm{C}}_k\right)^2}}\right] - \theta^{\mathrm{C}} + n^\beta_k \\ \gamma_k = -\tan^{-1}\left[\frac{\mu z_k - z^{\mathrm{C}}_k}{\sqrt{\left(x_k - x^{\mathrm{C}}_k\right)^2 + \left(y_k - y^{\mathrm{C}}_k\right)^2}}\right] - \theta^{\mathrm{C}} + n^\gamma_k. \end{cases} \tag{6}$$

In the third equation of (6), the constant μ is chosen so that the product μz_k corresponds to the height of the lower face's bound (i.e., approximately the chin). Again, the noises n^α_k, n^β_k, and n^γ_k are zero-mean Gaussians with $\sigma_\alpha = \sigma_\beta = \pi/45$ rad and $\sigma_\gamma = \pi/30$ rad.

Please note that in order to estimate the real absolute position of a human in the environment, the position and heading of the robot used in (3) and (5) should be provided by an accurate localization system. However, considering that our objective is only to track humans relative to the robot's position, the cumulative error of the odometry is not an issue. Furthermore, the odometry error between two consecutive estimations is very small and can be safely included in the noises of the observation models.

B. Multisensor Estimation With UKF

Kalman filtering is a well-known technique for multisensor tracking [20], which is also proved to be optimal for linear systems in case of Gaussian distributions. When the system is not linear, such as in our case, an extended Kalman filter (EKF) could be adopted; however, better performance is, in general, achieved by using a UKF [21]. The main difference is that the first-order linearization of the EKF is substituted in the latter with an unscented transformation (UT), which captures the mean and covariance of the probability distributions with carefully chosen weighted points, called "sigma points." In contrast with particle filters, the small number of points used by the UKF makes this estimator particularly appealing for real-time applications with limited computational power. Moreover, the advantage of the UKF with respect to the EKF is that the absence of linearization improves the estimation performance and avoids the calculus of Jacobian matrices. Most of the approaches found in the literature for tracking persons with a mobile robot are based on the EKF [22], [23] or on particle filters [8], [9], [18]. However, system nonlinearities and hardware constraints suggest the application of the UKF for our tracking system.

1) UT: Given the current state mean $\bar{\mathbf{x}}$, which is of size n, and its covariance matrix $\mathbf{P_{xx}}$, the $2n+1$ sigma points $\boldsymbol{\mathcal{X}}_i$ and associated weights W_i of the relative UT are calculated as follows:

$$\begin{aligned}
\boldsymbol{\mathcal{X}}_0 &= \bar{\mathbf{x}} \\
W_0 &= \rho/(n+\rho) \\
\boldsymbol{\mathcal{X}}_i &= \bar{\mathbf{x}} + \left[\sqrt{(n+\rho)\mathbf{P_{xx}}}\right]_i \\
W_i &= 1/\{2(n+\rho)\} \\
\boldsymbol{\mathcal{X}}_{i+n} &= \bar{\mathbf{x}} - \left[\sqrt{(n+\rho)\mathbf{P_{xx}}}\right]_i \\
W_{i+n} &= 1/\{2(n+\rho)\} \qquad (7)
\end{aligned}$$

where $i = 1, \ldots, n$ and ρ is a parameter for tuning the higher order moments of the approximation, normally set so that $n+\rho = 3$ for Gaussian distributions. The term $[\sqrt{(n+\rho)\mathbf{P_{xx}}}]_i$ is the ith column or row of the matrix square root of $\mathbf{P_{xx}}$.

Using these points, the mean and covariance of a generic (nonlinear) transformation $\mathbf{y} = \mathbf{g}(\mathbf{x})$ are calculated as follows:

$$\boldsymbol{\mathcal{Y}}_i = \mathbf{g}(\boldsymbol{\mathcal{X}}_i) \qquad (8)$$

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n} W_i \boldsymbol{\mathcal{Y}}_i \qquad (9)$$

$$\mathbf{P_{yy}} = \sum_{i=0}^{2n} W_i [\boldsymbol{\mathcal{Y}}_i - \bar{\mathbf{y}}][\boldsymbol{\mathcal{Y}}_i - \bar{\mathbf{y}}]^{\mathrm{T}}. \qquad (10)$$

As shown in [21], this procedure yields to a projected mean and covariance, which is correct up to the second order and is better than the linearization used by the EKF; however, it still keeps the same computational complexity.

2) UKF Estimation: The estimation procedure of the UKF, applied to our system, is the following. First of all, given the state vector $\mathbf{x}_k = [x_k, y_k, z_k, \phi_k, v_k]^{\mathrm{T}}$ of size $n = 5$, a UT is performed. This takes the last estimate $\hat{\mathbf{x}}_{k-1}$ and its relative covariance $\mathbf{P}_{k-1}$ to generate, using (7), the $2n+1 = 11$ sigma points $\boldsymbol{\mathcal{X}}_{ik-1}$. Note that in this case, the tuning parameter assumes a negative value $\rho = 3 - n = -2$. Julier *et al.* [24] showed that $\rho < 0$ can lead to a nonpositive semidefinite matrix when the state covariance is calculated with (10), and this was indeed a problem in our first implementation. As suggested by the authors, a modified version can be used, which consists in adding a term $[\boldsymbol{\mathcal{Y}}_0 - \hat{\mathbf{y}}][\boldsymbol{\mathcal{Y}}_0 - \hat{\mathbf{y}}]^{\mathrm{T}}$ to the sum in (10).

Therefore, with our prediction model $\mathbf{f}(\mathbf{x}_{k-1})$ defined in (2), the *a priori* estimate $\hat{\mathbf{x}}_k^-$ and covariance $\mathbf{P}_k^-$ are computed as follows:

$$\boldsymbol{\mathcal{X}}_{i_k^-} = \mathbf{f}(\boldsymbol{\mathcal{X}}_{ik-1}), \qquad \text{for} \qquad i = 0, \ldots, 10 \qquad (11)$$

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{10} W_i \boldsymbol{\mathcal{X}}_{i_k^-} \qquad (12)$$

$$\begin{aligned}
\mathbf{P}_k^- = &\sum_{i=0}^{10} W_i \left[\boldsymbol{\mathcal{X}}_{i_k^-} - \hat{\mathbf{x}}_k^-\right]\left[\boldsymbol{\mathcal{X}}_{i_k^-} - \hat{\mathbf{x}}_k^-\right]^{\mathrm{T}} \\
&+ \left[\boldsymbol{\mathcal{X}}_{0_k^-} - \hat{\mathbf{x}}_k^-\right]\left[\boldsymbol{\mathcal{X}}_{0_k^-} - \hat{\mathbf{x}}_k^-\right]^{\mathrm{T}} + \mathbf{Q} \qquad (13)
\end{aligned}$$

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

$$\begin{aligned}
\mathbf{Q} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_v^2 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 0 & \frac{\pi^2}{81} & 0 \\ 0 & 0 & 0 & 0 & 10^{-2} \end{bmatrix}. \qquad (14)
\end{aligned}$$

