Ann. Telecommun. (2012) 67:355–375
DOI 10.1007/s12243-012-0311-1

# Data fusion in ubiquitous networked robot systems for urban services

**Luis Merino · Andrew Gilbert · Jesús Capitán · Richard Bowden · John Illingworth · Aníbal Ollero**

Received: 29 June 2011 / Accepted: 6 March 2012 / Published online: 16 June 2012

**Abstract** There is a clear trend in the use of robots to accomplish services that can help humans. In this paper, robots acting in urban environments are considered for the task of person guiding. Nowadays, it is common to have ubiquitous sensors integrated within the buildings, such as camera networks, and wireless communications like 3G or WiFi. Such infrastructure can be directly used by robotic platforms. The paper shows how combining the information from the robots and the sensors allows tracking failures to be overcome, by being more robust under occlusion, clutter, and lighting changes. The paper describes the algorithms for tracking with a set of fixed surveillance cameras and the algorithms for position tracking using the signal strength received by a wireless sensor network (WSN). Moreover, an algorithm to obtain estimations on the positions of people from cameras on board robots is described. The estimate from all these sources are then combined using a decentralized data fusion algorithm to provide an increase in performance. This scheme is scalable and can handle communication latencies and failures. We present results of the system operating in real time on a large outdoor environment, including 22 nonoverlapping cameras, WSN, and several robots.

**Keywords** Ubiquitous networked robots · Decentralized data fusion · Service and social robotics · Person tracking

## 1 Introduction

There is an increasing interest in service robotics, that is, robot systems that provide services to human users. The EU project called URUS (Ubiquitous Networking Robotics in Urban Settings) [43] considers a team of mobile robots, a set of static cameras and a wireless sensor network (WSN) in a urban environment to offer urban services. All these elements can communicate through wireless links (using 3G and WiFi) and constitute an example of a system with ubiquitous networked robots (UNR). The system integrates robots, sensors, communications, and mobile devices in a cooperative way, which means not only a physical interconnection between these elements, but also, for example, the development of novel intelligent methods of cooperation for task-oriented purposes.

The system is used, among other tasks, for person guiding by robots. This scenario consists of having a robot guiding a person towards a given destination. This person is to be detected and then must be

L. Merino (✉)
School of Engineering, Pablo de Olavide University,
41013, Seville, Spain
e-mail: lmercab@upo.es

A. Gilbert · R. Bowden · J. Illingworth
Centre for Vision Speech and Signal Processing,
University of Surrey, Guildford, GU2 7XH, UK

J. Capitán
Institute for Systems and Robotics,
Instituto Superior Tecnico, Lisbon, Portugal

A. Ollero
School of Engineering, University of Seville, Seville, Spain

A. Ollero
Centre for Advanced Aerospace Technology,
Parque Tecnológico y Aeronáutico de Andalucía,
C. Wilbur y Orville Wright 17-19-21,
41309, La Rinconada, Spain

tracked continuously to allow the robot to accomplish its task. Real scenarios involve dynamic environments and varying conditions. The robustness and reliability of autonomous perception in these scenarios are critical. In most cases, a single autonomous entity (i.e., a robot or a static surveillance camera) is not able to acquire all the information required for the application because of the characteristic of the particular task, i.e., loss of visibility. Thus, the cooperation among robots and between robots and heterogeneous sensors embedded in the environment through information fusion is relevant. The set of fixed cameras can obtain global views of the scene; however, as they are static, they cannot react to non-covered zones and illumination changes such as shadows can affect the system. Robots carry local cameras and can move to suitable positions, reacting to the changing conditions. However, their field of view is limited and they can lose the person they are tracking. Wireless devices can also help to localize the people, estimating their positions by measuring the signal strength from different static receivers, but the resolution obtained is usually low, and depends on the density of anchored receivers.

This paper presents the perception system developed in URUS for the UNR (see Fig. 1). The objective was to create a cooperative perception system, in which the different elements of the UNR system collaborate to obtain more precise information for the task assigned. In order to cope with scalability, a decentralized data fusion algorithm is employed, in which only local estimations and local communications are used. The main novelty of this work is the fusion of the various elements into a single system. Furthermore, some aspects of the work have been only demonstrated in simplified settings. Here, we extend the techniques and apply the approaches to a real outdoors application, using the fusion to overcome the limitations of any single approach. This paper builds on the preliminary work of the authors [15]; however, it has been extensively enhanced in terms of description and evaluation. The different techniques employed are described more thoroughly, including new parts of the system and the approach is thoroughly evaluated with extensive complex outdoor tests.

The next section presents related work. After an overview of the full system in Section 3, the paper will present the individual input sensor algorithms. Thus, Section 4 firstly describes the process to extract information from a set of fixed cameras. In Section 5 the main ideas of person tracking from onboard robot cameras are presented, and Section 6 describes the use of the signal strength from wireless sensors for tracking. Finally, the results of the tracking from all sensors are used to infer the position of the person in a global coordinate system through a data fusion process. This system is described in Section 7. The paper ends showing results obtained during the experiments of the URUS project in an urban scenario involving 22 fixed cameras, a WSN of 30 wireless Mica2 nodes, and several robots.

## 2 Related work

### 2.1 Tracking with camera networks

There has been many attempts to track people and other moving objects inter camera. The early tracking algorithms [4, 7, 27, 33] require both camera calibration

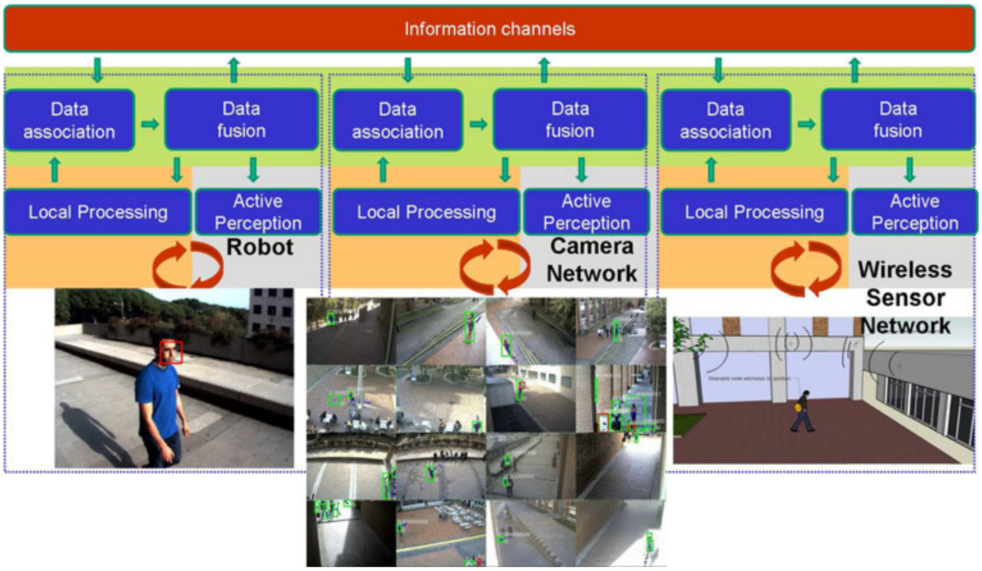


**Fig. 1** Overview of the URUS perception system. Information from robots, fixed cameras, and a WSN is considered

and overlapping fields of view to compute the handover of objects of interest between cameras. Others [22, 28] can work with nonoverlapping cameras but still require calibration. The use of nonoverlapping cameras is more realistic and reduces hardware limitations, though, as the handover of objects between cameras cannot be explicitly observed, reasoning must be used. Probabilistic approaches have been therefore proposed; [22] presents an approach to track cars on a highway modelling appearance and transition times as Gaussian distributions, though this was within a relatively controlled environment. Through the use of supervised off-line learning period, it has been possible to model the camera topology and path probabilities of objects [8, 48]. However, often probabilistic solutions for nonoverlapping cameras are used under restrictive assumptions or within limited applications. Illumination changes between cameras can be a challenge; therefore, a number of approaches [16, 23, 39] adjust the color appearance of cameras to improve performance.

Approaches have been proposed that do not require a priori correspondences to be explicitly stated [9, 25, 26]; instead they use the observed motion over time to establish reappearance periods. Nevertheless, batch processing was initially performed on the data. Therefore, if the environment changes significantly, the system must be “rebooted” and correspondences relearned. The use of local invariant features has also seen an increase in popularity over the last few years [11, 17].

Within this paper, the fixed camera tracking builds on the limited work of [14]. The camera transition relationships were incrementally learned, to model both the color variations and posterior probability distributions of the spatiotemporal transition links between cameras.

### 2.2 Radio-signal based tracking

There is an increasing interest in systems that use the signal strength received by wireless devices for localization purposes. Many systems are devoted to the localization of static devices by using the signal received from a small set of very well-localized static devices (called beacons) [40], or the signal received by a well-localized mobile node, usually on board a robot [3].

There is also a work devoted to the tracking of mobile nodes by using radio signals, which is the problem of estimating the position of a mobile node from the signal received by a set of static devices whose position are known. A tutorial on the main issues and approaches for the problem is presented in [20]. Many algorithms use, beside signal strength, additional information to obtain range estimates or even direction of arrival estimates. For instance, [37] considers the use of particle filters for tracking a mobile node using time of arrival, difference of time of arrival, and power measurements, presenting results in simulation. The works [29, 30] use the Doppler shift of interference signals to estimate the velocity and position of mobile nodes. These approaches require the precise synchronization of the emission of signals. In our approach, only signal strength is used through a calibrated model for radio propagation. Particle filters are used in [34, 44] for localization in indoor scenarios. Here, a similar approach is used but outdoor urban scenarios are considered. Moreover, the previous works require a full calibration of a signal map model. Here, a simple model of radio propagation is calibrated, and map information is used just in the prediction phase of the filter. Also, the tracking is benefited from the data fusion with other sensor networks. It is worth to mention that there are approaches in which the signal strength model is learned [12, 21].

### 2.3 Robot person tracking

Tracking from mobile platforms like robots in outdoor scenarios is a hard problem affected by clutter, illumination changes in the case of vision approaches, occlusions, etc. Most of the approaches combine people detection and people tracking modules for this task. The people detection module tries to obtain person hypotheses analyzing the sensor data and is usually computationally demanding. Many classification techniques are used for this task, like boosting [32], SVM [35], etc. The tracking module is usually a feature tracking algorithm applied to the initial hypothesis given by the detection module, which can be run at a higher rate than the detection algorithm, like CamShift [2]. In most cases, both modules support each other; so when the tracker is lost, new hypotheses from the detector can be used. More complex combinations, including what is called cognitive feedback, are also considered [10, 13].

In the work presented here, a combination of state of the art algorithms for detection [49] and tracking [2] from a single robot camera is used. This algorithm works relatively well, although outdoor scenarios pose difficulties to it. The key issue in the paper is to show how the combination of the local information obtained by the robot with the information received from other elements of the UNR system can improve the results.

### 2.4 Bayesian decentralized data fusion

Fusion of data gathered from a network of heterogeneous sensors is a highly relevant problem in robotics that has been widely addressed in the literature. Most of those works are based on Bayesian approaches, where the sensors are modeled like uncertain sources. However, there are alternative methods for dealing with uncertainties.

Possibility theory, which is built on the arithmetic of fuzzy sets, has been used for uncertain reasoning. For instance, in [5], possibility theory is used for cooperative localization and ball position estimation within the framework of Robocup. The decentralization of possibility-based systems is not clear though. Moreover, fuzzy techniques are mainly suitable for control systems, in which there are small sets of rules and no chains of inferences. In consensus theory [38, 41], the idea is to reach agreements among different sources, so it can also be used for sensor fusion. These techniques are typically used to achieve collective coordinated dynamics in multi-agent systems. They are, however, less adequate for applications in which the state evolves with time (e.g., tracking). For typical consensus algorithms, convergence to a same value among the sources is proved, but it cannot be assured that this value is the correct one [41].

In multi-sensor data fusion, Bayesian approaches provide a sound mathematical framework and allow for a better modelling when the uncertain sources are complex. Morever, in some cases (like the one proposed in this paper), they can be decentralized in an efficient manner. Even though fusing all the information in a central node is simpler, decentralized systems are scalable and moSre robust under communication failures since only local information and communication are used.

The main issues and problems with decentralized information fusion in Bayesian settings can be traced back to the work of [18], where the information filter (IF, dual of the Kalman Filter (KF)) is used as the main tool for data fusion for process plant monitoring. The works [18, 36, 46] demonstrate that, for the case of static states (for instance, in mapping applications, when estimating the location of a set of static objects), the decentralized implementation of the IF allows a local estimation that is the same as the one obtained by a centralized IF with access to all the information (provided that sufficient information is exchanged). In the case of dynamic states, for instance in tracking applications (like the one considered here), it was noticed in [1, 42] that if only an estimation about the last state is exchanged between the decentralized nodes, information will be missed with respect to a centralized node. The problem is due to the fact that there are some information not taken into account when performing the prediction steps in each fusing node.

