Information Fusion 36 (2017) 296–312

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Efficient integration of RSSI for tracking using Wireless Camera Networks

A. De San Bernabe, J.R. Martinez-de Dios*, A. Ollero

Robotics, Vision and Control Group, Escuela Tecnica Superior de Ingenieros, Avenida de los descubrimientos s/n, Sevilla 41092, Spain

ARTICLE INFO

Article history:
Received 12 February 2016
Revised 3 October 2016
Accepted 1 November 2016
Available online 4 November 2016

Keywords:
Camera networks
RSSI
Wireless Sensor Networks

ABSTRACT

This paper proposes a scheme that efficiently exploits synergies between RSSI and camera measurements in cluster-based target tracking using Wireless Camera Networks (WCNs). The scheme is based on the combination of two main components: a training method that accurately trains RSSI-range models adapted to the conditions of the particular local environment; and a sensor activation/deactivation method that decides on the individual activation of sensors balancing the different information contributions and energy consumptions of camera and RSSI measurements involved in sensing. The scheme also includes a distributed Extended Information Filter that integrates all available measurements. The combination of these components originates self-regulated behaviors that drastically reduce power consumption and computational effort with no significant tracking degradation w.r.t. existing schemes based exclusively on cameras. Besides, it shows better robustness to target occlusions. The proposed scheme has been implemented and validated in real experiments.

1. Introduction

In the last few years intense research has been devoted to localization and tracking in indoor and GPS-denied environments. Although a high variety of sensors, techniques and approaches have been developed, it is still an important open research field. This paper deals with Wireless Camera Networks (WCNs), which can implement successful perception schemes typical of conventional camera networks with the flexibility and re-configurability of Wireless Sensor Networks (WSN).

A high variety of WCN-based tracking schemes have been reported. Most of them rely exclusively on camera measurements disregarding other types of sensors that are actually integrated in most wireless camera nodes. The greatest majority of wireless camera nodes can measure the Radio Signal Strength Indicator (RSSI) of incoming packets with negligible energy and computational consumption. If the targets are tagged with emitting nodes, each camera node can gather camera and RSSI measurements of the targets that are within its camera field of view and radio range. Fusion of bearing and range measurements – like RSSI – can originate interesting synergies, involving higher accuracy in the localization and tracking of the target. Also, RSSI can be used when the target is occluded in the images or when it is out of the cameras field of view. Despite these advantages, very few existing WCN-based tracking schemes integrate RSSI measurements. RSSI is highly affected by reflections, multi-path propagation and other interactions of the radio signal with the environment making RSSI-range models very dependent on the setting. Besides, the few techniques that integrate RSSI and camera measurements use RSSI only when the camera measurements are not available, without exploiting all the synergies between both types of measurements.

Our scheme assumes that targets are tagged with emitting nodes. This is the case in many tracking applications in which the smartphones and portable computing devices carried by people act as Wi-Fi emitting nodes and the Wi-Fi cameras deployed in the environment act as wireless camera nodes.

This paper proposes a cluster-based scheme that exploits the synergies between RSSI and camera measurements to improve resource consumption efficiency in target tracking. The scheme is based on the combination of two main components: a training method in which each camera node accurately calibrates its RSSI-range model to the particular environment and settings; and a sensor activation/deactivation method that balances the use of accurate but high energy-consuming camera measurements and of energy-efficient but inaccurate RSSI measurements. The interaction between these components originates an emerging self-regulated effect in which the camera measurements are used to train RSSI-range models and once calibrated, the sensor selection method tends to deactivate cameras due to their higher energy consump-

* Corresponding author.
E-mail addresses: adesanbernabe@us.es (A. De San Bernabe), jdedios@us.es (J.R. Martinez-de Dios), aollero@us.es (A. Ollero).

http://dx.doi.org/10.1016/j.inffus.2016.11.001
1566-2535/

tion. The scheme also includes a distributed Extended Information Filter (EIF) that efficiently integrates all available measurements. The proposed scheme obtains similar accuracies to methods based only on cameras but with drastic improvements in energy consumption and computational burden. Its performance and robustness has been validated in the *CONET Integrated Testbed* [1].

Although the components adopted in our scheme have value by themselves, it is the combination and interaction between them what originates the aforementioned emerging effect that is key for its superiority over existing schemes. According to the authors the combination of these components is the main novelty of this paper.

This paper focuses on efficiency in resource consumption. It is not the objective of this paper to provide a complete WCN-based tracking solution. In cluster-based tracking each cluster is used to track a target. Our scheme is not constrained to the single-cluster case, it is also valid for multi-cluster scenarios. However, in order to deal with multi-target tracking, specific mechanisms should be added to address issues such as measurement association, track crossing, cluster merging/splitting or specific management of cluster heads. Many of the techniques to solve these issues are research topics by themselves and their description often requires full specific papers, such as [2–4]. For brevity and simplicity in the description of our scheme, these mechanisms have been left out of the scope of our paper since we preferred to focus on the core of our research.

This paper is inspired by some ideas we sketched in [5]. The main improvements over [5] are:

- a new sensor activation/deactivation method that allows the independent activation of each sensor of each node providing higher flexibility and power consumption reductions. The method considers transmission errors and uses the trace of information matrix, which enables distributed computation and improves scalability;
- presentation of the distributed implementation of the scheme;
- proof-of-concept examples that illustrate the performance of the scheme and the synergies between the components;
- new and more detailed experimental performance and robustness evaluation and comparison with existing methods;
- extension and more detailed related work. Also, the paper has been re-structured and all sections have been completed and rewritten for clarity.

The paper is organized as follows. Related work is in the next section. The general description of the proposed scheme is in Section 3. The distributed EIF, the RSSI-range model training and the sensor activation components are described respectively in Sections 4–6. The experimental validation as well as the performance and robustness analyses are presented in Section 7. Conclusions is the final section.

2. Related work

Localization and tracking in indoor and GPS-denied environments have attracted high interest in the recent years. A wide variety of techniques using different technologies and sensors have been reported. They can be coarsely classified into: infrared positioning systems, ultrasound systems, radio frequency systems, magnetic systems and vision-based systems. In infrared-based systems, the light emitted from badges or tags is used to localize and track objects, see e.g. [6,7]. They are very accurate and there are some successful products in the market such as the VICON system, among others. Most ultrasound systems measure the range to the tracked tags and use them to estimate their location, see e.g. [8,9]. A huge variety of radio frequency localization systems have been developed ranging from RADAR-based schemes [10] to Ultra Wide Band (UWB) systems [11]. Magnetic systems have long tradition in localization and tracking [12]. They have high accuracy and do not suffer from line-of-sight constraints. Vision-based networks have been widely used for indoor localization and tracking since decades. Their main advantage is that targets do not need to be tagged.