The next step is to generate the expected observation for the legs' measurement $\mathbf{z}_k = [b_k, r_k]^{\mathrm{T}}$ and face measurement $\mathbf{z}_k = [\alpha_k, \beta_k, \gamma_k]^{\mathrm{T}}$. Using the observation model $\mathbf{h}(\mathbf{x}_k)$, defined, respectively, in (4) and (6), and the sigma points predicted in (11), the UT is applied again as follows:

$$\boldsymbol{\mathcal{Z}}_{ik} = \mathbf{h}\left(\boldsymbol{\mathcal{X}}_{i_k^-}^-\right), \qquad \text{for} \qquad i = 0, \ldots, 10 \qquad (15)$$

$$\hat{\mathbf{z}}_k = \sum_{i=0}^{10} W_i \boldsymbol{\mathcal{Z}}_{ik} \qquad (16)$$

$$\begin{aligned}
\mathbf{P}_{\nu\nu k} = &\sum_{i=0}^{10} W_i [\boldsymbol{\mathcal{Z}}_{ik} - \hat{\mathbf{z}}_k][\boldsymbol{\mathcal{Z}}_{ik} - \hat{\mathbf{z}}_k]^{\mathrm{T}} \\
&+ [\boldsymbol{\mathcal{Z}}_{0k} - \hat{\mathbf{z}}_k][\boldsymbol{\mathcal{Z}}_{0k} - \hat{\mathbf{z}}_k]^{\mathrm{T}} + \mathbf{R} \qquad (17)
\end{aligned}$$

where $\hat{\mathbf{z}}_k$ is the predicted observation, $\mathbf{P}_{\nu\nu k}$ is the innovation covariance, and $\mathbf{R}$ is the covariance of the observation noise. In case of laser readings, the latter is set as follows:

$$\mathbf{R} \equiv \mathbf{R}^{\mathrm{L}} = \begin{bmatrix} \sigma_b^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} = \begin{bmatrix} \frac{\pi^2}{3600} & 0 \\ 0 & 10^{-2} \end{bmatrix}. \qquad (18)$$

For the camera, however, the following matrix is used:

$$\mathbf{R} \equiv \mathbf{R}^{\mathrm{C}} = \begin{bmatrix} \sigma_\alpha^2 & 0 & 0 \\ 0 & \sigma_\beta^2 & 0 \\ 0 & 0 & \sigma_\gamma^2 \end{bmatrix} = \begin{bmatrix} \frac{\pi^2}{2025} & 0 & 0 \\ 0 & \frac{\pi^2}{2025} & 0 \\ 0 & 0 & \frac{\pi^2}{900} \end{bmatrix}. \tag{19}$$

The cross correlation $\mathbf{P}_{\mathbf{xz}k}$ and the gain $\mathbf{K}_k$ are then computed using the following:

$$\mathbf{P}_{\mathbf{xz}k} = \sum_{i=0}^{10} W_i \left[\boldsymbol{\mathcal{X}}_{i_k}^- - \hat{\mathbf{x}}_k^-\right] \left[\boldsymbol{\mathcal{Z}}_{ik} - \hat{\mathbf{z}}_k\right]^{\mathrm{T}} \tag{20}$$

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{xz}k} \mathbf{P}_{\nu\nu k}^{-1}. \tag{21}$$

Finally, given the innovation $\nu_k = (\hat{\mathbf{z}}_k - \mathbf{z}_k)$, which is the difference between the predicted and real measurements provided either by the laser or the camera, the *a posteriori* estimate $\hat{\mathbf{x}}_k$ and its covariance $\mathbf{P}_k$ are calculated as follows:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \nu_k \tag{22}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{\nu\nu k} \mathbf{K}_k^{\mathrm{T}}. \tag{23}$$

3) Sequential Update: In case of asynchronous uncorrelated measurements, the correction step of a Kalman filter can be performed sequentially using only the data of the sensors available at the considered instant [20], [25]. Furthermore, even when all the measurements are synchronized, the sequential update, starting from the most to the least precise sensor data, gives a better estimation for nonlinear systems and is computationally more efficient.

Under the same assumptions, i.e., the measurements provided by the robot's laser and camera are independent, the UKF can also be updated sequentially with the same benefits. In case both the measurements are available at the same time step, the filter is updated first using the laser data, which is more accurate, and then the camera. The procedure is shown schematically in Fig. 9 for a single iteration. After the prediction step, the UKF is initially updated by the laser, provided that some legs have been actually detected. Then, in case a face observation is also available, the estimate is further corrected. This assures that all the available measurements are processed in order to get always the best possible estimate. The modularity of the approach also permits an easy integration of future extensions such as motion detection, sound detection, etc.

C. Data Association

To handle multiple targets, we adopted a nearest neighbor (NN) data association [20], which is a reasonable compromise between performance and computational cost, giving good results in most of the cases where the set of entities to track is not too dense [18], [19].

For each candidate track, the observations are predicted using the relative models. Then, after a gating procedure, a measure of similarity between predicted and real observations is used to fill an association matrix $\mathbf{S}_{M\times N}$, where M is the number of sensor measurements and N is the number of tracks. Finally, the elements of $\mathbf{S}_{M\times N}$ with the highest similarities are chosen, and

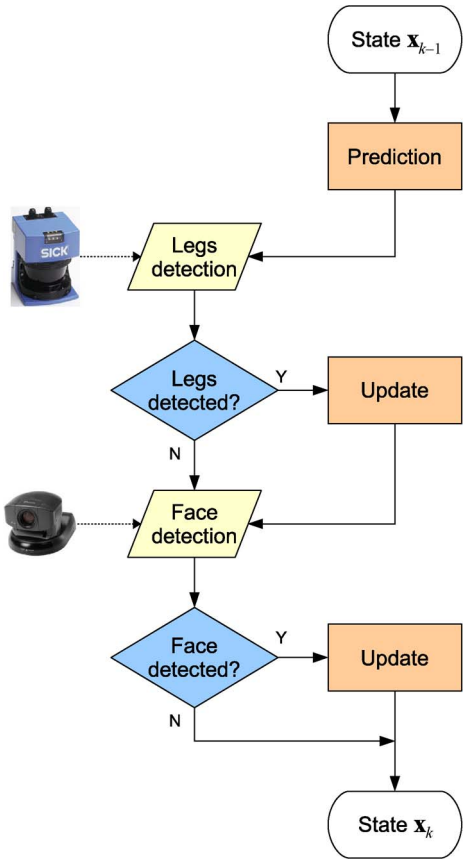

Fig. 9. Sensor data fusion with sequential UKF estimation.

each measurement m is used to update the associated track n. Note that NN is a one-to-one association, i.e., only one measurement is assigned to one track and *vice versa*.

We adopt a common gating approach, excluding all the measurements $\mathbf{y}_m$ outside a validation region. This region is constructed around the predicted observation $\hat{\mathbf{z}}_n$ according to the relation $d_{mn} \leq \lambda$, where λ is a threshold and d_{mn} is the Mahalanobis distance

$$d_{mn} = \sqrt{(\mathbf{y}_m - \hat{\mathbf{z}}_n)^{\mathrm{T}} \boldsymbol{\Sigma}_{mn}^{-1} (\mathbf{y}_m - \hat{\mathbf{z}}_n)} \tag{24}$$

with $\boldsymbol{\Sigma}_{mn}$ as the covariance matrix of the innovation $(\mathbf{y}_m - \hat{\mathbf{z}}_n)$. The value of λ can be determined from tables of the χ^2 distribution. In order to have a probability $P_G = 0.99$ that a measurement generated by a human target falls inside the validation region, we chose $\lambda_{\mathrm{L}} = 3.03$ for the legs' measurements and $\lambda_{\mathrm{C}} = 3.37$ for faces.

To create the association matrices, one for legs and one for faces, we make use of the following similarity measure [26]:

$$s_{mn} = \frac{1}{\sqrt{(2\pi)^\eta |\boldsymbol{\Sigma}_{mn}|}} \exp\left(-\frac{d_{mn}^2}{2}\right) \tag{25}$$

where η is the size of the observation vector ($\eta = 2$ for legs and $\eta = 3$ for faces). It is clear that the bigger the value of s_{mn}, the higher the similarity between $\mathbf{y}_m$ and $\hat{\mathbf{z}}_n$.