The idea of the Channel filters in order to fuse the information in a consistent manner (non-overconfident) is considered in the works [19, 31]. These works require a fixed topology between nodes with no loops. Other options are conservative fusion rules that achieve a consistent estimation without the need for Channel filters when no assumptions can be made about the network topology, like the covariance intersection algorithm [24]. Moreover, [47] presents the Covariance union method, which tries to deal with disagreement in a Gaussian decentralized fusion setup.

In this paper, a decentralized information filter over the state trajectory is proposed as the main algorithm for scalable data fusion. It will be seen that the exact centralized estimation can be recovered due to the use of delayed states. In previous works like [31], this fusion was not possible for dynamic states without missing some information. Besides, the filter proposed deals with these issues and communication delays in an efficient manner.

## 3 UNR system overview

The ubiquitous networked robots system developed in the URUS project consists of a team of mobile robots, equipped with cameras and other sensors for localization, navigation, and perception; a fixed camera network for environment perception (see Fig. 2); and a wireless sensor network that uses the signal strength of the received messages from a mobile device to determine the position of a person carrying it. An architecture for urban robots networked with the environment has been developed [43]. This architecture provides a decisional layer and communication capabilities. The robots can switch between WiFi and 3G to communicate with the control station, other robots, and the camera and sensors networks. This paper is focused on the perception part of the system.

From the perception point of view, the information obtained by the fixed camera network or the wireless sensor network can be shared with the local information from each robot to improve the perception. That way, each robot obtains a better picture of the world than it would do alone. In this case, the tracks on the image plane obtained by the camera network will be fused with the information from the other systems (robots and WSN) to improve the tracking of the person being guided. Thus, it is possible to cope with occlusions and obtain better tracking capabilities since information of different modalities is employed; and cope with non-covered zones since the robots can move to cover these zones.

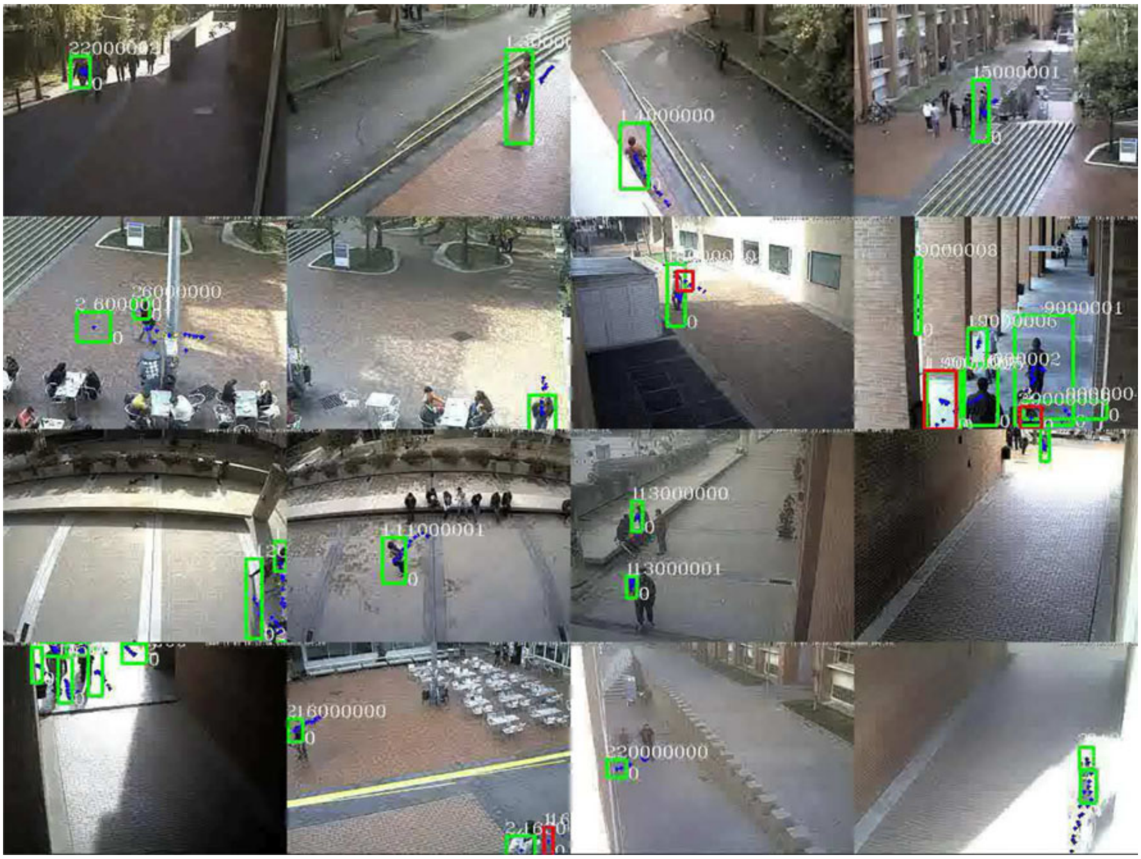
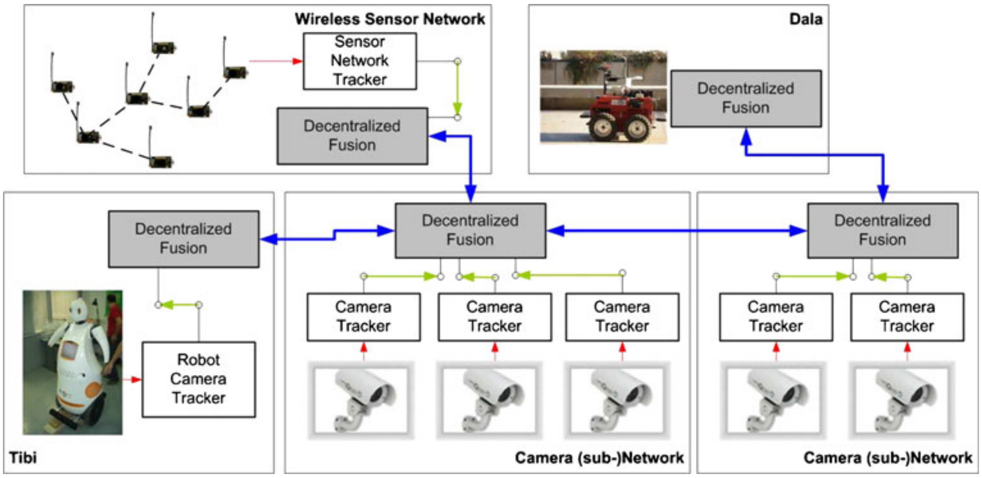


**Fig. 2** 16 of the 22 cameras within the experiment system. The cameras can provide overall information about the complete scenario

The system consists of a set of fusion nodes which implements a decentralized data fusion algorithm. Each fusion node only employs local information (data from local sensors; for instance, a camera subnet, or the sensors on board the robot) to obtain a local estimation of the variables of interest (in this case, the position of the person being tracked). Then, these nodes share their local estimations among themselves if they are within communication range. As the nodes only use local data and communications, the system is scalable. Also, as each node accumulates information from its local sensors, temporal communication failures can be tackled without losing information.

Figure 3 shows a block description of the system; for the camera network, each fusion node considers information from a small subset of cameras, which are processed in a distributed way, with a separate tracker obtaining estimations from each camera. The WSN

**Fig. 3** A block description of the URUS perception system. The different subsystems are integrated in a decentralized manner through a set of decentralized data fusion nodes. Locally, each system can process and integrate its data in a central way (like the WSN) or in a distributed way (like the camera network). Some systems can obtain information from the rest of the network even in the case they do not have local sensors



process messages from all the network in a gateway to localize the mobile node using the signal strength. Similarly, the onboard robot camera tracks nearby people. Then, the local estimations of the different nodes are fused in a decentralized way.

## 4 Fixed camera tracking

The fixed cameras cover a wide area of the experiment site and therefore, in many cases, they are the foundation for the fusion of the other sensors within the UNR; they are able to track objects of interest both on and across different cameras without explicit calibration periods.

Figure 4 gives a general overview of the processes for a single camera. Each camera is a self-contained node connected to others via a network, meaning that it can easily be distributed over multiple processors or machines. There are two inputs to the node, the camera image pixel values at (a) and also the estimated location of objects from previous frames at (f). The estimated location of previous frames allows the Kalman filter to use data from other sources to overcome occlusion that will occur with the foreground objects. There is a single output, *(e)*, which contains the location in camera coordinates of all the detected and tracked objects in the frame.

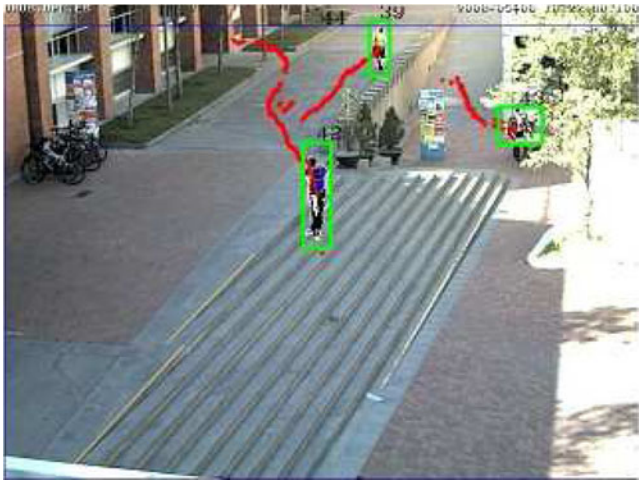


**Fig. 5** The track paths of three objects over time

To detect moving objects within an image, the static background is modelled in a similar fashion to that originally presented by [25]. The foreground objects are identified from the background mask through connected component analysis. This provides a bounding box centered over each object. A path track of each object over time is created using correlation between frames. A Kalman Filter is used to provide temporal correspondence between the detected foreground objects inter frame. A histogram is used as an object descriptor as it is spatially invariant, and through quantization, a degree of invariance to illumination can be achieved. Each object is then given a unique label for identification. Figure 5 shows an example of tracking multiple moving objects on a single camera at the experimental site.

When the object of interest enters a new camera, the transfer of the object's label to the new camera is a

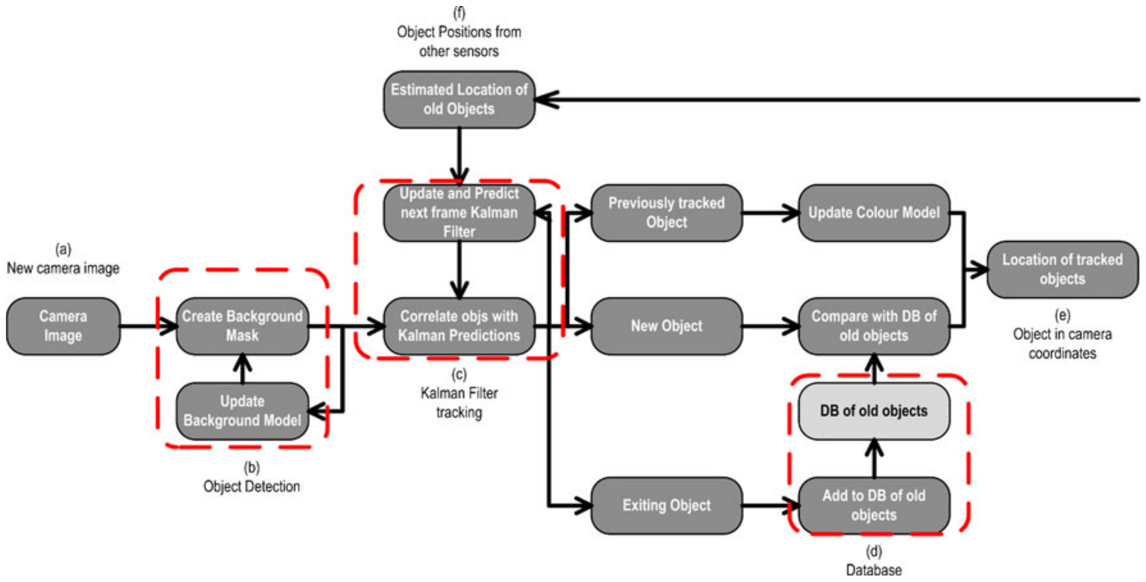

(f)
Object Positions from other sensors
Estimated Location of old Objects
Update and Predict next frame Kalman Filter
Previously tracked Object
Update Colour Model
(a)
New camera image
Location of tracked objects
Camera Image
Create Background Mask
Correlate objs with Kalman Predictions
New Object
Compare with DB of old objects
(e)
Object in camera coordinates
Update Background Model
(c)
Kalman Filter tracking
DB of old objects
(b)
Object Detection
Exiting Object
Add to DB of old objects
(d)
Database

**Fig. 4** System overview of tracking objects using a single camera

challenge, as the cameras have no overlapping fields of view, making many traditional image plane calibration techniques impossible. In addition, the large number of cameras means traditional time consuming calibration is infeasible. Therefore, the approach needs to learn the relationships between the cameras automatically. This is achieved by way of two individually weak cues, modelling the color, and movement of objects inter camera. These two weak cues are then fused to allow the technique to determine if objects have been previously tracked on another camera or are new object instances. By incrementally learning the cues over time, the accuracy is able to increase without any supervised input.