A high variety of localization and tracking techniques for Wireless Sensor Networks in ubiquitous computing environments have been reported. Many different types of measurements have been used including RSSI [13], Time of Flight (ToF) [14], Time Difference of Arrival (TDoA) [15] and Angle-of-Arrival (AoA) [16], among many others. These measurements have been combined using many different information fusion techniques. In the last years the integration of low energy CMOS vision chips with programmable image processing capabilities in ubiquitous computing systems originated the so-called WCNs [17].

WCN combine the sensing capacity of camera networks with the ease of deployment and scalability of WSNs. A good number of WCN-based tracking systems based only on camera measurements have been reported [18–23]. Almost all existing techniques integrate only cameras measurements without benefiting from other sensors that are actually integrated in most camera node models, such as RSSI.

RSSI is by far the most widely extended measurement used in localization and tracking in Wireless Sensor Networks. RSSI can be measured by most node models using many different wireless communication technologies like Bluetooth, RFID (Radio-frequency Identification), ZigBee, Wi-Fi or WiMAX (Worldwide Interoperability for Microwave Access). A high number of RSSI-based localization systems have been developed. *Range-based* methods, such as multilateration [24] or least squares [25], among many others, use RSSI measurements to estimate the distance to anchor nodes. Reflections and other interactions with the environment make RSSI-range models very dependent on the setting, making them unpredictable. *Range-free* methods, such as ROC-RSSI [26] or APIT [27], avoid these drawbacks by using geometrical considerations instead of RSSI-range models. However, their accuracies are usually poorer. Another approach is to learn RSSI characteristics from the environment. Fingerprinting methods, see e.g. [28], compare measurements with a previously obtained RSSI map. They require accurate RSSI maps of the environment, which should be re-calculated if the setting changes. In [29] each node uses the location of surrounding nodes to train its RSSI-range model. However, the method cannot capture the interactions with the local environment surrounding the target.

Bearing and range measurements have interesting synergies. However, the number of WCNs that use range measurements is very low. Work [30] combines camera and ultrasound measurements. Range is estimated using TDoA sensors. The method combines three different steps: least squares minimization, Kalman filtering and outlier rejection. Of course, this work assumes that nodes are equipped with ultrasound sensors.

RSSI can be measured by most camera nodes with no hardware or software costs. However, few works combine camera and RSSI. Miyaki et al. proposed to estimate target location individually using cameras and RSSI and then, to integrate both estimates using a sensor fusion method [31] or a Particle Filter (PF) [32]. Work [31] combined both results by switching between the algorithms: camera-based tracking is used in areas covered by cameras and RSSI, in areas with no camera coverage. In [32] both measurements are combined continuously: the PF relies more on cameras measurements when the target is present in the images or on RSSI when the object is occluded. Also, a framework based on PFs for integrating camera and RSSI measurements is presented in [33]. The PF uses camera or RSSI measurements depending on their availability. When the object is not visible, an observation model

based on RSSI is used to update the prediction of the object position. All the above methods use RSSI only in case of lack of camera measurements. The scheme proposed in this paper exploit the synergies between cameras and RSSI mainly when both measurements are available.

Cluster-based tracking schemes need to address issues such as measurement integration, node inclusion/exclusion and cluster head selection. Our scheme includes components specifically designed to address these issues. The main components of our scheme are the RSSI training module and the sensor activation/deactivation. Below we summarize the main related work in both components.

Cluster-based tracking schemes include node inclusion/exclusion methods in order to cope with moving targets. These methods critically affect energy consumption: nodes out of the cluster can be turned-off saving energy. Some works, such as [34], use the distance to the target as the main criterion for node inclusion. However, proximity does not imply that the camera is capable to acquire information about the target. Others include the node in the cluster when the target is estimated to enter the camera field of view and excludes the node when the camera stops detecting the target [18]. This criterion keeps many cameras unnecessarily active unless the deployment has been optimized [35].

Other methods, such as [36,37], attempt to select the set of sensors that minimize the Mean Square Error (MSE) of the localization and tracking estimations. The applicability of this scheme is limited to small networks because it incurs in a high computational burden. Other typical criteria for sensor selection are dynamic information driven schemes [38,39], where the objective is to maximize the information gain based on the dynamic information gathered; and entropy-based schemes [40,41], where the selection schemes aim to minimize the entropy of the estimation. However most of those schemes are centralized and not scalable which make them impractical in many scenarios. The trace of the information matrix has been used for sensor selection in [42].

Some works such as [5,22,43] use methods that analyze the energy required for activating a node versus the improvement in the uncertainty produced by the new measurements. Existing methods in cluster-based tracking schemes decide on the inclusion/exclusion of nodes from the cluster. The method used in our scheme addresses sensor activation/deactivation instead of node inclusion/exclusion: the objective is to select the individual sensors from the cluster nodes which measurements are integrated for tracking. This approach has higher flexibility and enables the aforementioned emerging self-regulated effect.

The difficulties in modeling the interactions of RSSI with the environment has originated a number of techniques based on RSSI training. Most of them rely in two phases; off-line training and localization, where the location estimation is performed based on the previously obtained RSSI map. In [44] the environment is subdivided into small cells and readings are taken in these cells from several known anchor nodes (the training phase). In the localization phase the cell that best fits the current measurement is selected. In [45] a probability distribution of the target location is defined over the area of the movement. The goal is to reach a single mode for this distribution. That mode represents the most likely location of the tracked target. All the aforementioned methods rely on a long training phase where the entire target area is measured with some spatial precision. Such data collection/measurement requires significant labor and the training phase has to be redone in case of changes in the radio environment.

Some techniques, such as [46], train simple RSSI-range models using regressions. The low computational burden in these methods allows the models to be dynamically updated. These methods operate using only RSSI measurements and, although they succeed improving the precision of the RSSI model, their tracking accuracy is far from what can be obtained with other sensors such as cameras. In our method camera measurements are available and each node on-line trains its RSSI-range model using estimates of the target position that are obtained from the cameras. When a sufficient number of cameras are activated it is capable of calibrating RSSI-range models with accuracies similar to those from cameras. Moreover, our method also includes a supervisor that estimates the accuracy of the trained model and decides to employ the trained or a default RSSI-range model.