D. Creating and Removing Tracks

The sensor readings discarded by the gating or the assignment procedure are used to create new tracks. Different criteria can be adopted in order to discriminate false positives, which would otherwise generate nonexisting tracks. The most reliable solution would be to consider a new person only when both legs and face are detected. However, it would not be possible to create tracks out of the camera's field of view, which is a big limitation in many applications (e.g., when the robot looks for someone to interact with or when it needs to avoid people walking nearby). Instead, we prefer to use mainly the legs detected by the laser, exploiting the differences among leg patterns. In particular, only LA and FS patterns are selective enough to be considered reliable for the task. The remaining SL pattern, instead, can contribute to the creation of new tracks only when a face is also detected on the same direction of the legs.

The whole procedure, then, is implemented as follows. Parallel to the human tracks database, we keep another list containing all the possible candidates. Each one of these is generated by a sequence of laser readings, which can be either LA, FS, or SL legs (the latter is validated by faces). The readings have to fall inside a certain region, delimited by the distance covered by a person when moving at the maximum speed of 1.5 m/s. Each candidate is also assigned a maximum lifetime. If during this interval there are enough readings falling inside its region, the candidate is promoted to human track; otherwise, it is considered a false positive and removed.

Of course, proper tracks can be updated with any of the leg patterns or faces. They are eventually deleted from the database if not updated for more than 2 s or if the uncertainty of their 2-D position is too big, i.e., the sum of the variances in x and y is greater than 2 m^2.

IV. Experimental Results

To test the performance and the portability of the proposed solution, the system has been implemented on two different mobile robots. The first one is a Pioneer 2, shown in Fig. 1, which is provided with a Sick laser and a Pan-Tilt-Zoom (PTZ) camera. This is mounted on a special support at approximately 1.5 m from the floor in order to facilitate the face detection. The onboard PC is a Pentium III 800 MHz with 128 MB of RAM. The second one is an interactive service robot based on a Scitos G5 platform, as shown in Fig. 10. This is also provided with a laser and a camera, the latter being embedded in the robotic head. The onboard computer is a Core Duo 1.66 GHz with 1 GB of RAM. A touch screen is also available for interaction. Both robots run on a Linux operating system.

The whole software has been written in C++ and runs in real time on the robot PCs, although it is possible to use an external client, connected via wireless, for remote control and debug. The resolution of the laser devices is ± 1 cm, with a scan every 0.5° at 5 Hz for the Pioneer and 32 Hz for the Scitos, whereas the cameras provide images with a resolution of 320×240 pixels at 10 and 25 Hz, respectively. The updating frequency of our program, which includes other functionalities for motion control and data logging, is approximately 4 Hz for the Pioneer and 16 Hz for the Scitos.

Fig. 10. Scitos robot with laser and embedded camera in the robotic head.

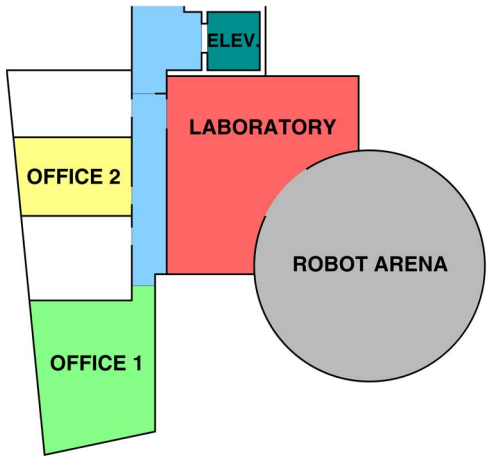

Fig. 11. Floor plan of the test environment.

The dimensional constraints of the leg detection have been empirically determined after analyzing many recorded data of different people walking in typical indoor environments. The best results have been obtained setting the width of a leg to $10 < a < 20$ cm, the maximum step length to $b < 40$ cm, and the width of two adjacent legs to $10 < c < 40$ cm. These values are still valid for small changes of the laser's position, as is the case with our robots.

The experiments have been conducted in our laboratory, a robot arena, and adjacent offices, as shown in Fig. 11, moving between different rooms and, often, crossing doorways or narrow corridors. During the experiments, the robots were controlled remotely to follow the persons being tracked, often moving faster than 0.5 m/s and with a turn rate of up to $45°/s$. The performance of the system is also documented by several videos in the multimedia appendix. This will be available at http://ieeexplore.ieee.org.

Fig. 12. Cluttered environment for leg-detection experiment.

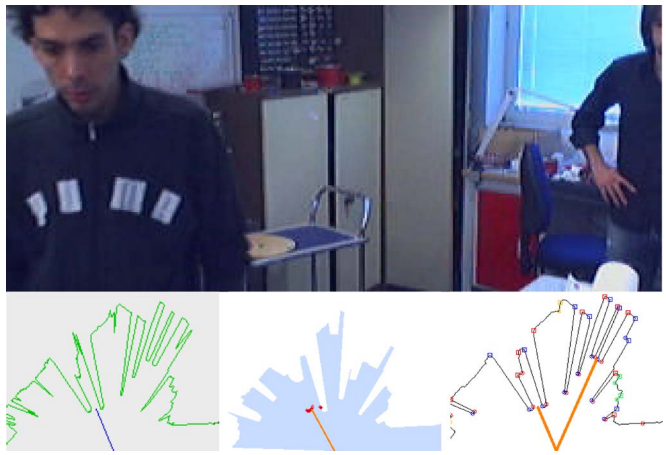

Fig. 13. Example of leg detection with the following three different algorithms: minima on the left, motion in the middle, and patterns on the right. In this case, the person on the right is missed by the first two algorithms.

A. Leg Detection

To evaluate the performance of the leg-detection algorithm, two other techniques frequently found in the literature have been implemented for comparison. The first one is a procedure adopted in [6] and [11], which finds the distance minima in a laser scan and then keeps those points possibly generated by legs, discarding all the others. The minima are extracted from sequences of (at least four) adjacent laser readings with a gradient lower than 0.1 m/°. Legs in this case must be at 30 cm or more from any background object, and pairs are grouped if the inner gap is less than 0.5 m.

The second procedure implemented is the motion-detection algorithm described in [3], which is similar to other variants that can be found in the literature [27], [28]. The basic idea consists in detecting entities that violate a "free space," which is the union of empty regions built from past laser scans, using the odometry to compensate the ego motion of the robot. If a certain number of readings fall inside a region that was previously unoccupied, then these readings are supposed to belong to a moving object. The implementation used for our comparison considers the last three laser scans and a margin difference of 0.2 m between consecutive free spaces.

The algorithms for leg detection have been tested on data recorded in the following two different situations: 1) pioneer robot in a static position, pointing the laser toward a cluttered area of the laboratory, with up to three people walking around, and 2) robot moving together with some persons, following one of them inside a cluttered office and between different rooms. The environment of the first case is shown in Fig. 12, with a picture taken from the top of the robot. As can be seen, several objects, such as tables, chairs, bags, etc., contributed to make the detection more difficult. The second case is represented by some snapshots taken with the robot's camera, which are shown in Figs. 15 and 16. Aside from the laboratory, this test scenario also included a cluttered office, a corridor, and the robot arena.