### 4.1 Forming temporal movement links inter camera

We make use of the key assumption that given time, objects (such as people) will follow similar routes inter camera and that the repetition of the routes will form marked and consistent trends in the overall data. These temporal transition links are used to link regions of the cameras together, producing a probabilistic distribution of object movements between cameras.

Initially, the system is divided so that each camera is defined as a single region. This coarse detail allows immediate operation and tracking of people, unlike the approaches of [9, 25] that require batch processing. While incoming data is stored to allow refinement and subdivision of the entry and exit regions using people tracked inter camera, all newly detected objects are compared to previously tracked objects within a set time window. The color similarity is calculated and used to increment a probability distribution with respect to the reappearance period as shown in Fig. 6. Regularly, the noise floor level is measured for each link, if the maximum peak of the distribution is found to exceed the noise floor level, this indicates a possible correlation between the two regions. Figure 6 shows the probability distribution for two regions with a distinct link at around 13 s.

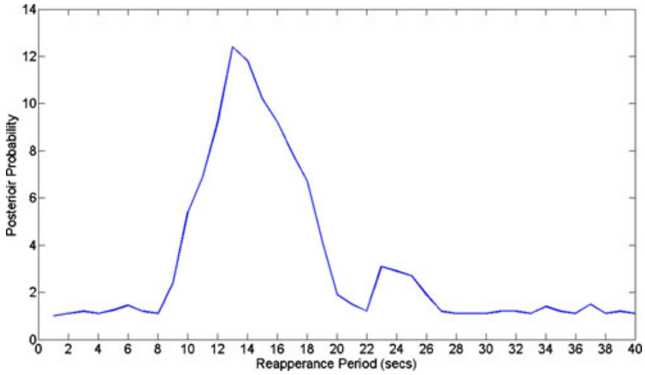


**Fig. 6** An example of a probability distribution, $f^{x|y}$ showing a distinct link between two regions $x$ and $y$

When a link is found between two regions, they can be subdivided to create four new equal-sized subregions, this aims to increase the detail level of the entry and exit areas. The previous data is then reused and incorporated with future evidence to form links in the newly subdivided regions.

It is likely that many of the subdivided regions will not form coherent links; therefore, if a link between two regions has no data in it, it is removed to minimize the number of links maintained. In addition, if a region is found to have no links to any other region, the region is also removed. This policy of removing unused and invalid regions improves system scalability. As the process proceeds, the regions start to visually represent entry and exit points of the cameras.

Figure 7 shows the entry and exit regions of two cameras in a experimental site. Even though regions cover initially the whole image, over time they start to represent the main entry and exit area of the cameras. The regions are continuously subdivided if a link to another region is found. However, when the regions are subdivided a number of times, it is likely that many neighboring regions contain similar links to other neighboring regions. Therefore, the correlation of neighboring region's link distributions is examined using Bhattacharyya coefficient. If the distributions from two neighboring regions are found to be highly correlated, the regions' spatial areas are combined to form a single region to further increase the overall number of samples.

### 4.2 Modelling color variations

The color quantization descriptor used to form temporal reappearance links in the previous section assumes a similar color response between cameras. However, this is seldom the case, especially on outdoor environments. Therefore, a color calibration of the cameras is proposed that can be learned incrementally and simultaneously with the temporal relationships discussed in the section above. The idea of using a color transformation matrix to calibrate the cameras has been proposed before; however, the experiments are often inside and limited [14, 23, 39]. we propose to use the color transformation matrix to calibrate the multiple outdoor cameras. The people tracked inter camera are used as the calibration objects, and a transformation matrix is formed incrementally to model the color changes between specific cameras.

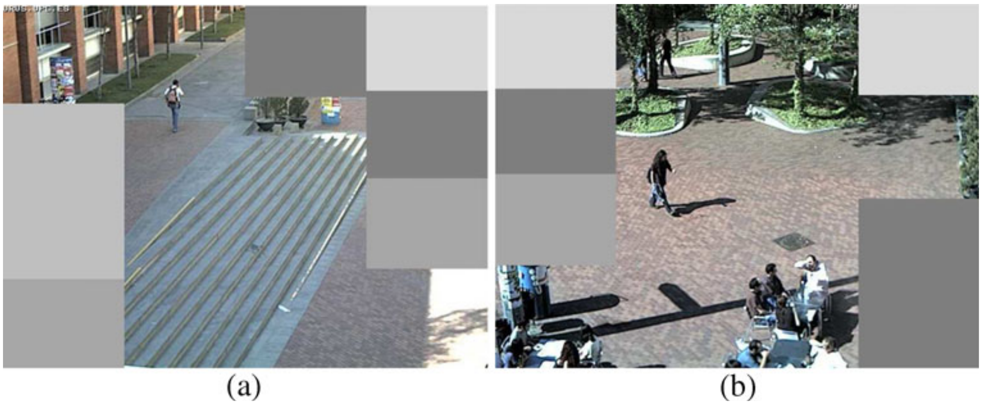


**Fig. 7** Entry and exit regions of two of the cameras, the darker the region the greater the quantity of track entry and exits

Between each set of cameras, a color transformation matrix is formed. Initially, this is an identity matrix. This assumes the ideal case of a uniform prior of color variation between cameras. When a person is tracked inter camera and is identified as the same object, the difference between the two color descriptors is modelled by a transform matrix. The matrix is calculated by computing the transformation that maps the person's descriptor from the previous camera to the person's current descriptor. This transformation is computed via SVD. The matrix is then averaged with the appropriate camera transformation matrix and repeated with other tracked people to gradually build a color transformation between cameras. This method will introduce small errors; however, it is in keeping with the incremental theme of the work. This is especially important when the environment is outdoors, as the lighting between cameras is constantly changing slightly, and this approach allows the method to continually update and adapt to the color changes between cameras over time.

With the weak cues learned, the reappearance probability of an object is then used to weight the observation likelihood obtained through color similarity to obtain a posterior probability of a match. Tracking objects is then achieved by maximizing the posterior probability within a set time window.

## 5 Robot camera tracking

The robots carry onboard cameras that are used for person guiding. These cameras can be used to obtain local estimations on the position of the person to be guided. A combination of state-of-the-art algorithms for person detection and tracking is used.

The person detection algorithm applied to the image is the one of [49]. This detection module is launched when the robot is requested to guide a person and it is close to the location where the person is waiting. Once the person is detected, it is tracked using an algorithm based on the CamShift technique [2]. While the algorithm is able to handle temporal occlusions (see Fig. 8), due to changes in illumination, the changing field of view of the camera when the robot moves, or even the person going out of the field of view, the tracking system is not sufficient to maintain the track on the person continuously. Therefore, the results from the tracking and the detection applications are combined, so that the robot employs the person detector whenever the tracker is lost to recover the track. The algorithm determines that the person is lost, employing some heuristics, like the track going out to the limits of the image or size restrictions on the blob. As a result, the robots can obtain estimations of the pose of the person on the image plane.

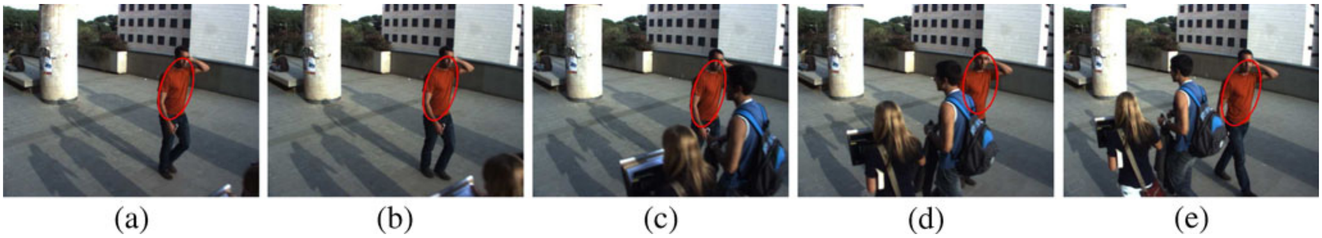


**Fig. 8** By using the local people tracking and detection module, the robot can obtain estimations of the person on the image plane. The employed tracking algorithm is able to handle temporal occlusions

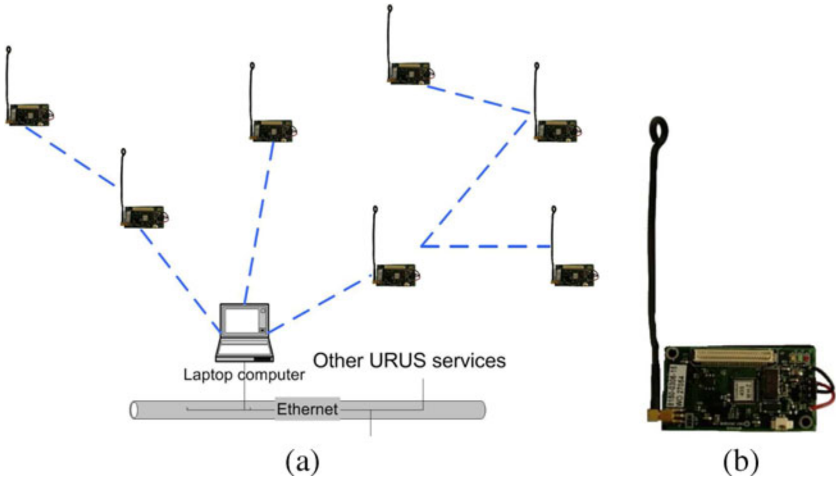


**Fig. 9** *Left*: WSN architecture: the nodes can establish an ad hoc network to relay information to a gateway. This information (messages indicating the power received from a mobile node) is processed in the gateway to obtain an estimation of the position of the mobile node. *Right*: WSN Mica2 node

More complex algorithms like [13] could be used and included into the system. However, it is out of the scope of the paper to develop a robust person tracking system based on a single camera onboard the robot. Nevertheless, the claim is that in general, a system based only on local information will not be robust enough to be able to guide one person through the whole scenario. Moreover, information from one camera alone is not sufficient to estimate the full 3D position of the person. The following sections will show how the combination of the local camera information and the information from the other subsystems (camera network and WSN) can overcome these problems.

## 6 Wireless sensor network tracking

Another element considered in the UNR system is a network of wireless Mica2 sensor nodes. These Mica2 nodes are able to sense different quantities, like pressure, temperature, humidity, etc. Moreover, they have wireless communication devices, and are able to form networks and relay the information they gather to a gateway (see Fig. 9).

In addition, the signal strength received by the set of static nodes (received signal strength indicator, RSSI) can be used to infer the position of a mobile object or a person carrying one of the nodes (the emitter). In the application considered here, the user that wants to be guided carries one of the nodes.

The algorithm to estimate and track the node position is based on particle filtering. In the particle filter, the current belief about the position of the mobile node is represented by a set of particles $\{\mathbf{x}_t^{(i)}\}$, which represent hypotheses about the current position of the person that carries the node (see Fig. 10).