3. Problem definition

Consider a scenario where a number of static wireless camera nodes have been deployed. Transmission of images is not suitable in our scheme due to bandwidth, delay and energy constraints. Thus, each camera node processes locally its images using simple image segmentation algorithms and transmits the results for data fusion. The target is assumed tagged with a wireless sensor node: each camera node can measure the RSSI of the packets it receives from the target. For simplicity in the description of our scheme we will assume single-target scenarios. Our scheme is also valid for multi-cluster problems but in these cases specific mechanisms to deal with multi-target tracking issues should be added to our scheme. Our scheme is modular can could be combined with these techniques but for brevity and simplicity we preferred to focus on the core of our research and these mechanisms have been left out of the scope of our paper.

In our scheme the camera nodes sensing one target are organized into a single-hop cluster. The cluster has a head – in our case it is the node carried by the target – that acts as the main scheduler of the cluster. Among others, the head is responsible for integrating the measurements gathered by the cluster member nodes and for selecting the nodes that are included/excluded from the cluster. At time k any camera node can be in one of the following modes, see Fig. 1: *Tracking*, the node – represented by a black circle – participates in the cluster, i.e. its measurements are currently being used for tracking; *Alert*, the node – gray circle – is not currently participating in the cluster but it is at single-hop distance from the cluster head and could be included at that time; and *Inactive*, the node is not involved in tracking and cannot be included in the cluster at that time. Nodes in the *Tracking* mode can be divided into three modes depending if they contribute to tracking with camera measurements (*TrackingC*), with RSSI measurements (*TrackingR*) or with both (*TrackingCR*). AS_k is the set of nodes that are in the *Alert* mode at time k. TS_k^C, TS_k^R and TS_k^{CR} are the sets of nodes in *TrackingC*, *TrackingR* and *TrackingCR*. TS_k is the set of nodes in the *Tracking* mode, $TS_k = TS_k^C \cup TS_k^R \cup TS_k^{CR}$.

Our scheme uses three probabilistic components that exploit synergies between RSSI and camera measurements, see Fig. 1. Lack of accuracy of RSSI-range models is the main drawback of RSSI-based tracking. In the first component each camera node uses estimates of the target location in order to on-line train its own RSSI-range model and adapt it to the local environment. Training is performed dynamically to adapt to the target motion. The accuracy of the trained model is monitored, and the component is disabled if required. When camera and RSSI measurements are available this method highly improves the accuracy of RSSI-range models, see Section 5.

The second component is an information-based tool for selecting the individual sensors which measurements are integrated for tracking balancing their reward (information gain) and cost (energy), see Section 6. This component uses the trace of the information matrix as uncertainty metric, which enables decentralized implementation and computation reuse. Finally, a distributed *Infor-*

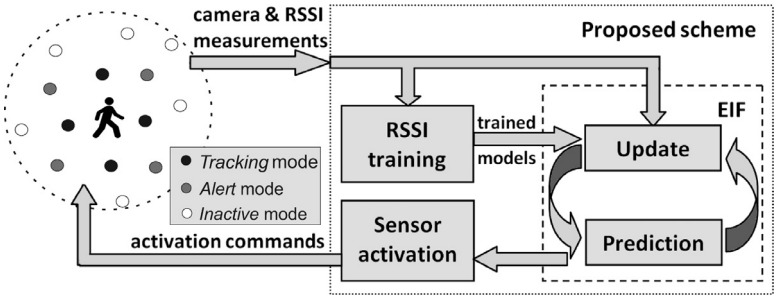


Fig. 1. Schematic illustration of the proposed scheme.

Table 1
Main terms and symbols used in Sections 4–6.

Target state and observations	
$\mathbf{q}_k$	Target state vector at time k
(x_k, y_k)	Target location at time k
$(\dot{x}_k, \dot{y}_k)$	Target local velocity at time k
$d_{i,k}$	Distance from node i to the target at time k
$z^c_{i,k}$	Camera measurement from node i at time k
$z^r_{i,k}$	RSSI measurement from node i at time k
h^c_i	Camera observation model of node i
h^r_i	RSSI observation model of node i
$H^c_{i,k}$	Jacobian of the camera observation model of node i at time k
$H^r_{i,k}$	Jacobian of the RSSI observation model of node i at time k
$\mathbf{A}$	Prediction model of the target motion
R_k	Covariance matrix of the prediction model uncertainty at time k
$Q_{i,k}$	Covariance matrix of the observation noise of node i at time k
Extended Information Filter	
Ω_k	Updated information matrix at time k
ξ_k	Updated information vector at time k
μ_k	Updated mean of the target state at time k
$\Omega_{i,k}$	Contribution of node i to the information matrix update at time k
$\xi_{i,k}$	Contribution of node i to the information vector update at time k
$\overline{\Omega}_k$	Predicted information matrix at time k
$\overline{\xi}_k$	Predicted information vector at time k
$\overline{\mu}_k$	Predicted mean of the target state at time k
RSSI-range model training	
$RSSI_{i,k}$	RSSI of the target packets measured by node i at time k
(a_i, b_i)	Parameters of the RSSI-range model of node i
$\sigma^2_{tm,i}$	Variance of the error of the trained RSSI-range model of node i
Sensor activation/deactivation	
A_k	Universe of possible actions at time k
$J(\mathbf{q}_k, a_k)$	Objective function of action a_k
$r(\mathbf{q}_k, a_k)$	Reward of action a_k
$c(\mathbf{q}_k, a_k)$	Cost of action a_k
p_m	Packet reception rate of the packets sent by node m to the cluster head
$ui(a_k)$	Uncertainty reduction caused by action a_k

mation Filter (IF), see Section 4, is used to efficiently integrate all camera and/or RSSI measurements gathered by the cluster nodes.

Table 1 summarizes the main terms and symbols employed in the paper.

4. Camera and RSSI measurement integration using distributed EIFs

Recursive Bayesian Filters (RBFs) provide a solid statistical basis for measurement integration. Information Filters (IFs) employ the canonical representation of Gaussian distributions. The standard representation of a multivariate Gaussian distribution is based on the mean vector μ and the covariance matrix Σ, whereas the canonical representation is based on the information vector $\xi = \Sigma^{-1}\mu$ and the information matrix $\Omega = \Sigma^{-1}$. IFs are duals to Kalman Filters (KFs). The update stage of IFs is more computationally efficient than that of KFs. Therefore, IFs are more efficient than KFs in cases with high number of measurements and a simple prediction model, as in our problem. As we are dealing with nonlinear measurement models we use the nonlinear version of the IF, the Extended Information Filter [47]. Although distributed EIFs are well known tools for multimodal measurement integration, for completeness and clarity we preferred to include a brief description in the paper.