The total number of laser scans recorded from the static position was 813, from which we manually counted 1067 "detectable" persons, i.e., with legs not occluded by other objects. An example of detection with the three different approaches is shown in Fig. 13. Our algorithm, looking for all the three patterns of LA, FS, and SL, confused only one wrong

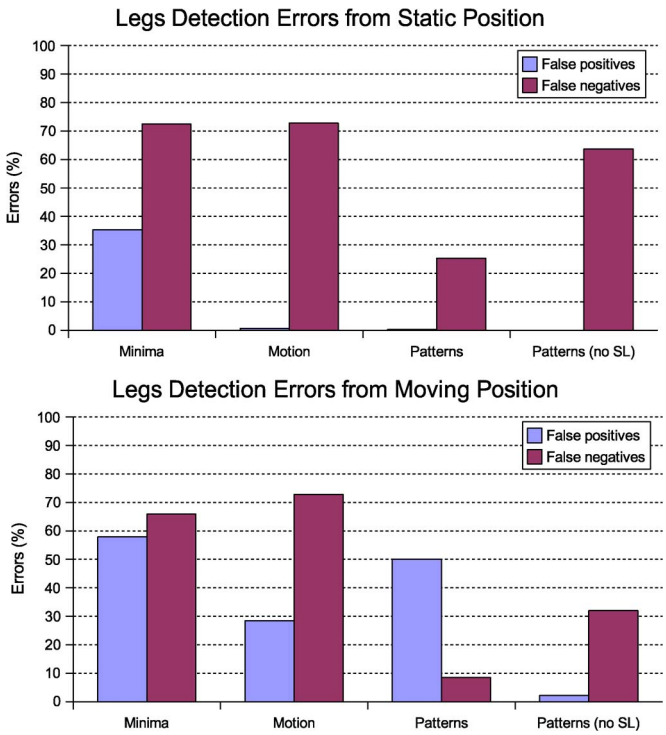

Fig. 14. Comparison of leg-detection errors using algorithms based on distance minima, motion detection, and leg patterns (with and without SL). Errors are expressed as ratios of false positives versus total leg detections and as false negatives versus total detectable persons.

object as a leg (false positive) out of 798 detections and totally missed 270 legs (false negatives) during the experiment. The performance of the detection is compared with the other two algorithms, which are the minima and motion detection, and with a modified version of our procedure that does not consider SL patterns. The results are shown in Fig. 14, which reports the ratio (in percentage) between the number of false positives and total detections. It also reports the ratio between the number of false negatives and total number of actual legs. As the first graph shows, the percentage of false positives is almost null for all but the minima case, where some false positives were occasionally generated by a moving chair or a bag. Our algorithm performed

considerably better than the motion detection in the number of false negatives, considering that the latter obviously missed all the cases where a person was standing still.

Probably more significant are the results from the experiment with the robot moving and shown by the second graph in Fig. 14. The laser scans recorded in this case were 619, which contained, totally, 802 persons. As expected, the number of false positives increased considerably because of the dynamic nature of the experiment, where almost each laser scan differs from the next one. For our detection algorithm, using all the three leg patterns, the percentage of false positives was 49.97%, bigger than the 28.37% of the motion-detection procedure. However, compared with the latter, our algorithm had a much lower percentage of false negatives, which is 8.48%, against the 72.82% of the motion detection. The latter indeed failed very often during the "following" behavior, considering that a person moving away from the robot does not violate the free space and, therefore, cannot be detected. Note also that by using our algorithm without considering the SL patterns, the false positives were reduce to very little, still keeping the percentage of false negatives much lower than that for the minima and motion detection. Usually, the best solution would be a balance between false positives and false negatives, which, in this case, could be further modified by adjusting the dimensional constraints of the leg patterns. However, in the next experiments, we found it more appropriate, for a robust tracking, to keep the current settings and consider all the three patterns, relying instead on the gating procedure to discard possible false positives.

B. Tracking in Cluttered Environments

A challenging task for mobile robots is following and tracking a person inside furnished rooms. The following experiment was conducted with the Pioneer robot in office 1. Inside, there were several desks, chairs, metallic shelves, one of which was located in the middle of the room, and many other objects, such as trash bins and school bags, lying on the floor. We wanted to represent a situation in which an instructor shows the environment to the robot, walking at normal speed and slowing down from time to time to give some indications. The person walked around the room twice, always followed by the robot at approximately 1.5 m, and each turn took about 30 s. A few snapshots of the experiment are shown in Fig. 15 (see also the attached Video 1, and this will be available at http://ieeexplore.ieee.org).

During the experiment, the face was detected only for 30% of the whole time; thus, the track estimate was mainly based on the laser. As reported in Table II (experiment "clutters"), which shows the ratios between the number of detected legs and faces versus the total tracking steps, the person's legs were indeed detected for 92% of the total tracking time. In particular, 49% of these legs were LA patterns, 13% were FS, and 30% were SL.

C. Following Across Different Rooms

Another difficult situation is when the robot has to move between different rooms, keeping track of a human along narrow corridors or door passages. This would be the case, for example, when a robot has to follow an instructor in the environment for map building.

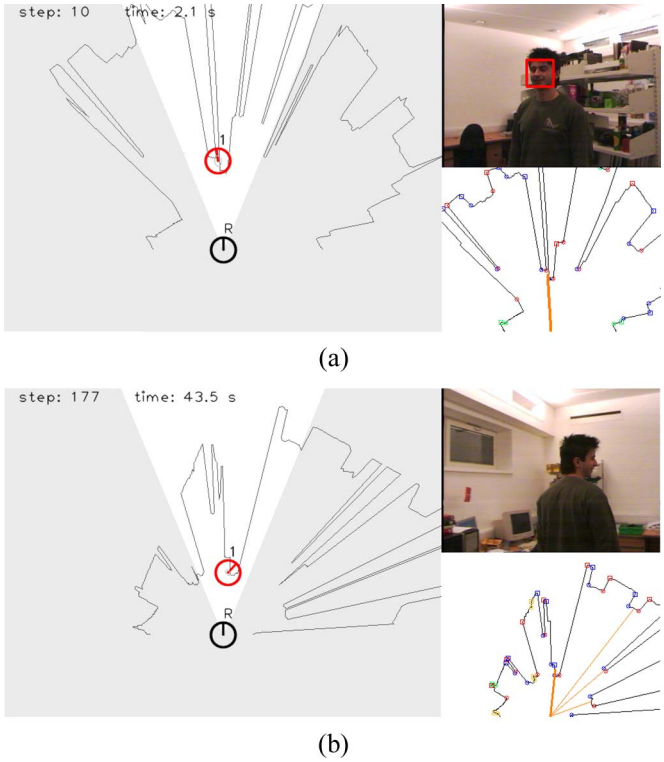

Fig. 15. Person walking around the room followed by the robot. Face and leg detections are shown on the right side of each figure. The left part illustrates the track and the robot "R," also highlighting the raw laser scan and the camera's field of view. (a) Both face and LA legs of the person are detected. (b) Detection of the FS leg pattern, together with some false positives caused by the furniture. Nevertheless, the tracking is always performed correctly.

TABLE II
OBSERVATION/TRACKING TIME RATIO

Experiment	LA Legs	FS Legs	SL Legs	Faces
clutters	49%	13%	30%	30%
follow 1	58%	12%	19%	30%
follow 2	16%	19%	46%	3%
interaction	47%	6%	32%	34%

In the experiments described next, one or more persons were followed by the Pioneer robot while moving from office 1 to the robot arena shown in Fig. 11. Along the path, two doors and two other rooms had to be crossed. A short sequence of images taken during the trials are shown in Figs. 16 and 17 (see also Videos 2 and 3, and these will be available at http://ieeexplore.ieee.org).