In each iteration of the filter, kinematic models of the motion of the person and map information are used to predict the future position of the particles. The likelihood of these particles is updated when new messages are received from the static network. The technique is summarized in Algorithm 1, where $\mathbf{z}_t^j$ is the

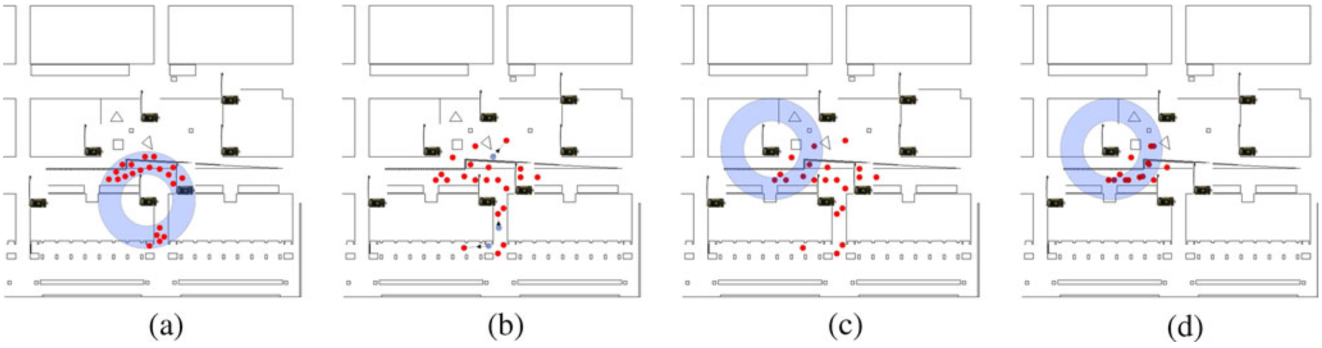


**Fig. 10** Particles (*red*) are used to represent person hypotheses. **a** The filter is initiated when the first message is received by sampling uniformly from a spherical annulus around the receiver. **b** Particles are predicted at each iteration; map information is also taken into account. **c** New measurements are used to weight the particles. **d** Re-sampling maintains the number of particles bounded

```
Algorithm 1 ${\mathbf{x}_t^{(i)}, \omega_t^{(i)}; i = 1, \ldots, L} \leftarrow$ Particle_filter(${\mathbf{x}_{t-1}^{(i)} \omega_{t-1}^{(i)}; i = 1, \ldots, L}, \mathbf{z}_t^j = {\mathbf{x}^j, RSSI_t^j}$)
1: for $i = 1$ to $L$ do
2:     $\mathbf{x}_t^{(i)} \leftarrow$ sample_kinematic_model ($\mathbf{x}_{t-1}^{(i)}$)
3: end for
4: if Message from network $\mathbf{z}_t^j$ then
5:     for $i = 1$ to $L$ do
6:         Compute $d_t^{(i)} = \|\mathbf{x}_t^{(i)} - \mathbf{x}^j\|$
7:         Determine $\mu(d_t^{(i)})$ and $\sigma(d_t^{(i)})$
8:         Update weight $\omega_t^{(i)} = p(RSSI_t^j|\mathbf{x}_t^{(i)})\omega_{t-1}^{(i)}$ with $p(RSSI|\mathbf{x}_t^{(i)}) = \mathcal{N}(\mu(d_t^{(i)}), \sigma(d_t^{(i)}))$
9:     end for
10: end if
11: Normalize weights ${\omega_t^{(i)}}, i = 1, \ldots, L$
12: Compute $N_{\text{eff}} = \frac{1}{\sum_{i=1}^{L}(\omega_t^{(i)})^2}$
13: if $N_{\text{eff}} < N_{th}$ then
14:     Re-sample with replacement $L$ particles from ${\mathbf{x}_t^{(i)}, \omega_t^{(i)}; i = 1, \ldots, L}$, according to the weights $\omega_t^{(i)}$
15: end if
```

measurement provided by each static node $j$, consisting of its position $\mathbf{x}^j$ and the strength $RSSI_t^j$ of the received signal from the mobile node. Following subsections describe the main steps in this algorithm.

### 6.1 Prior prediction and importance functions

As a prior, the filter is initialized with the first message received from the mobile node, considering a uniform distribution on a spherical annulus around the receiver. The map of the scenario is taken into account when sampling from this prior (see Fig. 10a), considering that the person is not inside any building.

Each time step, the position of the particles are predicted from their previous position (line 2 of Algorithm 1). No further information is assumed, and similar to [44], the prediction function employed is a Brownian motion model, in which new particles are sampled from a Gaussian distribution centered at the previous particle position [45] (Fig. 10b). However, this model also considers map information to discard infeasible motions (like going through walls).

### 6.2 The likelihood function

The likelihood function $p(RSSI_t|\mathbf{x}_t)$ plays a very important role in the estimation process since each time a message is received, this likelihood is used to update the particles weights (lines 5 to 9, Fig. 10c). The likelihood models the correlation that exists between the distance that separate two nodes and the $RSSI$ value. Figure 11 shows experimental data on the RSSI values for given distances. It can be seen that the correlation between RSSI and distance decreases with the distance between the two nodes, transmitter and receiver [3]. This is mainly caused by radio-frequency effects such as radio reflection, multipath or antenna polarization.

The model used here considers that the conditional density $p(RSSI_t^j|\mathbf{x}_t)$ can be approximated as a Gaussian distribution for a given distance $d_t^j = \|\mathbf{x}_t - \mathbf{x}^j\|$ between the mobile node and static node $j$, as follows:

$$RSSI_t^j = \mu(d_t^j) + \mathcal{N}(0, \sigma(d_t^j)) \quad (1)$$

From the experimental data, it can be seen that this model can represent adequately the relations for distances below 15 m. At the same time, this function allows an efficient evaluation within the particle filter. Please notice that the functions $\mu(d_t^j)$ and $\sigma(d_t^j)$ are themselves non-linear functions of the distance (which itself is a non-linear function of the state), so the full model is non-linear. These functions are estimated during a calibration procedure (Fig. 11 shows also the estimated functions for the data set).

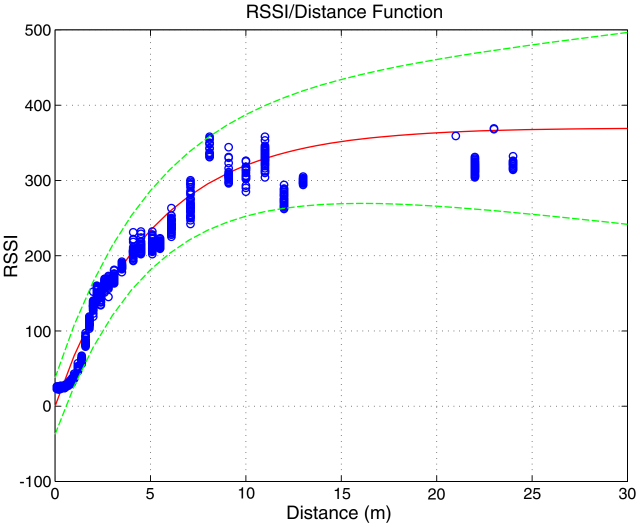


**Fig. 11** RSSI-distance functions, $\mu(d_k)$ and $\sigma(d_k)$. These functions relate the distance between two nodes and the RSSI received in mean and std. deviation. It has been experimentally computed using a large set of RSSI/distance couples. The RSSI representation is the one used in the Mica2 nodes, 0 is the maximum signal strength and 375 the minimum. *Dots* represent a sub-set of the experimental set of data. *Solid line* represents estimated mean $\mu(d_k)$. *Dashed lines* represent standard deviation confidence interval based on $\sigma(d_k)$

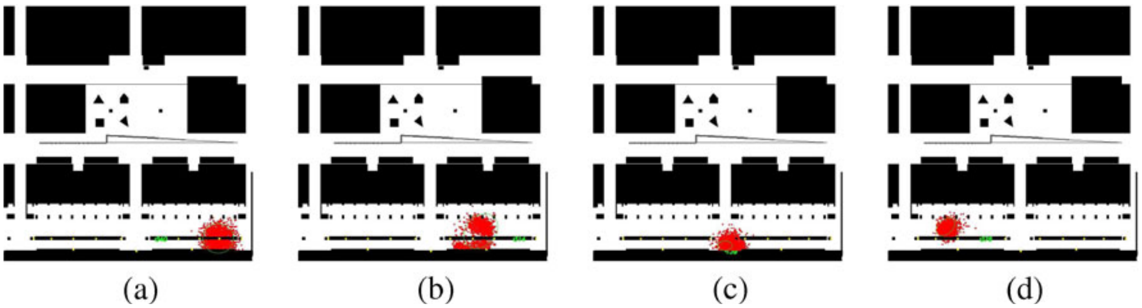


**Fig. 12** A sequence of the 500 particles employed in the filter for an experiment. *Red points* represent the particles. *Yellow points* represent the static nodes, being the *green* one the emitter at each frame

### 6.3 Filter evolution

Although Section 8 will show additional results, Fig. 12 presents the evolution of the particles for a particular tracking experiment performed at the experimental site. Five hundred particles are employed, and the algorithm runs at more than 1 Hz. Figure 13 shows the estimated position of the mobile node carried by a person estimated by the WSN. It is compared to that of the guiding robot, which is some meters ahead. In this experiment, this means that the robot has a lower X and higher Y coordinates.

When the filter converges to a Gaussian distribution, the estimated mean and covariance can be fed to the decentralized fusion system that will be explained in the next section. The convergence is determined by analyzing the Kullback–Leibler divergence between the particle distribution and a Gaussian distribution with the same mean and covariance as the particles.

## 7 Decentralized data fusion for person tracking

Using the trackers described above, the camera network, the robots, and the WSN are able to obtain local estimations of the position of the people on the image plane or in a 3D coordinate system. This information, characterized as Gaussian distributions (mean and covariance matrix), can be fused in order to obtain a more accurate estimation of the 3D position of the person.

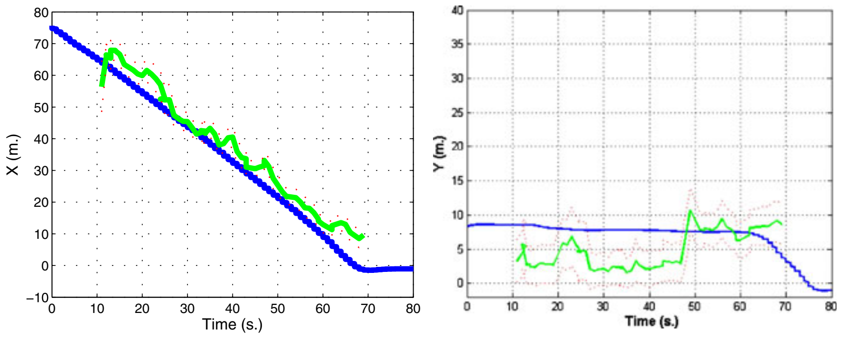
As commented in Section 3, the idea is to implement a decentralized fusion approach, in which each fusion node only employs local information (data from local sensors; for instance, a camera subnet, or the sensors on board the robot) and then shares its estimation with neighboring nodes (see Fig. 3, right). Thus, scalability and robustness are improved and bandwidth requirements alleviated.

The question is how to integrate measurements from all the sources and deal with communication delays without losing any information with respect to a centralized solution. A novel Bayesian filter that keeps track of delayed states is proposed to recover exactly the centralized estimation. Moreover, it is shown how in the case of Gaussian distributions, these state trajectories can be maintained in an efficient manner.

### 7.1 Delayed-state information filter

The IF, which corresponds to the dual implementation of the KF, is a suitable approach for decentralized multi-robot estimation. Whereas the KF represents the distribution using its first $\boldsymbol{\mu}$ and second $\boldsymbol{\Sigma}$ order moments, the IF employs the so-called "canonical representation". The fundamental elements are the information vector $\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and the information matrix

**Fig. 13** Estimated position of the person by the WSN (*green*) and position of the guiding robot (*blue*) estimated by using its navigation modules. *Dashed lines* represent sigma intervals



$\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$. Prediction and updating equations for the (standard) IF can also be derived from the standard KF [6]. In the case of non-linear prediction or measurement models, first order linearization leads to the extended information filter (EIF). Even though the prediction stage becomes more complex for the IF, the update stage is simpler. Hence, the use of the IF for multi-robot applications is justified by the additive nature of its updating steps. This simplifies a lot the filter when there is a single prediction step but multiple updates for the different data sources.

Formally, in a delayed-state information filter, the belief over the full trajectory of the state up to the current time step $t$, is denoted by $\mathbf{\Omega}^t$ and $\boldsymbol{\xi}^t$. Thus, delayed states are also considered instead of just estimating the current state $\mathbf{x}_t$. Let us consider the system:

$$\mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \boldsymbol{\nu}_t \tag{2}$$

$$\mathbf{z}_t = \mathbf{g}_t(\mathbf{x}_t) + \boldsymbol{\varepsilon}_t \tag{3}$$

where $\mathbf{x}_t$ is the person's position and velocity at time $t$, $\mathbf{z}_t$ represents the estimations obtained by the camera network, robots or WSN at time $t$, $\mathbf{A}_t$ is the prediction model, $\mathbf{g}_t$ the corresponding measurement model, and $\boldsymbol{\nu}_t$ and $\boldsymbol{\varepsilon}_t$ are Gaussian noises. Knowing the information matrix and vector for the person trajectory up to time $t-1$, $\mathbf{\Omega}^{t-1}$ and $\boldsymbol{\xi}^{t-1}$, the estimation of this trajectory can be updated up to time $t$ incorporating the local measurements. This is done according to Algorithm 2 [6], where $\mathbf{M}_t = \nabla \mathbf{g}_t(\bar{\boldsymbol{\mu}}_t)$, $\mathbf{R}_t$ is the covariance of the additive noise for the prediction model (2) and $\mathbf{S}_t$ is the covariance matrix of the noise in the measurement (3). **Add_M** adds a block row and a block column of zeros to the previous information matrix and **Add_Row** adds a block row of zeros to the previous information vector. For the movement of the person, a linear model $(\mathbf{A}_t, \mathbf{R}_t)$ is used like in [6]. Here, an initial estimation of the person position is also assumed in order to initialize the filter.