IFs use a prediction model and an observation model for each type of measurement. We selected a state vector widely used in tracking problems, $\mathbf{q}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T$, where (x_k, y_k) is the 2D target location at time k and $(\dot{x}_k, \dot{y}_k)$ is its local velocities. As many tracking problems, we adopted a simple linear model to represent the local motion of the target. More complex models require a priori knowledge on the target motion, which is often not available in many problems. The camera observation model of each camera node i derives from the simple pin-hole model. Let P_k be the location of the target at time k in the global reference frame G, see Fig. 2. $p_{i,k}$ is the projection of the target on the image plane of camera node i expressed in F_i, the local reference frame of camera node i, which is related to G by transformation matrix T_i. The

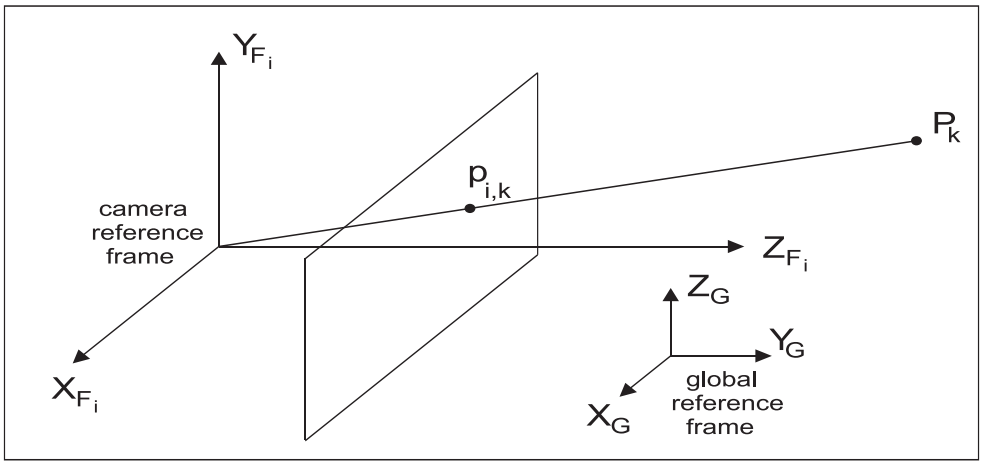


Fig. 2. Reference frame used for camera measurement integration.

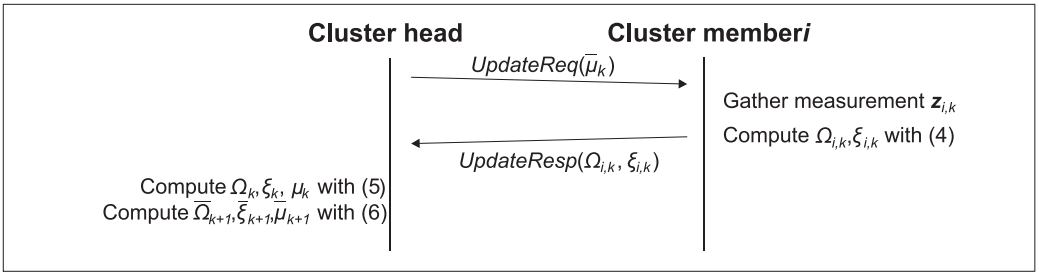


Fig. 3. Operations and communications in the distributed EIF.

observation model for camera i is:

$$p_{i,k} = h_i^c(P_k) = \begin{bmatrix} t_{i,1}\begin{bmatrix} P_k & 1 \end{bmatrix}^T / t_{i,3}\begin{bmatrix} P_k & 1 \end{bmatrix}^T \\ t_{i,2}\begin{bmatrix} P_k & 1 \end{bmatrix}^T / t_{i,3}\begin{bmatrix} P_k & 1 \end{bmatrix}^T \end{bmatrix}, \quad (1)$$

where $t_{i,j}$ is the jth row of T_i.

The scheme proposed in this paper includes a component in which each node trains its own RSSI-range model. If this component is active, the EIF uses the trained model as described in Section 5. Otherwise, the EIF uses the following widely accepted RSSI-range model [13]:

$$RSSI_{i,k} = h_i^r(P_k) = a \log d_{i,k} + b, \quad (2)$$

where $RSSI_{i,k}$ is the RSSI measured by node i at time k from the packets it receives from the target, $d_{i,k}$ is the distance between node i and the target at k and a and b are model parameters. We assume a default RSSI-range model is available. It could be taken from the literature or obtained using the measurements between camera nodes. The parameters of the default model are a_D, b_D and its variance is σ_D^2.

Camera and RSSI-range models are nonlinear. The *Extended Information Filter* (EIF) uses their Jacobians $H_{i,k}^c$ and $H_{i,k}^r$:

$$H_{i,k}^c = \frac{\partial h_i^c(P_k)}{\partial \mathbf{q}_k}, \quad H_{i,k}^r = \frac{\partial h_i^r(P_k)}{\partial d_{i,k}} \frac{\partial d_{i,k}}{\partial \mathbf{q}_k} \quad (3)$$

RBFs assume that measurement noise is statistically independent. Camera measurement noise is known to be statistically independent. RSSI measurement noise is considered correlated in many works [48–50] due to the influence of the environment when models such as (2) are adopted. However, as will be explained in Section 5 the RSSI measurement noise in our scheme has a negligible level of correlation and assuming uncorrelated measurement noise involves no practical influence in our scheme enabling the integration of RSSI measurements in RBFs.