Initially, the robot started tracking a couple of persons in office 1 and then followed one of them to the robot arena. The first part of the path included a door passage and a narrow corridor, with some objects on the floor and a column that made the leg detection more difficult. As shown in Fig. 16(a), a sign on the wall also generated a false positive on the face detection. Nevertheless, the human tracking continued successfully; the robot entered the door in Fig. 16(b), where the person was out of the camera's view, and crossed the laboratory in Fig. 16(c)

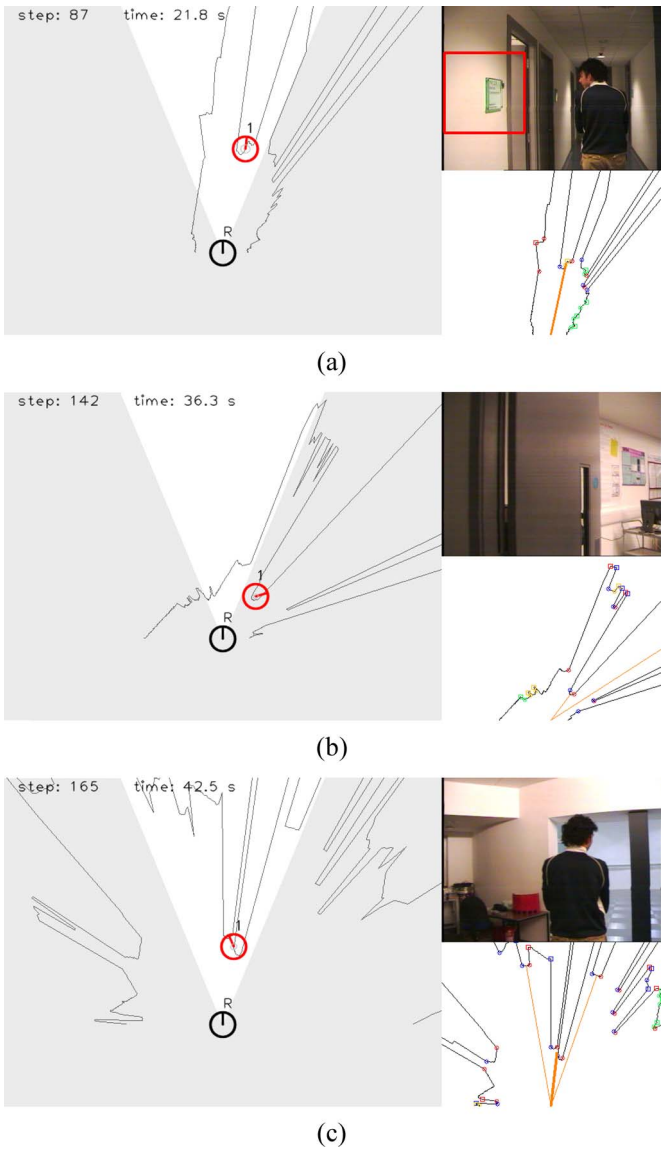

Fig. 16. Tracking one person between different rooms. (a) False positive of the face detection, correctly discarded. (b) Door passage with the target outside the camera's field of view. (c) Crossing the laboratory toward the robot arena.

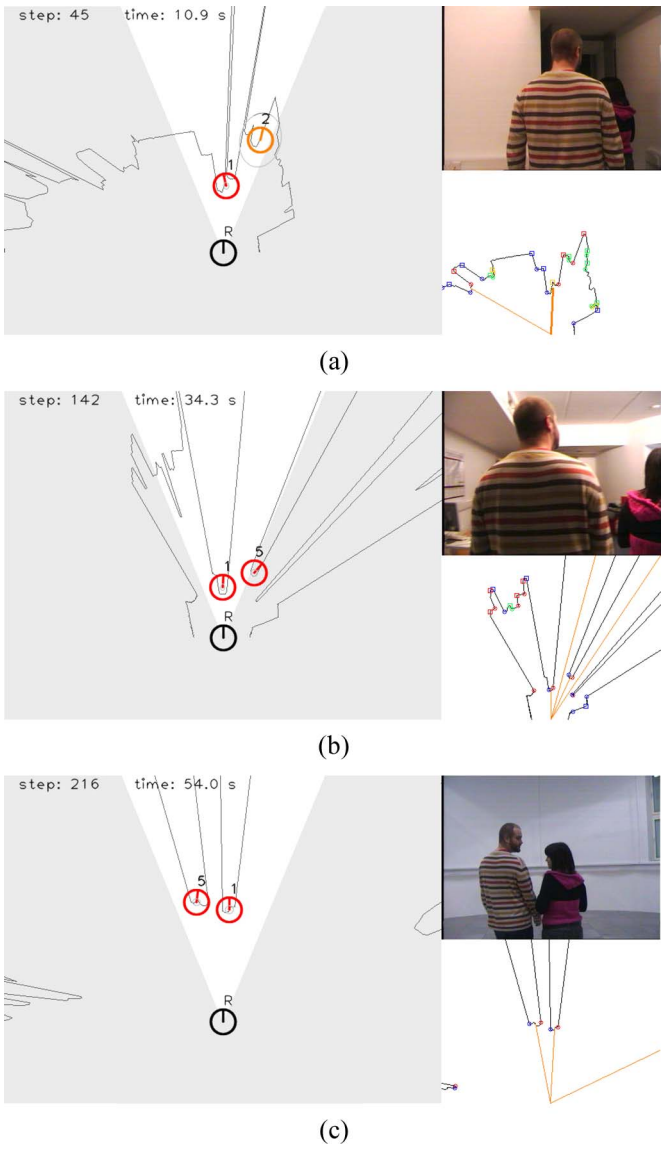

Fig. 17. Two persons being tracked from office 1 to the robot arena. (a) Initial tracks of two persons walking out of office 1. (b) Both the persons being tracked in the laboratory. (c) Track swap due to data-association error.

to finally reach the robot arena. Eventually, some other people joined the previous person in the same room.

A similar experiment was also conducted to test the performance of the tracking system in case the robot follows more than one person at the same time. The situation was particularly challenging because, walking very close, people occlude each other very often and the data-association problem becomes really difficult.

Fig. 17(a) shows the initial tracking of two persons, in office 1, who were walking toward the corridor to reach the robot arena. During the initial path, the person closest to the robot (target 1) was often occluding the other one, whose track was therefore lost several times. However, once they reached the laboratory, the robot could finally detect both and keep two distinct tracks, as shown in Fig. 17(b), until the final destination.

Although the tracking was generally correct most of the time, we have to note that inside the robot arena, a data-association error caused a swap between the tracks. This happened because the two persons were walking very close to each other, with similar orientation and speed and without facing the robot's camera. The error is clear when comparing Fig. 17(b) with Fig. 17(c), where the track of the tall subject (target 1) swapped with the other one's (target 5). In situations like this, a more sophisticated data-association algorithm would probably perform better but at a higher computational cost. Moreover, the integration of an additional visual tracker, such as the histogram-based solution of Comaniciu *et al.* [29], could help in keeping the tracks apart even when the faces are not detectable.