In general, for a state trajectory of $n$ steps, the storage required would be $O(n^2)$. However, in the IF filter proposed, when the trajectory grows, the matrix structure is block tridiagonal and symmetric at any time, what leads to a storage $O(n)$. Moreover, the computational complexity of the algorithm itself is $O(1)$, as the prediction and updating computations at each time instant only involve the previous block. These advantages allow the proposed approach to deal with delayed states more efficiently than a normal KF would do.

**Algorithm 2** $(\boldsymbol{\xi}^t, \mathbf{\Omega}^t)$ ←Delayed State Information Filter$(\boldsymbol{\xi}^{t-1}, \mathbf{\Omega}^{t-1}, \mathbf{z}_t)$

```
1: $\bar{\mathbf{\Omega}}^t = \textbf{Add\_M}(\mathbf{\Omega}^{t-1}) + \begin{pmatrix} \begin{pmatrix} \mathbf{I} \\ -\mathbf{A}_t^T \end{pmatrix} \mathbf{R}_t^{-1} (\mathbf{I} - \mathbf{A}_t) & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$
2: $\bar{\boldsymbol{\xi}}^t = \textbf{Add\_Row}(\boldsymbol{\xi}^{t-1})$
3: $\mathbf{\Omega}^t = \bar{\mathbf{\Omega}}^t + \begin{pmatrix} \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{M}_t & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$
4: $\boldsymbol{\xi}^t = \bar{\boldsymbol{\xi}}_t + \begin{pmatrix} \mathbf{M}_t^T \mathbf{S}_t^{-1} (\mathbf{z}_t - \mathbf{g}_t(\bar{\boldsymbol{\mu}}_t) + \mathbf{M}_t \bar{\boldsymbol{\mu}}_t) \\ \mathbf{0} \end{pmatrix}$
```

If several measurements from different sensors $\mathbf{z}_t^i$ arrive at time $t$, and these measurements are conditionally independent given the state $\mathbf{x}_t$, with measurement functions $\mathbf{g}_t^i$ and noise $\mathbf{S}_t^i$, then step 3 of Algorithm 2 becomes (equivalently for step 4):

$$\mathbf{\Omega}^t = \bar{\mathbf{\Omega}}^t + \sum_i \begin{pmatrix} \mathbf{M}_t^{iT} (\mathbf{S}_t^i)^{-1} \mathbf{M}_t^i & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{4}$$

$$\boldsymbol{\xi}^t = \bar{\boldsymbol{\xi}}^t + \sum_i \begin{pmatrix} \mathbf{M}_t^{iT} (\mathbf{S}_t^i)^{-1} (\mathbf{z}_t^i - \mathbf{g}_t^i(\bar{\boldsymbol{\mu}}_t) + \mathbf{M}_t^i \bar{\boldsymbol{\mu}}_t) \\ \mathbf{0} \end{pmatrix} \tag{5}$$

Moreover, maintaining delayed states allows the filter to incorporation of delayed and a sequent data by adding their contribution to the corresponding elements of the information vector and matrix of the state trajectory.

### *7.1.1 Measurement functions*

Regarding the measurements functions $\mathbf{g}_t(\mathbf{x}_t)$ considered in the system, the following considerations must be noted:

1. The camera network, as described in Section 4, obtains measurements on the image plane. The position of the tracked objects can be transformed into the world coordinate system through a set of homographies that are obtained beforehand through a calibration process (although these homographies are not used by the camera trackers themselves).
2. The robot obtains observations on the image plane, and then $\mathbf{g}_t(\mathbf{x}_t)$ is in this case the camera pin-hole model.
3. The WSN provides 3D estimations on the position of the person in the world coordinate system (see Section 6).

## 7.2 Decentralized information filter

The main interest of the IF is that it can be easily decentralized. In a decentralized approach, each node $i$ of the network employs only its local data $\mathbf{z}_t^i$ to obtain a local estimation of the person trajectory (given by $\boldsymbol{\xi}^{i,t}$ and $\mathbf{\Omega}^{i,t}$) and then shares its belief with its neighbors. The received information $\boldsymbol{\xi}^{j,t}$ and $\mathbf{\Omega}^{j,t}$ from other nodes is locally fused in order to improve the local perception of the world. Ideally, the decentralized fusion rule should produce the same result locally as that obtained by a central node employing a centralized filter.

If a node $i$ runs Algorithm 2 using only its local information, the one step equations are:

$$\boldsymbol{\Omega}^{i,t} = \underbrace{\begin{pmatrix} \mathbf{0} & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0} & \multicolumn{2}{c}{\multirow{2}{*}{$\boldsymbol{\Omega}^{i,t-1}$}} \\ \mathbf{0} & & \end{pmatrix}}_{prior} + \underbrace{\begin{pmatrix} \mathbf{R}_t^{-1} & -\mathbf{R}_t^{-1}\mathbf{A}_t & \mathbf{0}^T \\ -\mathbf{A}_t^T\mathbf{R}_t^{-1} & \mathbf{A}_t^T\mathbf{R}_t^{-1}\mathbf{A}_t & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}}_{prediction}$$

$$+ \underbrace{\begin{pmatrix} \mathbf{M}_t^{iT}(\mathbf{S}_t^i)^{-1}\mathbf{M}_t^i & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}}_{update} \tag{6}$$

$$\boldsymbol{\xi}^{i,t} = \underbrace{\begin{pmatrix} \mathbf{0} \\ \boldsymbol{\xi}^{i,t-1} \end{pmatrix}}_{prior} + \underbrace{\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}}_{prediction}$$

$$+ \underbrace{\begin{pmatrix} \mathbf{M}_t^{iT}(\mathbf{S}_t^i)^{-1}(\mathbf{z}_t^i - \mathbf{g}_t^i(\bar{\boldsymbol{\mu}}_t) + \mathbf{M}_t^i\bar{\boldsymbol{\mu}}_t) \\ \mathbf{0} \end{pmatrix}}_{update} \tag{7}$$

When this node $i$ is within communication range of another node $j$, they can share their beliefs, represented by their information vectors $\boldsymbol{\xi}^{i,t}$ and $\boldsymbol{\xi}^{j,t}$, and matrices $\boldsymbol{\Omega}^{i,t}$ and $\boldsymbol{\Omega}^{j,t}$. Assume, without loss of generality, that nodes $i$ and $j$ have common priors $\boldsymbol{\Omega}^{j,t-1} = \boldsymbol{\Omega}^{i,t-1}$ and $\boldsymbol{\xi}^{i,t-1} = \boldsymbol{\xi}^{j,t-1}$ (for instance, due to previous communications). This node $j$ will apply the same Eqs. 6 and 7, but with its own measurement $\mathbf{z}_t^j$ and covariance $\mathbf{S}_t^j$. Then, it can be seen that the the next fusion rule, proposed by the authors [6]:

$$\boldsymbol{\Omega}^{i,t} \leftarrow \boldsymbol{\Omega}^{i,t} + \boldsymbol{\Omega}^{j,t} - \boldsymbol{\Omega}^{ij,t} \tag{8}$$

$$\boldsymbol{\xi}^{i,t} \leftarrow \boldsymbol{\xi}^{i,t} + \boldsymbol{\xi}^{j,t} - \boldsymbol{\xi}^{ij,t} \tag{9}$$

where

$$\boldsymbol{\Omega}^{ij,t} = \underbrace{\begin{pmatrix} \mathbf{0} & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0} & \multicolumn{2}{c}{\multirow{2}{*}{$\boldsymbol{\Omega}^{i,t-1}$}} \\ \mathbf{0} & & \end{pmatrix}}_{prior} + \underbrace{\begin{pmatrix} \mathbf{R}_t^{-1} & -\mathbf{R}_t^{-1}\mathbf{A}_t & \mathbf{0}^T \\ -\mathbf{A}_t^T\mathbf{R}_t^{-1} & \mathbf{A}_t^T\mathbf{R}_t^{-1}\mathbf{A}_t & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}}_{prediction} \tag{10}$$

$$\boldsymbol{\xi}^{ij,t} = \underbrace{\begin{pmatrix} \mathbf{0} \\ \boldsymbol{\xi}^{i,t-1} \end{pmatrix}}_{prior} + \underbrace{\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}}_{prediction} \tag{11}$$

allows node $i$ to recover locally the same estimation as the one with a central entity receiving the information from $i$ and $j$ would obtain. The equations mean that each node must sum up the information received from other nodes. The additional terms $\boldsymbol{\Omega}^{ij,t}$ and $\boldsymbol{\xi}^{ij,t}$ represent the common information between the nodes. This common information is due to previous communications between nodes (previous priors) and common prediction information, and should be removed to avoid double counting of information (i.e., rumor propagation). After eliminating this common information, the remaining information from $j$ is due to its local data (or data gathered from other nodes connected to $j$).

These fusion equations can be applied also to the general case in which the nodes exchange information at arbitrary instants. The bottom line is that each node computes part of the update that a central node would compute (which is a sum for all the information received, see Eqs. 4 and 5). The fusion rule allows a fusing node to recover the centralized solution if the common information is properly removed. As long as a tree-shaped logical topology in the perception system (no cycles or duplicated paths of information) is assumed, this common information can be maintained by a separated EIF, a so-called channel filter [31].

It is important to remark that, using these fusion equations and considering trajectories (delayed states), the local filter can obtain an estimation that is equal to that obtained by a centralized system [6], which is not the case when using just the last state [36, 42] (unless all measurements arrive in order and no measurement is lost). Another advantage of using delayed states is that the belief states can be fused asynchronously without missing information. Each node in the UNR system can accumulate evidence and send it whenever it is possible. Also, as commented before, out-of-order and delayed measurements can be incorporated in the filter. However, as the state grows over time, the size of the message needed to communicate its belief also does. For the normal operation of the system, only the state trajectory over a time interval is needed, so these belief trajectories can be bounded by marginalizing out old states (which is a cheap operation due to the block diagonal nature of the information matrix). Note that the trajectories should be longer than the maximum expected delay in the network in order not to miss any measurement information.

Finally, when no assumptions about the network topology can be made (e.g., due to the existence of mobile robots, possible losses of communication links, etc.), another option to remove the common information is to employ a conservative fusion rule, which ensures that the system does not become overconfident even in presence of duplicated information. For the case of the IF, there is an analytic solution for this, given by the covariance intersection algorithm of [24]. Therefore, the conservative rule to combine the local belief of a robot $i$ with that received from another robot $j$ is given by:

$$\mathbf{\Omega}^{i,t} \leftarrow \omega\mathbf{\Omega}^{i,t} + (1-\omega)\mathbf{\Omega}^{j,t} \tag{12}$$

$$\boldsymbol{\xi}^{i,t} \leftarrow \omega\boldsymbol{\xi}^{i,t} + (1-\omega)\boldsymbol{\xi}^{j,t} \tag{13}$$

for $\omega \in [0\ 1]$. It can be demonstrated that the estimation is consistent (in the sense that no overconfident estimations are done) for any $\omega$. The value of $\omega$ can be selected following some criteria, such as maximizing the obtained determinant of $\mathbf{\Omega}^{i,t}$ (minimizing the entropy of the final distribution). Here, the option chosen is to use $\omega$ as a fixed weight that determines the system confidence in its own estimation and the neighbor's ones.

Although employing the CI formula avoids the need to maintain an estimation of the common information transmitted to the neighbor systems, as these fusion rules are conservative, some information is lost with respect to the purely centralized case.

### 7.3 Data association

Each fusion node of the system should be able to associate its local observations to the current tracks. In the case of the camera network, this is done by combining the inter camera information and geometric information. As commented in Section 4, the system is able to handle inter camera tracking without calibration, using as weak cues reappearance probabilities and color information. Therefore, the system uses this information for data association. As this scheme may fail, the non-associated observations are also passed through a data association procedure based on the Mahalanobis distance, using the estimated 3D position obtained from the homographies.

The data association in the case of the WSN node is straightforward, as the messages from the WSN are tagged with an ID. The image tracker in the case of the robot maintains the identity of the tracked persons while they are on the image plane. The Mahalanobis distance is also used to associate new measurements with previous tracks.