In RBFs measurement integration is performed using the Bayes rule, which becomes an addition if represented in logarithmic form [47]. Hence, IFs can integrate measurements collected in a decentralized manner. Our scheme uses a distributed EIF in which each cluster member computes its contribution to the EIF update and sends it to the cluster head, which computes the updated stage by summing up the contributions it receives. The operation of the distributed EIF is shown in Fig. 3. The predicted state for time k ($\overline{\Omega}_k$, $\overline{\xi}_k$ and $\overline{\mu}_k$) is assumed available. It was computed at k 1. First, the cluster head broadcasts a *UpdateReq* packet that includes $\overline{\mu}_k$. Each camera node i in *Tracking* mode that receives the packet gathers measurement $z_{i,k}$ using its currently active sensors. If the node is in the *TrackingC* mode, it gathers an image and applies simple image processing techniques to obtain $z_{i,k}^c$, the coordinates of the center of the target in its image plane. If the node is in the *TrackingR* mode, it measures $z_{i,k}^r$, the RSSI of the packet. If the node is in the *TrackingCR* mode it gathers one measurement of each type. Next, the node calculates the Jacobian for each of its active sensors and computes its contribution to the EIF update using the following expressions:

$$\begin{aligned} \Omega_{i,k} &= H_{i,k}^T Q_{i,k}^{-1} H_{i,k} \\ \xi_{i,k} &= H_{i,k}^T Q_{i,k}^{-1} [z_{i,k} - h_i(\overline{\mu}_k) + H_{i,k}\overline{\mu}_k] \end{aligned} \quad (4)$$

These expressions apply for each type of sensors. h_i stands for h_i^c or h_i^r – the camera or RSSI observation models – or for both. The same applies for $H_{i,k}$ with $H_{i,k}^c$ and $H_{i,k}^r$. For the camera nodes in the *TrackingCR* mode, $\Omega_{i,k}$ and $\xi_{i,k}$ are obtained summing up the contributions from both measurements, each of them computed sequentially with the expressions in (4). Then, each node in *Tracking* transmits to the head a *UpdateResp* packet that includes $\Omega_{i,k}$ and $\xi_{i,k}$. The head receives these packets and computes the updated

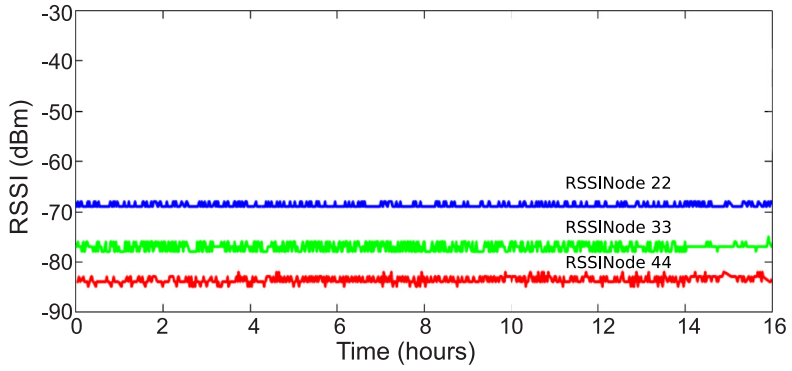


Fig. 4. 40,000 different RSSI measurements between three emitters and one receiver WSN node.

state – Ω_k, ξ_k and μ_k – using the following expressions:

$$\Omega_k = \overline{\Omega}_k + \sum_i \Omega_{i,k}, \quad \xi_k = \overline{\xi}_k + \sum_i \xi_{i,k}, \quad \mu_k = \Omega_k^{-1}\xi_k \tag{5}$$

Next, the cluster head computes the predicted state with the following expressions:

$$\begin{aligned} \overline{\Omega}_{k+1} &= (A\Omega_k^{-1}A^T + R_{k+1})^{-1} \\ \overline{\xi}_{k+1} &= \overline{\Omega}_{k+1}\overline{\mu}_{k+1} \\ \overline{\mu}_{k+1} &= A\mu_k \end{aligned} \tag{6}$$

It is easy to notice that the computation of the EIF update stage is distributed among all the cluster members. The head only adds the EIF update contributions from all nodes and executes the EIF prediction stage, which requires a low number of operations due to the simple prediction model adopted. Hence, the distributed EIF can be executed roughly in constant time regardless of the cluster size. Also, the RSSI between the target and the cluster nodes can be measured from the packets they actually interchange, involving negligible energy consumption and computational effort.

5. On-line RSSI-range model training

The RSSI-range model between two wireless nodes is critically affected by the local environment surrounding the nodes. For instance, Fig. 4 shows 40,000 RSSI readings between three TelosB nodes deployed in the *CONET Integrated Testbed*. The experiment was performed in stationary conditions and all the RSSI measurements were very similar evidencing that it is the interactions with the environment and not measurement noise itself what originates RSSI difficulties for localization and tracking. Our approach is not to model these interactions but to circumvent these difficulties and provide a solution suitable for online implementation in COTS devices. In our method each node uses the current target location to dynamically adapt its own RSSI-range model to the local environment of the emitter and the receiver.

Let P_k be the target location at time k. Let $RSSI_{i,k}$ be the RSSI measured by node i from the packets it received from the target at time k. Assuming that each node knows its location, any node i in the *Tracking* mode can collect a set of measurement pairs $\{(RSSI_{i,k}, d_{i,k})\}$, where $d_{i,k}$ is the distance between node i and P_k. The objective is to on-line train a RSSI-range model for node i adapted to the local environment of the target at time k. We adopted a linear RSSI-range model:

$$RSSI_{i,k} = a_i d_{i,k} + b_i, \tag{7}$$

which reflects the local nature of the model and can be efficiently trained on-line by regression:

$$a_i = \frac{\sum_k RSSI_{i,k} d_{i,k} - \overline{RSSI}_i \sum_k d_{i,k}}{\sum_k (d_{i,k})^2 - \overline{d}_i \sum_k d_{i,k}}, \tag{8}$$

$$b_i = \overline{RSSI}_i - (a_i \overline{d}_i), \tag{9}$$

where $\overline{RSSI}_i$ and $\overline{d}_i$ are the mean of $RSSI_{i,k}$ and $d_{i,k}$. To cope with the target motion only the last few M pairs are used for fitting in order to avoid measurements taken with different local environments.