Compared with the previous cases, the results in Table II (experiment "follow 2") also show a different contribution of the detected leg patterns to the tracking. Indeed, this time, the LA pattern was that one giving the smaller contribution to the estimation. This was because the experiment has been performed almost exclusively with people in motion, when most of the postures detected by the laser are therefore FS or SL patterns. In addition, considering that the persons were almost

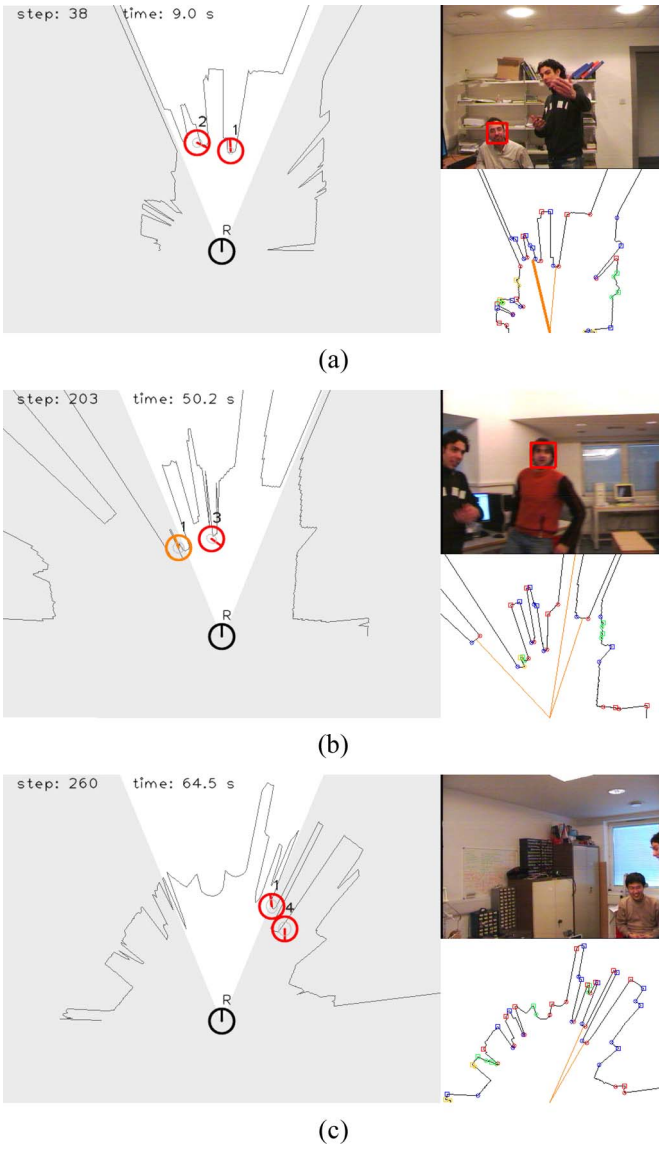

Fig. 18. Pioneer robot is introduced to some people in different rooms. (a) Tracking in office 2, where one of the persons is initially seated. (b) Only the face is used to update the track of target 3. (c) Track swap error, out of the camera's view.

always heading forward, very few faces were detected during the trial.

D. Application to Human-Robot Interaction

This experiment replicates a possible case where a service robot is introduced to several people, which interacts with them in various locations. Considering that our research also includes people recognition, this part is very important for the objective. Indeed, situations similar to that one described in this paper could be used to train the robot with different individuals and recognize them under different conditions.

The trial started in office 2, with the Pioneer robot following the two people shown in Fig. 18(a) (see also Video 4, and this will be available at http://ieeexplore.ieee.org). Considering that the legs of the left person were promptly detected, he was tracked almost immediately, even if he was still sitting

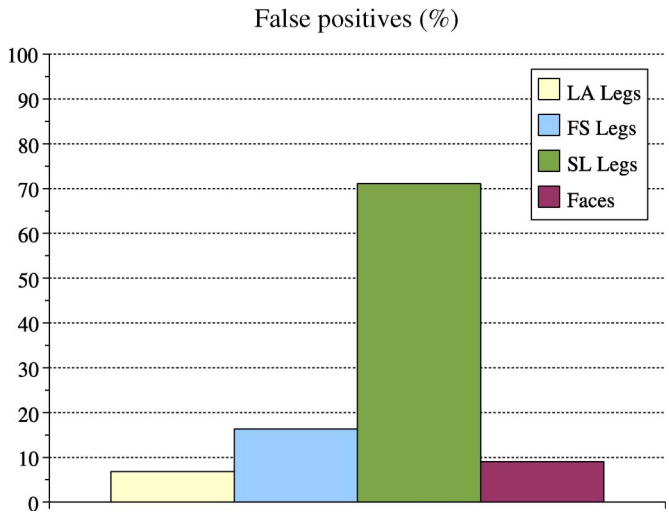

Fig. 19. Percentages of false positives for leg and face detections versus total number of human detections.

on a chair. Then, always following and keeping track of one person (target 1), the robot moved to the laboratory, reaching the desk of another colleague. Fig. 18(b) shows a case where one estimate (target 3) was updated only by the face detection, which also increased the robustness of the tracking in several other occasions.

The experiment terminated on another side of the laboratory, where the robot had to track up to three people at the same time. During this part, there was an error, considering that the tracks of the two persons shown in Fig. 18(c) (targets 1 and 4) swapped once when they got too close to each other. The main reasons for this problem were the temporary misdetection of one target and the low update frequency of the program. Indeed, a time step of 0.25 s (i.e., 4-Hz update) can sometimes be too high for a good estimation, particularly when both robot and persons are moving fast. Compared with the previous error in Section IV-C, this one differs in that it happened out of the camera's field of view; therefore, vision could not help avoiding it. With the future integration of a human-recognition system, the track labels could be corrected as soon as the subject is captured by the camera again.

E. Evaluation of Human Detection and Tracking

To evaluate the accuracy of the human detection, Fig. 19 shows the ratios between the number of false positives versus the total number of leg and face detections for all the tracking experiments described so far (Sections IV-B, -C, and -D) with the Pioneer robot. As expected, we can notice that faces and LA patterns had very few false positives (9.0% and 6.8%, respectively). However, the number of false positives for the SL patterns was very high, although this was correctly handled by the gating procedure. Nevertheless, the contribution given by these patterns was also very important. Indeed, in repeating the experiments with the same data but without using SL legs, the tracks were sometimes lost. When considering all the three patterns, however, the tracks were constantly maintained as long as they were in the detection range.

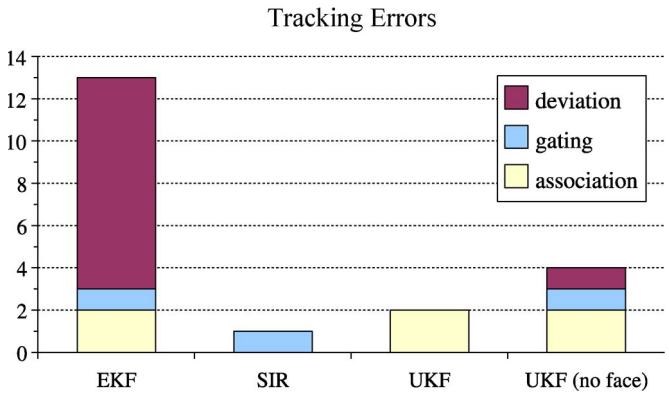

Fig. 20. Comparison of tracking errors with different filters.

With the same data recorded during the previous experiments, for a total of 1180 time steps, we also repeated the tracking using a classic EKF and a sampling importance resampling (SIR) particle filter, which are both very common in the literature, and compared the performance with our solution. The Jacobian matrices for the EKF have been calculated from the state and observation models in (2), (4), and (6). The estimation was performed using the well-known Kalman equations [30] and adopting the same noise covariances defined in (14), (18), and (19). The SIR particle filter [31] was implemented using the models and noises cited earlier. In particular, the prior distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and the likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ were, respectively, the Gaussians $\mathcal{N}[\mathbf{x}_k; \mathbf{f}(\mathbf{x}_{k-1}), \mathbf{Q}]$ and $\mathcal{N}[\mathbf{z}_k; \mathbf{h}(\mathbf{x}_k), \mathbf{R}]$, using the same quantities already defined in Section III-B2. The number of samples used was 1000, similar to other existing solutions.