Moreover, the decentralized nodes should be able to associate the received tracks to the local tracks. For this track-to-track fusion, the Mahalanobis distance is used again.

## 8 Experimental results

The techniques described above have been tested during the experimental sessions of the URUS European project. The experiments were carried out outdoors at the UPC Campus North site, in Barcelona, Spain. The experimental setup consisted of 22 fixed color cameras with mostly nonoverlapping fields of view. Moreover, a network of 30 Mica2 nodes was deployed in the campus. A set of different robots were involved in the experiments. Figure 14 shows a moment of these experiments and the final deployment of sensors. Usually, the camera network is used as the initial point for the person-guiding task. First, some partial results will be described concerning the camera network. Then, the results obtained with the full system will be shown.

### 8.1 Fixed camera sensor results

A series of partial experiments concerning just tracking of people using the fixed camera sensors were performed. The experiments were conducted in October, with cloudy and sunny weather. The cameras are wall mounted around 5–10 m high. There are large gaps

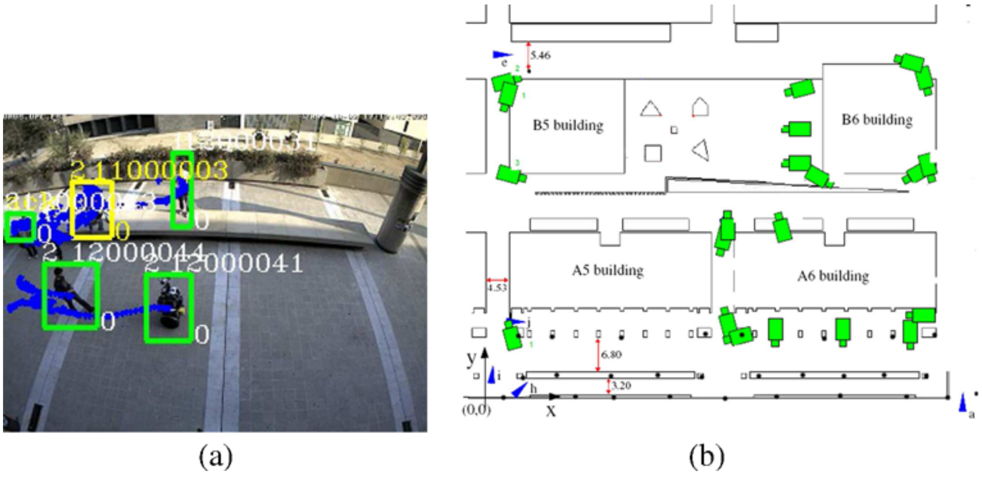


**Fig. 14** **a** One of the robots (on the image marked as 12000041) guiding a person (12000044). **b** Scenario. The dimension is 100 by 100 m, approximately. Cameras in *green* and Mica2 nodes as *black dots*

between many of the cameras of up to 30 s. The 22 time-synchronized IP video feeds are fed into three quad-core PC with real-time person tracking. There is no calibration of the camera environment with no a priori information provided. Over time, additional information is incorporated into the system to learn links between regions on the cameras and improve tracking accuracy. The experimental data was accumulated from 9 am for 5 days, tracking a total of around 140,000 people in total or around 1,200 people per camera per day. Often, the same person was tracked on multiple cameras as they moved around the site; therefore, the number of unique people was around 500 per day per camera or 55,000 unique people over the complete system. Figure 15 shows resultant temporal likelihoods for a number of inter camera links at a single subdivision level. The evaluation of the tracking was performed using three unique sequences taken at different times of day on different days.

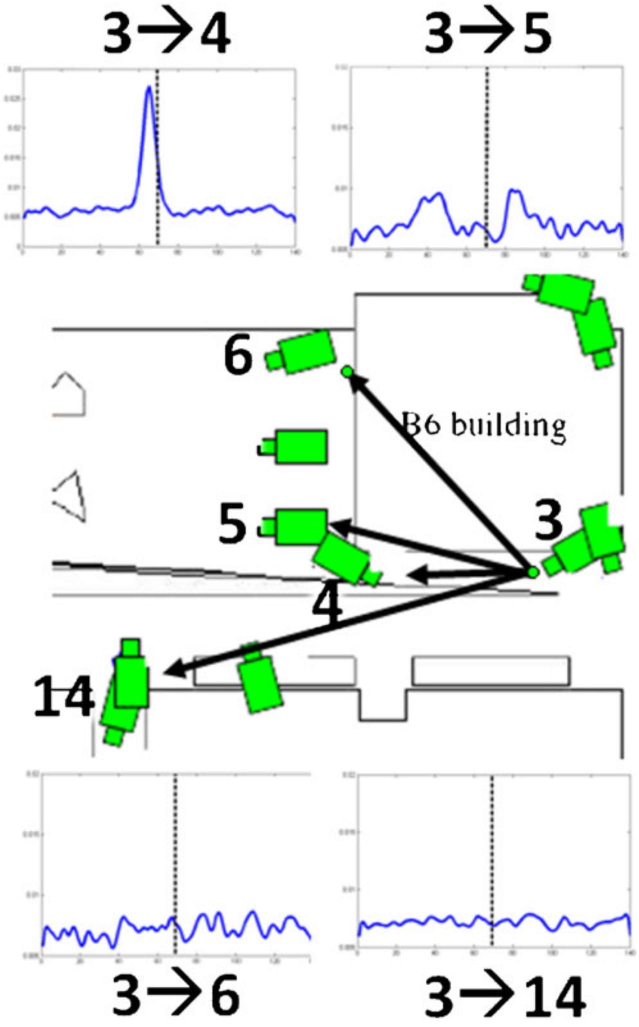


**Fig. 15** Inter camera temporal likelihoods

**Table 1** Details of the fixed camera test sequences

| Seq | Time of day | Length | Weather | Num of people |
|---|---|---|---|---|
| 1 | 11:00 | 80 min | Sunny | 1,200 |
| 2 | 12:00 | 180 min | Cloudy | 3,750 |
| 3 | 16:00 | 200 min | Sunny | 1,750 |

The black vertical line indicates a reappearance of 0 s. It can be seen that there are strong links between cameras 3 and 4 and between 3 and 5, whereas there are no visible links between 3 and 6 and between 3 and 14. This is due to the increased distance and people will rarely reappear on cameras 6 and 14 after they were tracked on camera 3. Table 1 shows the details of the three test sequences used to evaluate the approach. Table 2 shows the results of tracking people inter camera on the three sequences. A subdivisions of 0 means using no region link cues, only basic color correlation to match and track people inter camera. A subdivision of 1 is a single region per camera, i.e., the initial camera to camera linking, whereas a subdivision of 2 is where any suitable single camera regions are subdivided into 4 equally sized news regions as described in Section 4.1. For the test sequence, all people that occurred on multiple cameras were ground-truthed and a true positive occurred when a person was assigned the same ID that they were assigned in a previous region. A false positive indicates when a person was assigned an incorrect ID, and a false negative is when a person who has moved inter camera is given a new ID instead of the ID from their previous region.

The column for 0 subdivisions indicates performance without learning the temporal and color relationships between regions. It is generally poor because of the large color variations inter camera as well as shadow and lighting changes. The subdivision level 1 performs far better, with the additional detail of two subdivisions providing a further improvement. The reason for the greater performance on sequence 3 is due to the time

**Table 2** Fixed camera tracking of people

| Seq | Subdivisions | True positive (%) | False positive (%) | False negative (%) |
|---|---|---|---|---|
| 1 | 0 | 5 | 35 | 60 |
| | 1 | 50 | 26 | 24 |
| | 2 | 52 | 24 | 23 |
| 2 | 0 | 5 | 46 | 49 |
| | 1 | 47 | 22 | 31 |
| | 2 | 52 | 24 | 23 |
| 3 | 0 | 3 | 17 | 80 |
| | 1 | 58 | 21 | 21 |
| | 2 | 62 | 21 | 17 |

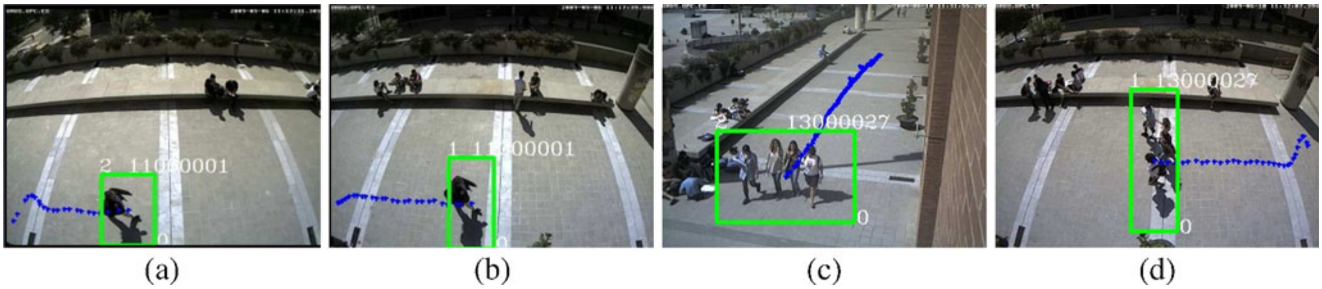


**Fig. 16** Cross camera tracking **a** Person 11000001 on camera 11; **b** Person 11000001 correctly identified on camera 12; **c** Person 13000027 on camera 13; **d** Person 13000027 correctly identified on camera 12

of day of the experiment. Being later in the day there was less simultaneous traffic on the system, this meant there were less possible correlation options for people tracked cross camera. Figure 16 gives example frames of tracking inter camera for two separate people. Figure 17 shows the estimated position of the person using only information from the camera network (the cameras are homography-calibrated, although this is not used by the intra and inter camera tracking algorithms).

### 8.2 Robot and WSN

In order to illustrate the benefits from the data fusion process, a first setup is presented here. This setup considers information from one camera on board the robot Romeo (four-wheel vehicle) and the WSN. The objective was to track one person cooperatively. In this case, just two nodes of the decentralized fusion are used: one on board the robot and one for the WSN. These nodes locally integrate information from a monocular camera (see Fig. 18) and from the signal strength-based estimations (Section 6, see Fig. 18a), respectively.

Figure 19 shows the X and Y estimations obtained by the robot alone and when the robot combines its information with the one provided by the WSN. For these outdoor and urban experiments, obtaining the real position of the person was not easy since the tests were run online and there were some areas without GPS coverage. Therefore, the person moved together with the robot, and the robot position was used as ground truth to check the estimation. The trajectory

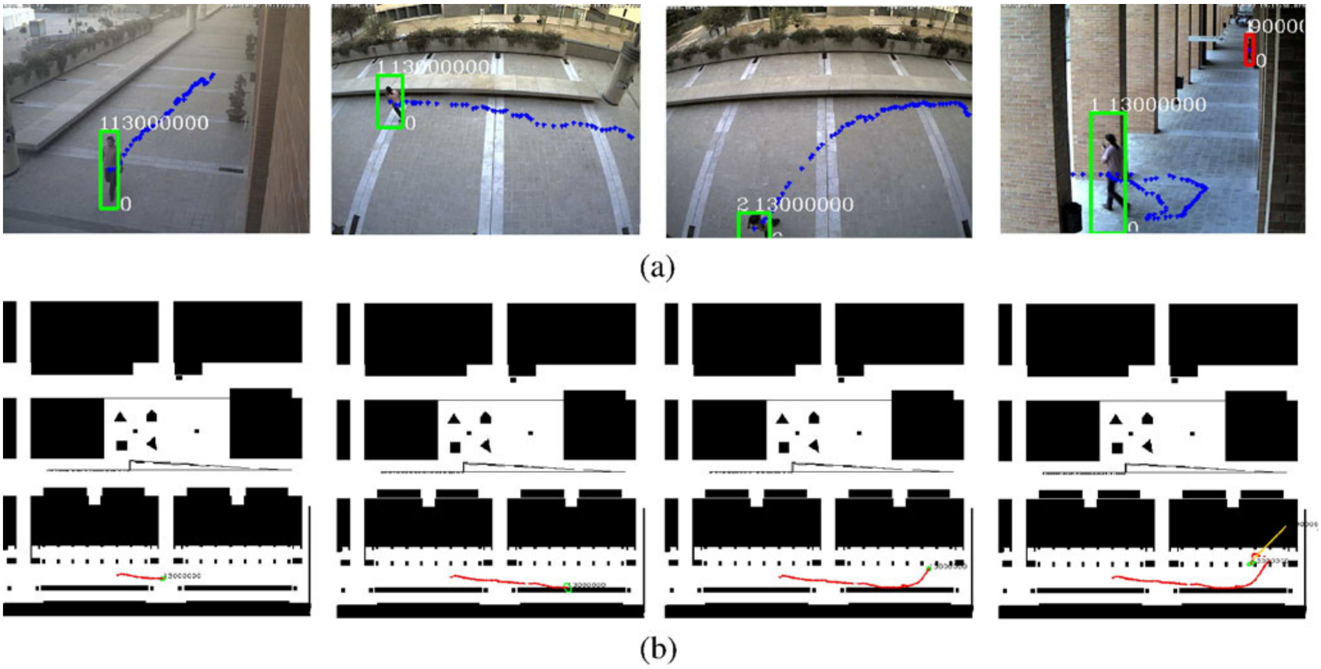


**Fig. 17** **a** Tracks on the image plane of 4 different cameras. The identity is correctly handed over the cameras using the weak cues described in Section 4. **b** Estimated position of the person on the UPC campus