The actual target location is not known. Instead, we use estimates obtained with an auxiliary EIF as that presented in Section 4 but which only integrates camera measurements. Of course, if not addressed carefully camera-based estimation inaccuracies may perturb the RSSI training. To cope with that the RSSI training includes a supervisor that estimates the accuracy of the trained model and decides which RSSI-range model should be used for that node: the trained or the default model. Take $d'_{i,k} = d_{i,k} + u_i$, where $d'_{i,k}$ is the distance from node i to the target location estimated by the auxiliary EIF, $d_{i,k}$ is the actual distance and u_i is the estimation error. $RSSI_{i,k} = a_i d_{i,k} + b_i$ is the exact RSSI-range model and the measured RSSI is $RSSI'_{i,k} = RSSI_{i,k} + v_i$, where v_i is the RSSI measurement error. The training method uses pairs $\{(RSSI'_{i,k}, d'_{i,k})\}$ to fit the model $RSSI'_{i,k} = a_i d'_{i,k} + b_i + v_i$. It is easy to check that:

$$RSSI'_{i,k} = a_i(d_{i,k} + u_i) + b_i + v_i = RSSI_{i,k} + a_i u_i + v_i \tag{10}$$

Assuming that u_i and v_i are Gaussian White noises with zero means and variances $\sigma^2_{u_i}$ and $\sigma^2_{v_i}$, the variance of the error of the trained model can be expressed as:

$$\sigma^2_{tm,i} = a_i^2 \sigma^2_{u_i} + \sigma^2_{v_i} \tag{11}$$

$\sigma^2_{tm,i}$, the variance of the error of the trained RSSI-range model, depends on $\sigma^2_{v_i}$, $\sigma^2_{u_i}$ –the variance of the target error estimated by the auxiliary EIF– and a_i, the slope of its RSSI-range model. Trained RSSI-range models with higher a_i are more sensitive to target location errors. If a trained model has $1/a_i = 0$, it is approximated by $1/a_i = \epsilon$ and b_i is recomputed using (9) with the new a_i.

Fig. 5 (top) shows the default RSSI-range model computed by fitting (2) with RSSI measurements (in red color) from every pair of nodes deployed in the *CONET Integrated Testbed*. Fig. 5(bottom) shows two trained RSSI-range models for node i computed at two different times along the target path. The differences in the fitting error in both cases are evident. Trained RSSI-range models are specific for each node and valid for current local surroundings of the target but they are significantly more accurate than the default model.

The auxiliary EIF estimates the target location and its covariance. Thus, each node can compute $\sigma^2_{tm,i}$ using (11) to estimate the accuracy of its trained model. Each node decides the RSSI-range model it uses for integrating its measurements in the main EIF filter. For instance, it chooses the trained model if $\sigma^2_{tm,i} < \sigma^2_D$. Thus, the malfunctioning of the auxiliary EIF or badly trained RSSI-range models can be easily detected and avoided. As shown in

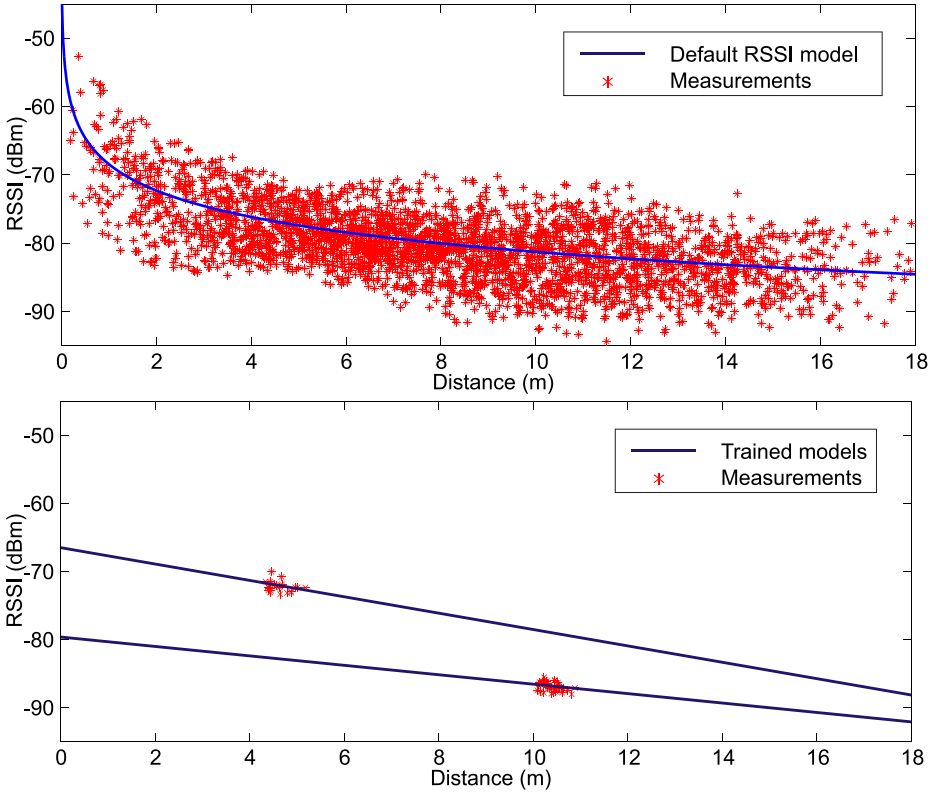


Fig. 5. RSSI-range models obtained in an experiment: (top) default logarithmic model, (bottom) trained linear models obtained at two times along the target path. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Section 7 the scheme is robust to the most common sources of errors in WCNs.

RSSI is often modeled by (2), which does not consider multi-path propagation effects. Actual radio propagation is not well captured by model (2) and multi-path effects originate noise, which is correlated since it is influenced by the particular environment surrounding the emitter and the receiver. The objective of the adopted training-based RSSI-range approach is to capture the full radio propagation – including multi-path effects – regressing a simple model with few consecutive actual RSSI measurements taken at the specific environment and the specific locations of the emitter and the receiver. The trained RSSI-range model for a node is used as the observation model for that node in the EIF and the regression error is the observation noise for that node. The trained RSSI-range model captures the effects of multi-path propagation significantly better than models such as (2) and the regression observation noise is very low affected by multi-path effects. Besides, regression error are very small as can be noticed in Fig. 5. We experimentally analyzed the correlation of our RSSI measurement noise in a wide variety of representative experiments and concluded that it has negligible level of correlation, enabling its valid integration in Bayesian EIF schemes. Besides, along the paper it is shown with a significant variety of experimental results that the proposed scheme always performed satisfactorily, even in the robustness analysis in Section 7.3.

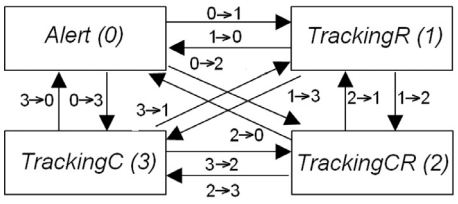
The proposed training method can be seamless integrated together with the distributed EIF. When node i with active camera (i.e. in modes *TrackingC* or *TrackingCR*) receives the *UpdateReq* packet from the head it performs as follows: measures $RSSI_{i,k}$ – the RSSI of the packet, computes $d_{i,k}$ using the target estimate from the auxiliary EIF and uses the new pair $\{(RSSI_{i,k}, d_{i,k})\}$ for fitting its RSSI-range model. Next, it decides whether to use the trained or the default model.