We measured the robustness of the different techniques in terms of tracking errors, considering only the 2-D coordinates (x, y) for the sake of simplicity. Each one of the following situations was counted as an error: 1) the track deviates from the correct trajectory of the human target and is eventually deleted by the system; 2) the track "jumps" to a static object, adjacent to the path of the person, due to a false positive (gating error); and 3) the track switches to another person close to the original one (data-association error). All these cases, indeed, are strictly related to the estimate of the filter and to the distribution of its uncertainty. The graph in Fig. 20 clearly shows that the result obtained with the UKF is better than that with the EKF. The nonlinearity of the system, in fact, made the latter filter fail in several occasions, particularly when both the robot and the person being tracked were moving. The performance of the UKF was instead similar to the SIR estimation in terms of error number; however, it differs on the type of error. Despite an occasional error due to a false positive in the cluttered office 1, the major accuracy of the particle filter in representing the probability distribution of the estimate seemed to be an advantage for the data association; thus, the previous errors shown in Figs. 17(c) and 18(c) did not take place. Note, however, that a solution based on particle filters was not feasible for our Pioneer robot, as, unfortunately, it could not run in real time (even when using just a few hundreds samples) and, particularly, when two or more persons were tracked at the same time.

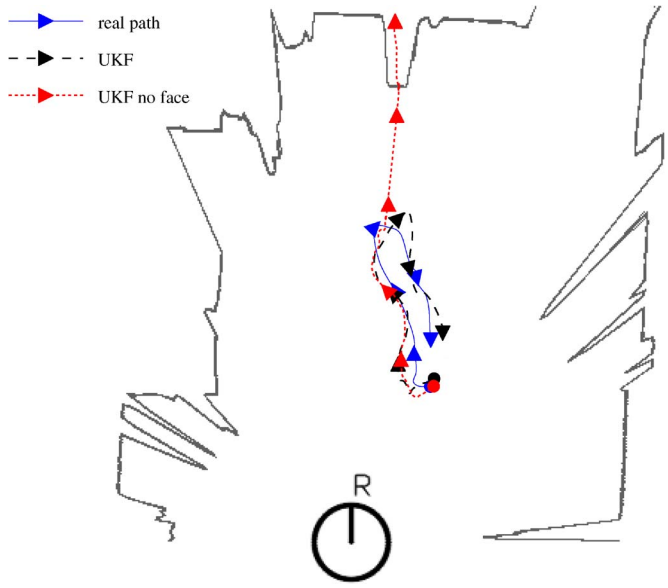

Fig. 21. Tracking with and without face detection. The real trajectory of person 1 is shown with the solid line; the UKF estimation, including face detection, is shown with the dashed line, and the one without face is represented by the dotted line. The start point of each line is highlighted by a circle.

Fig. 20 also shows the number of errors that occurred when using the same sensor data and applying the UKF without face detection. As already said before, legs were the most useful features during the experiments; hence, the tracking performance in the last case did not decrease excessively. However, aside from providing height information (that might be useful for human recognition), face detection improved the tracking robustness when people were facing the robot. An example of tracking with and without face detection is shown in Fig. 21 and is relative to the first part of the interaction experiment in Section IV-D, which is also shown in Fig. 18(a). The real path of the person (target 1), plotted in Fig. 21 for comparison, is manually extracted from a sequence of 30 time steps. It can be noticed that the original UKF followed the correct path quite well; however, the estimation without face detection failed when the person turned back toward the initial position.

F. Portability to Different Robot Platforms

The experiments reported so far have been carried out with a Pioneer robot. In order to test also the portability of the tracking system to different platforms, we performed several other tests with the Scitos robot shown in Fig. 10. One of these is shown in Fig. 22 (see also Video 5, and this will be available at http://ieeexplore.ieee.org), which shows a few moments of the robot following and tracking up to three people between the robot arena and the elevator. In some cases, one person was occluding the others, such as in the situation shown in Fig. 22(a). The tracking, however, continued successfully, and the robot eventually reached and entered the elevator with the two persons shown in Fig. 22(b). Note that even in this case, the track of the person on the right would have been lost without face detection because his legs are very close to the elevator's wall.

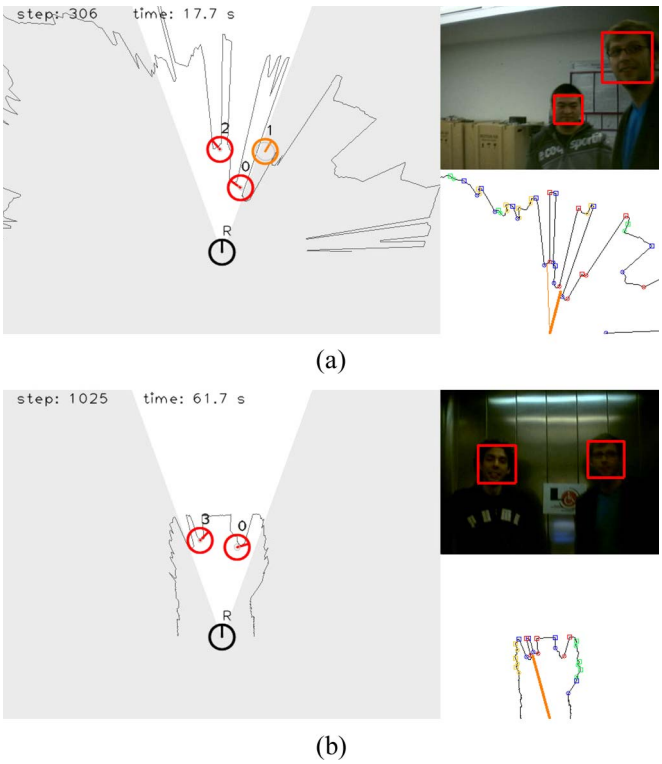

Fig. 22. Scitos robot tracking people in different locations. (a) Tracking three people simultaneously with occlusions. (b) Entering the elevator with two persons.

Despite the fact that the laser was positioned a few centimeters higher than on the Pioneer, the leg detection did not need any change in its parameters. The algorithm indeed showed to be quite robust and not too sensitive to those settings. The tracking performance was generally similar or better, thanks, particularly, to the faster update of the estimation. The approach seems, therefore, to be feasible for any mobile robot similar to ours, equipped with a laser and a camera.

V. Conclusion and Future Work

In this paper, we have presented a multisensor-based human-detection and tracking system for mobile service robots. The proposed approach is a practical and effective solution for real-time applications of interactive robots in populated environments. A hybrid approach to human tracking was adopted, which is based on a new algorithm for leg detection and a robust face detection. More precisely, the leg information extracted from a single laser scan has been fused to the face detected with a camera using a sequential implementation of UKF. This has demonstrated to be a good alternative to standard EKF, considering that it does not need linearization, and to particle filters, as it is computationally more efficient.

Several experiments have been presented to illustrate the good performance and the benefits of our solution. It has been shown that a mobile robot can perform accurate human detection, following one or more persons between different rooms and tracking users for possible human-robot interaction. Our approach works well even in challenging situations, where clutters and the size of the environment make human detection a difficult task. The portability of the proposed solution has also been verified successfully on different mobile platforms.

The current solution could be further improved increasing the success ratio of the leg-detection algorithm, for which other geometric features or pattern recognition techniques should be investigated and possibly integrated. Aside from this, our future research will focus on the data-association part for a more robust tracking of multiple people, particularly when they gather in front of the robot. We are also currently integrating a vision-based recognition system in order to identify users and perform dedicated human-robot interactions.