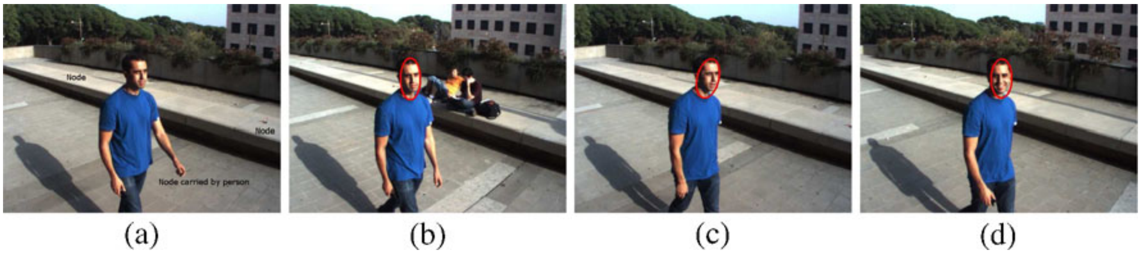


**Fig. 18** **a** The person is carrying a Mica2 node during the experiment. **b–d** The robot is able to obtain local observations on the image plane of the face of the person

of the robot was measured accurately by its navigation software (laser, map knowledge, GPS, etc.). The person is following behind the robot (see Fig. 18) (which in this trajectory means that the X coordinates of the person are larger than that of the robot) and some meters beside the robot (a lower Y coordinate).

Regarding the accuracy of the method, it can be seen that the error in the estimation (which is determined by the standard deviation) is enough for tracking people in an area of around 2,500 m$^2$. Moreover, the key point is that the fusion system improves the accuracy of independent sensors. Thus, it can be seen how the introduction of the WSN reduces the uncertainty; as we have a monocular camera, the uncertainty on the person position is quite big in both axes when the robot is alone. In this case, the initial position of the person is computed assuming a known height of the face.

### 8.3 UNR experiments with decentralized fusion

In this setup, one robot, the WSN of 30 nodes, and seven IP cameras are used. In the experiment, one person was following the robot, which was manually controlled. The setup of the perception system is a decentralized node on the robot, one for the WSN and two for the IP cameras, one integrating measurements from three cameras and the other from 4 cameras.

Figure 20 shows some examples of the tracks obtained by the onboard camera and the camera network. Along the trajectory, of more than 350 m, there were gaps in the camera coverage. Moreover, the robot lost track several times due to the changes in illumination. Finally, the WSN coverage was limited to a certain part of the campus.

Figure 21a shows the estimated position of the person with the full system running. The total length of the experiment is around 350 m and 5 min. The person is usually beside the robot (which means that the X or Y coordinates are the same) and the robot position is used as ground truth again. The system is able to maintain the estimation on the person position for the full trajectory. There are WSN coverage between 0 and 150 s, approximately. Figure 21b shows an interval of the trajectory. In this part, only WSN and robot

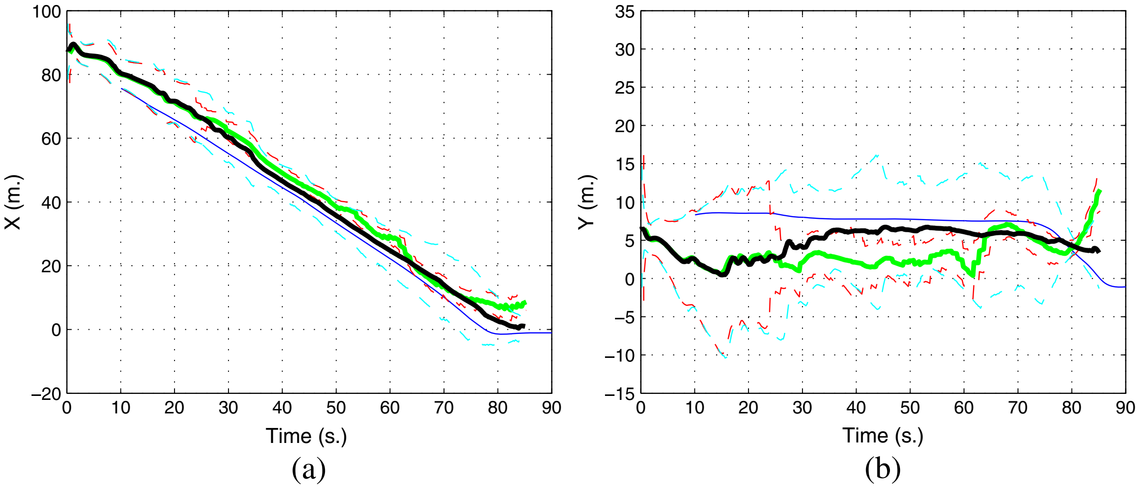
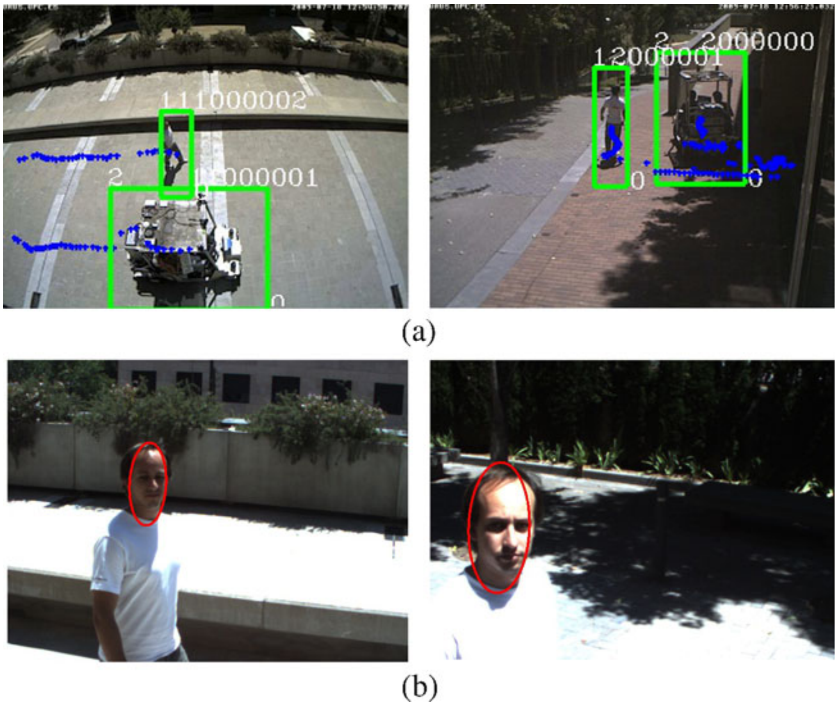


**Fig. 19** Tracking using one onboard camera and the WSN. *Black*: robot alone. *Green*: robot and WSN. *Dashed lines* are the sigma intervals and the *blue solid line* represents the robot trajectory

**Fig. 20** **a** Tracks obtained by the camera network. **b** Tracks obtained by the camera onboard Romeo

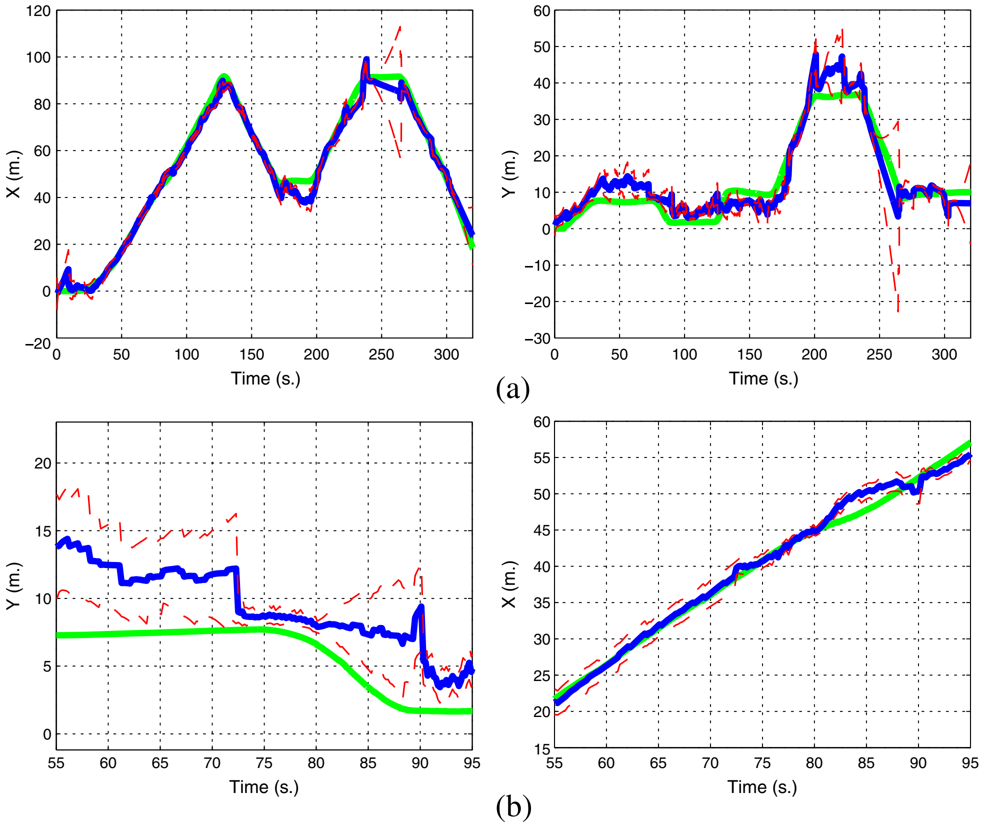


(a)

(b)

information are available. Sensor fusion is beneficial because, although the WSN measurements have lower precision, they bound the error from the monocular camera. At 75 s, the person enters under coverage of the camera network. This leads to a big reduction in uncertainty.

**Fig. 21** Estimated position of the person (*blue*) compared to the position of the robot (*green*). *Dashed lines* represent the standard deviation of the estimation. **a** Complete trajectory. **b** A section of the trajectory. The person is following the robot with the same X coordinate up to time 80 s. Then the robot changes orientation. The person is separated from the robot around 3–4 m



(a)

(b)

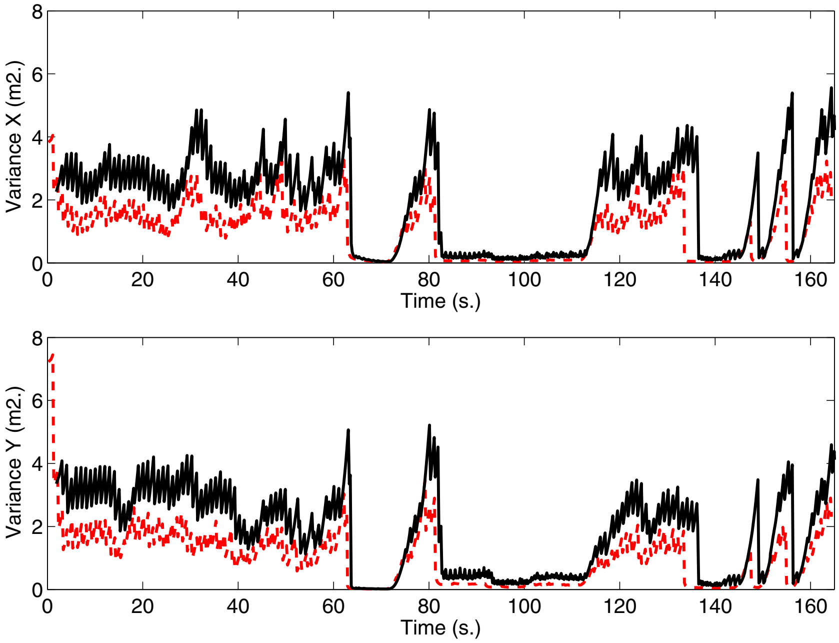


**Fig. 22** Estimated variance by a central node receiving all the information (*dashed* and *red*) compared to the estimation in a decentralized node (*solid* and *black*)

### *8.3.1 Other issues*

The same kind of experiment was repeated several times. The communication between the fusion nodes on board the robots and the fusion nodes related to the camera network and the WSN was done using WiFi and 3G. Software running on the robot was able to measure the quality of the WiFi link, and to switch to 3G whenever this quality dropped below a certain threshold.