6. Sensor activation/deactivation

This method decides which sensors should be integrated in tracking analyzing the usefulness of their measurements and the resources they consume. Non-selected sensors are kept inactive saving energy. We adopt an approach that decides the activation/deactivation action $\hat{a}_k$ that optimizes an utility function $J(\mathbf{q}_k, a_k)$ at each time k. Long-term optimization involves a high number of operations and scales badly with the problem size. Let A_k be the set of possible actions regarding sensor activation/deactivation that can be performed in the cluster at time k. Each action a_k impacts on how well the target is sensed, which can interpreted as a reward $r(\mathbf{q}_k, a_k)$. Each action a_k also involves an increase in the consumption of resources such as energy, which can be interpreted as a cost $c(\mathbf{q}_k, a_k)$. Of course, reward and cost can be positive or negative. Our method selects the action that maximizes: $J(\mathbf{q}_k, a_k) = r(\mathbf{q}_k, a_k) - \alpha c(\mathbf{q}_k, a_k)$, where α is a weighting factor.

Fig. 6 (left) depicts the mode transition model adopted in our sensor activation/deactivation method. To simplify the notation each mode has been assigned with a number: 0 for *Alert*; 1 for *TrackingR*; 2 for *TrackingCR*; and 3 for *TrackingC*, see Fig. 6. $Action^i_{0\to 1}$ changes the mode of camera node i from *Alert* to *TrackingR*. A node in *Alert* mode can be changed to *TrackingR* ($Action^i_{0\to 1}$), to *TrackingC* ($Action^i_{0\to 3}$) or to *TrackingCR* ($Action^i_{0\to 2}$).

We take $c(\mathbf{q}_k, a_k)$ as the difference in the energy consumed by the camera node before and after performing action a_k, see Fig. 6 (right). In some cases it is interesting to compute the cost for an action on node i taking into account the remaining energy at that node. For simplicity, we use the same cost for every node. Negative costs mean energy savings. $c2$ is the energy consumed by gathering and processing one image from the camera module. $c1$ is the energy consumed by gathering the RSSI of an incom-



Action	Cost
0→1	c1
1→2	c2
3→2	c1
0→3	c2
0→2	c1+c2
1→3	c2-c1

Action	Cost
1→0	-c1
2→1	-c2
2→3	-c1
3→0	-c2
2→0	-c1-c2
3→1	c1-c2

Fig. 6. Adopted transition model and action costs.

ing packet. During the *Inactive* mode the node is sleeping, i.e. in low-energy mode. When a node in mode *Inactive* receives a packet from the cluster head it changes to mode *Alert*. While in *Alert*, the node keeps its camera off and does not gather RSSI measurements but interchanges packets with the cluster head. In mode *TrackingR* the node performs as in *Alert* and also gathers RSSI measurements of the packets it receives from the cluster head. The difference in energy consumption between *Alert* and *TrackingR* is $c1$, the energy of measuring the RSSI value of a packet. In most COTS nodes RSSI can be measured directly by the radio module involving very low energy consumption and computational effort. Although $c1$ is not zero, it can be considered negligible when compared to $c2$.

Assume that node m is currently in mode *Alert*. We evaluate the reward of $a_k = Action^m_{0\to3}$, i.e. change node m from *Alert* to *TrackingC*. $r(\mathbf{q}_k, a_k)$ is the expected uncertainty decrease in the target state after performing $Action^m_{0\to3}$, i.e. after integrating in the EIF the camera measurement from node m. The contribution of node m to the EIF update reaches the head if: (1) if node m receives the *UpdateReq* packet sent by the cluster head and (2) if the head receives the *UpdateResp* packet with the contribution from node m. Both events are statistically independent. Let p_m be the Packet Reception Rate (PRR) from node m to the head. Assuming PRR is symmetric, the reward of a_k is:

$$r(\mathbf{q}_k, a_k) = p_m^2 ui(a_k), \tag{12}$$

where $ui(a_k)$ is the expected uncertainty decrease after performing a_k. In the following, we obtain $ui(a_k)$. TS_k is the set of nodes in mode *Tracking* at time k. If a_k is performed, the updated information matrix is estimated summing $\overline{\Omega}_{k+1}$ to the contributions from all active nodes, i.e. nodes in TS_k and node m:

$$\Omega^{a_k}_{k+1} = \overline{\Omega}_{k+1} + \Omega_{m,k+1} + \sum_{i\in TS_k} \Omega_{i,k+1}, \tag{13}$$

where $\Omega_{i,k+1}$ is the predicted contribution from camera node i to the EIF update and it is computed similarly to (4): $\Omega_{i,k+1} = (H_{i,k+1})^T Q^{-1}_{i,k+1} H_{i,k+1}$, where $H_{i,k+1}$ is the predicted Jacobian of node i computed with $\overline{\mu}_{k+1}$. $H_{i,k+1}$ refers to camera or RSSI Jacobians or both depending on which sensors of node i are active.

Similarly, if no action is performed, the updated information matrix is:

$$\Omega^{na}_{k+1} = \overline{\Omega}_{k+1} + \sum_{i\in TS_k} \Omega_{i,k+1} \tag{14}$$

$ui(a_k)$ is computed as the difference between the uncertainty in Ω^{na}_{k+1} and in $\Omega^{a_k}_{k+1}$. Entropy is one of the most widely-applied uncertainty metrics. However, computing entropy gain for every action in A_k involves a high number of operations and scales badly with the problem size. The trace of the information matrix is also a well known uncertainty metric that has also been employed for sensor selection, see e.g. [42]. We adopt the trace of the information matrix due to its interesting properties in our problem as will be pointed out below. Taking $ui(a_k) = tr(\Omega^{a_k}_{k+1}) - tr(\Omega^{na}_{k+1})$ we obtain:

$$r(\mathbf{q}_k, a_k) = p_m^2 tr(\Omega_{m,k+1}) \tag{15}$$

Similarly, if node n is currently in mode *TrackingC* the reward of changing its mode to *Alert* is $r(\mathbf{q}_k, a_k) = -p_n^2 tr(\Omega_{n,k+1})$. Using trace as uncertainty metric enables the distributed implementation of the method with evident computational and scalability advantages. Each node i can easily compute by its own $\Omega_{i,k+1}$ and hence the reward of the actions it is involved in. Therefore, every node transmits its rewards and costs to the head, which only has to select that which maximizes $J(\mathbf{q}_k, a_k)$. Section 7.2 compares the performance of the adopted activation method versus the same method but using entropy as uncertainty metric. In the experiments performed the tracking error and consumed energy was similar in both cases.