References

[1] W. Burgard, P. Trahanias, D. Hähnel, M. Moors, D. Schulz, H. Baltzakis, and A. Argyros, "TOURBOT and WebFAIR: Web-operated mobile robots for tele-presence in populated exhibitions," in *Proc. IROS Workshop Robots Exhib.*, 2002, pp. 1–10.

[2] J. N. K. Liu, M. Wang, and B. Feng, "iBotGuard: An Internet-based intelligent robot security system using invariant face recognition against intruder," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 1, pp. 97–105, Feb. 2005.

[3] M. Lindström and J.-O. Eklundh, "Detecting and tracking moving objects from a mobile platform using a laser range scanner," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Maui, HI, 2001, vol. 3, pp. 1364–1369.

[4] W. Zajdel, Z. Zivkovic, and B. J. A. Krose, "Keeping track of humans: Have I seen this person before?" in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 2093–2098.

[5] R. C. Luo, Y. J. Chen, C. T. Liao, and A. C. Tsai, "Mobile robot based human detection and tracking using range and intensity data fusion," in *Proc. IEEE Workshop Adv. Robot. Social Impacts*, 2007, pp. 1–6.

[6] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Plötz, G. A. Fink, and G. Sagerer, "Multi-modal anchoring for human–robot-interaction," *Robot. Auton. Syst.*, vol. 43, no. 2/3, pp. 133–147, May 2003.

[7] M. Scheutz, J. McRaven, and G. Cserey, "Fast, reliable, adaptive, bimodal people tracking for indoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, 2004, pp. 1347–1352.

[8] 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. Robot. Res.*, vol. 22, no. 2, pp. 99–116, 2003.

[9] P. Chakravarty and R. Jarvis, "Panoramic vision and laser range finder fusion for multiple person tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, 2006, pp. 2949–2954.

[10] G. Cielniak, A. Treptow, and T. Duckett, "Quantitative performance evaluation of a people tracking system on a mobile robot," in *Proc. 2nd Eur. Conf. Mobile Robots*, Ancona, Italy, 2005.

[11] N. Bellotto and H. Hu, "Multisensor integration for human–robot interaction," *IEEE Journal of Intelligent Cybernetic Systems*, vol. 1, Jul. 2005. [Online]. Available: http://www.cybernetic.org.uk/ics

[12] S. Feyrer and A. Zell, "Robust real-time pursuit of persons with a mobile robot using multisensor fusion," in *Proc. 6th Int. Conf. Intell. Auton. Syst.*, Venice, Italy, 2000, pp. 710–715.

[13] E. A. Topp and H. I. Christensen, "Tracking for following and passing persons," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, 2005, pp. 2321–2327.

[14] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[15] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proc. IEEE Int. Conf. Image Process.*, New York, 2002, vol. 1, pp. 900–903.

[16] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[17] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[18] M. Montemerlo, W. Whittaker, and S. Thrun, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, 2002, pp. 695–701.

[19] N. Bellotto and H. Hu, "People tracking with a mobile robot: A comparison of Kalman and particle filters," in *Proc. 13th IASTED Int. Conf. Robot. Appl.*, Wu¨rzburg, Germany, 2007, pp. 388–393.

[20] Y. Bar-Shalom and X. R. Li, *Multitarget–Multisensor Tracking: Principles and Techniques*. Storrs, CT: YBS Publishing, 1995.

[21] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
[22] D. Beymer and K. Konolige, "Tracking people from a mobile platform," in *Proc. IJCAI, Workshop Reasoning With Uncertainty Robot.*, Seattle, WA, 2001.
[23] J. Bobruk and D. Austin, "Laser motion detection and hypothesis tracking from a mobile platform," in *Proc. Australian Conf. Robot. Autom.*, Canberra, Australia, 2004. [Online]. Available: http://www.araa.asn.au/acra/acra2004
[24] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.
[25] Z. Duan, X. Li, C. Han, and H. Zhu, "Sequential unscented Kalman filter for radar target tracking with range rate measurements," in *Proc. 8th Int. Conf. Inform. Fusion*, Philadelphia, PA, 2005.
[26] *Handbook of Multisensor Data Fusion*, CRC Press, Boca Raton, FL, 2001.
[27] C. C. Wang and C. Thorpe, "Simultaneous localization and mapping with detection and tracking of moving objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 2918–2924.
[28] H. Zender, P. Jensfelt, and G. J. M. Kruijff, "Human-and situation-aware people following," in *Proc. 16th IEEE Int. Symp. Robot Human Interactive Commun.*, 2007, pp. 1131–1136.
[29] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Head Island, SC, 2000, vol. 2, pp. 142–149.
[30] G. Welch and G. Bishop, "An introduction to the Kalman filter," Univ. North Carolina, Chapel Hill, NC, Tech. Rep. 95-041, 2004.
[31] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

Nicola Bellotto (S'06) received the Laurea degree in electronic engineering from the University of Padua, Padova, Italy. He is currently working toward the Ph.D. degree in computer science in the Department of Computing and Electronic Systems, University of Essex, Colchester, U.K., where his doctoral thesis focuses on Bayesian estimation and sensor fusion for mobile service robotics.

He was an active member of the Intelligent Autonomous System Laboratory, University of Padua, Padua, Italy, and of the Centre for Hybrid Intelligent Systems, University of Sunderland, Sunderland, U.K. He is currently with the Human Centred Robotics Group, Department of Computing and Electronic Systems, University of Essex, he gained several years of professional experience as an Embedded Systems Programmer and Software Developer for entertainment robotics. His research interests include robot vision, localization, and embedded systems programming.

Huosheng Hu (M'94–SM'01) received the M.Sc. degree in industrial automation from the Central South University, Changsha, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993.

Currently, he is a Professor with the Department of Computing and Electronic Systems, University of Essex, Colchester, U.K., leading the Human Centred Robotics Group. His research interests include autonomous mobile robots, human–robot interaction, evolutionary robotics, multirobot collaboration, embedded systems, pervasive computing, sensor integration, RoboCup, intelligent control, and networked robotics. He has published over 250 papers in journals, books, and conferences. He was as a Member of the Editorial Advisory Board for the *International Journal of Industrial Robots* in 1997–2000, and is currently the Editor-in-Chief of the *International Journal of Automation and Computing*. He is a Reviewer for a number of international journals, such as IEEE TRANSACTIONS ON ROBOTICS, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON NEURAL NETWORKS, and the *International Journal of Robotics Research*. Since 2000, he has been a Visiting Professor with six universities in China, namely, Central South University, Shanghai University, Shanghai; Wuhan University of Science and Engineering, Wuhan; Kunming University of Science and Technology, Kunming; Chongqing University of Post and Telecommunication, Chongqing; and Northeast Normal University, Jilin.

Dr. Hu is the recipient of several best paper awards. He has been a founding member of Networked Robots of the IEEE Society of Robotics and Automation Technical Committee, since 2001, and was a member of the International Association for Science and Technology for Development Technical Committee on "Robotics," in 2001–2004. He was the General Cochair of the IEEE International Conference on Mechatronics and Automation, Harbin, China, in 2007; the Publication Chair of the IEEE International Conference on Networking, Sensing, and Control, London, in 2007; the Cochair of Special and Organized Sessions of the IEEE International Conference on Robotics and Biomimetics, Sanya, China, in 2007; the Chair for Special and Organized Sessions of the IEEE/American Society of Mechanical Engineers International Conference on Advanced Intelligent Mechatronics, Xi'an, China, in 2008; etc. He is a Chartered Engineer, a senior member of the Association for Computing Machinery, and a member of The Institution of Engineering and Technology and the Intelligent Autonomous Systems Society.