The switching between communication networks created from time to time gaps of several seconds. Moreover, although 3G had more stable coverage in the scenario, it had lower bandwidth and higher latencies.The use of decentralized nodes allowed the system to cope with communication gaps, as in the meantime, the local nodes were accumulating information. When the communication links were recovered, the nodes exchanged their estimations. Moreover, as delayed states were considered, this delayed information (and also information delayed due to the latencies) could be fused in a correct way, and no information was lost.

Figure 22 compares the decentralized estimation with a centralized off-line implementation. In this experiment, four different cameras and the WSN were running. For the centralized implementation, all the information is received and fused in a single node (and no information is missed; it can be considered as a gold standard estimation); whereas for the decentralized case, three fusion nodes were used (two of them processing locally information from two cameras; the remaining one processing locally information from the WSN). The estimation obtained by one of the fusion nodes is shown in Fig. 22. It can be seen how, with some latencies depending on the conditions, the decentralized node obtains an estimation quite close to the centralized one, even though some information is lost due to the conservative fusion rule employed in this case (covariance intersection, see Section 7. Moreover, the decentralized estimation is consistent, in the sense that it never accumulates more information than the one obtained in the ideal centralized filter.

## 9 Conclusions

The combination of robots and ambient intelligence (like embedded sensors and camera networks) seems a clear trend in the near future. This paper has presented a decentralized system that aims to use multiple sensors to accurately track people within a surveillance context. The system makes extensive use of data fusion procedures to incorporate all the information available.

The algorithms are real-time and operate on realistic outdoor environments. The fixed camera tracking provides high accuracy by incrementally learning the color and temporal relationships between regions on nonoverlapping cameras. Moreover, the signal strength of mobile devices is employed to estimate the position of the person using particle filtering. The combination of all these information obtained by robots allows for accurate person tracking in more challenging situations.

The system has been tested in an urban scenario, considering a camera network of 20 cameras, a WSN of 30 nodes, and robots.

Very complex algorithms employing just one source of information are usually unable to cope with all the potential situations in these scenarios, affected by changes in illumination, clutter, and wide area coverage. The combination of complementary systems can be useful for this problem. Signal-based localization is less accurate than camera-based localization, but it is less affected by occlusions. Robots have usually narrower fields of view, but they can adapt to cover places occluded from the camera networks. However, camera networks can provide overall information on the scene though. Scalability is an issue in these systems, and thus decentralized algorithms are required. The system presented is a mixture between distributed or centralized subsystems that are linked through a decentralized data fusion scheme. The addition of new robots or sub-nets of cameras does not affect the rest of the perception system in terms of storage, as only local communication and processing are used.

Future developments include the integration of active sensing behaviors in the system. The WSN can be actively controlled to save energy, activating those nodes more useful for tracking. The robots can also move maximizing the possibility of maintaining the person in the field of view. In some applications, the robots can use paths that are more informative as they are more covered by static cameras. Entropy-based information gain algorithms and partially observable Markov decision processes will be considered for these tasks.

**Acknowledgements** This work is partially supported by URUS, Ubiquitous networking Robotics in Urban Settings, funded by the European Commission (EC) under FP6 with contract number FP6-EU-IST-045062. In addition, the authors would like to thank to the rest of the partners of the URUS project for their help and support. Luis Merino is also funded by the EC through the project FROG (FP7-288235). Jesus Capitan is also funded by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the PIDDAC Program funds and projects PEst-OE/EEI/LA0009/2011 and CMU-PT/SIA/0023/2009.

## References

1. Bailey T, Durrant-Whyte H (2007) Decentralised data fusion with delayed states for consistent inference in mobile ad hoc networks. Technical report, Australian Centre for Field Robotics, University of Sydney
2. Bradski GR (1998) Computer vision face tracking as a component of perceptual user interface. In: Proc. of workshop on applications of computer vision, pp 214–219
3. Caballero F, Merino L, Gil P, Maza I, Ollero A (2008) A probabilistic framework for entire wsn localization using a mobile robot. J Robot Auton Syst 56(10):798–806
4. Cai Q, Aggarwal JF (1998) Automatic tracking of human motion in indoor scenes across multiple synchronized video streams. In: Proc. of IEEE international conference on computer vision (ICCV'98)
5. Cánovas JP, LeBlanc K, Saffiotti A (2004) Robust multi-robot object localization using fuzzy logic. In: Proc. of the international robocup symposium
6. Capitán J, Merino L, Caballero F, Ollero A (2009) Delayed-state information filter for cooperative decentralized tracking. In: Proceedings of the international conference on robotics and automation, ICRA
7. Chang TH, Gong S, Ong E (2000) Tracking multiple people under occlusion using multiple cameras. In: Proc. of BMVA British machine vision conference (BMVC'00), pp 566–575
8. Dick A, Brooks M (2004) A stochastic approach to tracking objects across multiple cameras. In: Proc. of Australian conference on artificial intelligence, pp 160–170
9. Ellis TJ, Makris D, Black JK (2003) Learning a multi-camera topology. In: Proc. of joint IEEE workshop on visual surveillance and performance evaluation of tracking and surveillance (VS-PETS), pp 165–171
10. Ess A, Leibe B, Schindler K, Van Gool L (2008) A mobile vision system for robust multi-person tracking. In: IEEE conference on computer vision and pattern recognition (CVPR)
11. Farenzena M, Bazzani L, Perina M, Cristani A, Murino V (2010) Person re-identification by symmetry-driven accumulation of local features. In: Proc. of IEEE international conference on computer vision and pattern recognition (CVPR'10)
12. Ferris B, Hohnel D, Fox D (2006) Gaussian processes for signal strength-based location estimation. In: Proc. of robotics science and systems
13. Fintrop S, Königs A, Hoeller F, Schulz D (2009) Visual person tracking using a cognitive observation model. In: International conference on robotics and automation (ICRA), workshop in people detection and tracking
14. Gilbert A, Bowden R (2008) Incremental, scalable tracking of objects inter camera. Comput Vis Image Underst (CVIU) 3:43–58
15. Gilbert A, Capitán J, Bowden R, Merino L (2009) Accurate fusion of robot, camera and wireless sensors for surveillance applications. In: Proc. ninth IEEE international workshop on visual surveillance (ICCV09), Kyoto, Japan
16. Gilbert A, Illingworth J, Bowden R (2008) Scale invariant action recognition using compound features mined from dense spatiotemporal corners. In: Proc. of European conference on computer vision (ECCV'08), vol I, pp 222–233
17. Gray D, Tao H (2008) Viewpoint invariant pedestrain recognition with an ensemble of localized features. In: Proc. of European conference on computer vision (ECCV'08), pp 262–275
18. Grime S, Durrant-Whyte HF (1994) Data fusion in decentralized sensor networks. Control Eng Pract 2(5):849–863
19. Grocholsky B, Makarenko A, Kaupp T, Durrant-Whyte HF (2003) Scalable control of decentralised sensor platforms. In: Lecture notes in computer science, vol 2634. Springer
20. Gustafsson F, Gunnarsson F (2005) Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements IEEE Signal Process Mag 22 (4)41–53
21. Hollinger G, Djugash J, Singh S (2008) Tracking a moving target in cluttered environments with ranging radios.

In: IEEE int conf robot autom, pp 1430–1435. doi:10.1109/ROBOT.2008.4543403

22. Huang T, Russell S (1997) Object identification in a Bayesian context. In: Proc. of international joint conference on artificial intelligence (IJCAI-97), pp 1276–1283
23. Javed O, Shafique K, Rasheed Z, Shah M (2008) Modeling inter-camera space-time and appearance relationships for tracking across nonoverlapping views. Comput Vis Image Underst 109(2):146–162
24. Julier SJ, Uhlmann JK (1997) A non-divergent estimation algorithm in the presence of unknown correlations. Proc Am Control Conf 4:2369–2373
25. KaewTrakulPong P, Bowden R (2003) A real-time adaptive visual surveillance system for tracking low resolution colour targets in dynamically changing scenes. J Image Vis Comput 21(10):913–929
26. KaewTrakulPong P, Bowden R (2005) Towards automated wide area visual surveillance: tracking objects between spatially separated, uncalibrated views. IEE Proc Vis Image Signal Process 152(10):213–224
27. Kelly P, Katkere A, Kuramura D, Moezzi S, Chatterjee S (1995) An architecture for multiple perspective interactive video. In: Proc. of the 3rd ACE international conference on multimedia, pp 201–212
28. Kettnaker V, Zabih R (1999) Bayesian multi-camera surveillance. In: Proc. of international conference on computer vision and pattern recognition, pp 253–259
29. Kusý B, Ledeczi A, Koutsoukos X (2007) Tracking mobile nodes using RF Doppler shifts. In: Proceedings of SenSys, pp 29–42
30. Kusý B, Sallai J, Balogh G, Ledeczi A, Protopopescu V, Tolliver J, DeNap F, Parang M (2007) Radio intereferometric tracking mobile wireless nodes. In: Proceedings of MobySys, pp 139–151
31. Makarenko A, Brooks A, Williams S, Durrant-Whyte H, Grocholsky B (2004) A decentralized architecture for active sensor networks. In: Proceedings IEEE international conference on robotics and automation, ICRA, vol 2, pp 1097–1102
32. Martinez-Mozos O, Kurazume R, Hasegawa T (2010) Multi-part people detection using 2D range data. Int J Soc Rob 2:31–40
33. Morariu VI, Camps OI (2006) Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics. In: Proc. of IEEE international conference on computer vision and pattern recognition (CVPR'06), vol I, pp 545–552
34. Morelli C, Nicoli M, Rampa V, Spagnolini U, Alippi C (2006) Particle filters for RSS-based localization in wireless sensor networks: an experimental study. In: International conference on acoustics, speech and signal processing, vol 4, p IV
35. Navarro-Serment LE, Mertz C, Hebert M (2009) Pedestrian detection and tracking using three-dimensional ladar data. In: Proc. of the 7th int. conf. on field and service robotics
36. Nettleton E, Durrant-Whyte H, Sukkarieh S (2003) A robust architecture for decentralised data fusion. In: Proc. of the international conference on advanced robotics (ICAR)
37. Norlund PJ, Gustafsson F, Gunnarsson F (2002) Particle filters for positioning in wireless networks. In: Proceedings of EUSIPCO
38. Olfati-Saber R, Fax J, Murray R (2007) Consensus and cooperation in networked multi-agent systems. Proc IEEE 95(1):215–233
39. Prosser B, Gong S, Xiang T (2008) Multi-camera matching using bi-directional cumulative brightness transfer functions. In: BMVC'08, pp 164.1–64.10
40. Ramadurai V, Sichitiu ML (2003) Localization in wireless sensor networks: a probabilistic approach. In: Proceedings of the 2003 international conference on wireless networks (ICWN 2003), Las Vegas, NV, pp 275–281
41. Ren W, Beard R, Atkins E (2007) Information consensus in multivehicle cooperative control. IEEE Control Syst 27(2):71–82
42. Rosencrantz M, Gordon G, Thrun S (2003) Decentralized sensor fusion with distributed particle filters. In: Proc. conf. uncertainty in artificial intelligence
43. Sanfeliu A, Andrade-Cetto J, Barbosa M, Bowden R, Capitán J, Corominas A, Gilbert A, Illingworth J, Merino L, Mirats JM, Moreno P, Ollero A, Sequeira J, Spaan MTJ (2010) Decentralized sensor fusion for ubiquitous networking robotics in urban areas. Sensors 10(3):2274–2314
44. Seshadri V, Zaruba GV, Huber M (2005) A Bayesian sampling approach to in-door localization of wireless devices using received signal strength indication. In: International conference on pervasive computing and communications, pp 75–84
45. Stone LD, Corwin TL, Barlow CA (1999) Bayesian multiple target tracking. Artech House, Inc., Norwood
46. Sukkarieh S, Nettleton E, Kim J-H, Ridley M, Goktogan A, Durrant-Whyte H (2003) The ANSER project: data fusion across multiple uninhabited air vehicles. Int J Rob Res 22(7–8):505–539
47. Uhlmann JK (2003) Covariance consistency methods for fault-tolerant distributed data fusion. Inf Fusion 4(3):201–215
48. Ukita N (2007) Probabilistic-topological calibration of widely distributed camera networks. Mach Vis Appl 18(3):249–260
49. Viola P, Jones M (2004) Robust real-time face detection. Int J Comput Vis 57:137–154