The presented sensor activation method can be straightforward integrated with the distributed EIF and RSSI-range model training methods. It is executed in the cluster head after the execution of the EIF as follows. The cluster head broadcasts packet *ActionReq* that includes $\overline{\mu}_{k+1}$. Each camera node i in modes *Tracking* or *Alert* receiving the packet: (1) computes the Jacobian $H_{i,k+1}$ using $\overline{\mu}_{k+1}$; (2) calculates $\Omega_{i,k+1}$; and (3) transmits to the head packet *ActionResp* that includes $tr(\Omega_{i,k+1})$ and p_i, its PRR to the head. With them, the head performs the sensor activation method and broadcasts the selected action, see Fig. 7. Each node evaluates $r(\mathbf{q}_k, a_k)$ using one-step predictions of the state – $\overline{\mu}_{k+1}$, which is computed in the EIF prediction. Hence, the execution of the sensor activation method starts after the EIF prediction stage. The full scheme – joining Figs. 3 and 7 – requires two communication steps between the head and the rest of the cluster members: one for the EIF update and one for the sensor activation method. The latter communication step could be avoided if using $\overline{\mu}^*_{k+1} = A\overline{\mu}_k$, two-step predictions of the state, instead of $\overline{\mu}_{k+1}$ for the evaluation of $r(\mathbf{q}_k, a_k)$. In that case the head computes $\overline{\mu}^*_{k+1}$ at the EIF update and broadcasts it in the *UpdateReq* packet. Each node i in *Tracking* and *Alert* that receives the packet: (1) computes $\overline{\Omega}^*_{i,k+1}$ as in (3) but using $\overline{H}^*_{i,k+1}$, the Jacobian predicted for k+1 computed using $\overline{\mu}^*_{k+1}$; and (2) transmits $tr(\overline{\Omega}^*_{i,k+1})$ and p_i in *UpdateResp* packets. In the experiments performed, the scheme using $\overline{\mu}^*_{k+1}$ transmitted 32% less packets than that using $\overline{\mu}_{k+1}$ with no significant difference in tracking error or energy consumption ($< 2\%$).

7. Experiments

The proposed scheme was experimented in the *CONET Integrated Testbed* [1]. Robots were used as targets in order to enable experiment repeatability. Robots were *Pioneer 3AT* and executed the AMCL algorithm [51] to self-locate. AMCL estimates the robot pose using measurements from odometry and laser range-finders by means of a Particle Filter. In preliminary experiments we measured AMCL had a mean localization error lower than 5 cm, sufficient for our case. These measurements were taken as ground truth. 21 camera nodes were deployed on the room floor, each of them comprised of a *CMUcam3* camera module connected to a *Crossbow TelosB*, see Fig. 8. Each *CMUcam3*, internally calibrated using the method in [52], captured 352 x 288 RGB images and executed efficient color and motion segmentation methods. The *CMUcam3* sent to its TelosB node the coordinates of the center of the

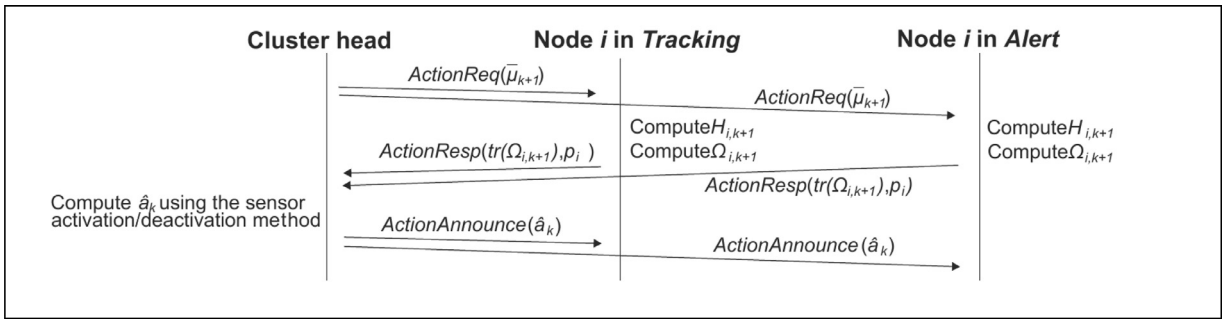


Fig. 7. Integration in the sensor activation/deactivation method.

Fig. 8. (Left) Picture of the *CONET Integrated Testbed* taken in the experiments. (Right) Camera nodes used in the experiments.

region segmented in each image. Camera nodes also measured the RSSI of the packets they received from the robot, which carried another *TelosB*.

Instead of using a default RSSI-range model taken from the literature, we performed preliminary tests to obtain a model for our specific environment by fitting the RSSI measurements between each pair of nodes. We used that model, shown in Fig. 5(top), as our default RSSI-range model. It was found that the image segmentation methods had errors with a standard deviation of 18 pixels in X and Y axes: we took them as the camera measurement covariance. For the sensor selection method we used the energy consumption given by the manufacturers. *CMUcam3* consumes 650 mW when active [53]. It is turned off when inactive. *TelosB* nodes can be in two modes: inactive, a.k.a. *low-energy mode*, which consumes 7.2 mW, and the active mode, in which it consumes 69 mW [54].

7.1. Proof of concept

Fig. 9 shows the results of the proposed scheme in an experiment. The ground truth robot location is represented in blue and the estimated location is in red. The camera local frames are also shown. Fig. 9 shows in green the path followed by the robot between second 550 and second 570. At the beginning of the experiment nodes ID1, ID2 and ID3 were in mode *Inactive*. Then, nodes ID1, ID2 and ID3 received packets from the target and changed their mode to *Alert*. In the next iterations, the head decided to activate their cameras and they changed to mode *TrackingC*. This time is taken as $k = 0$ in Fig. 10, which shows in a simplified way the operation of the proposed scheme between $k = 0$ and $k = 20$.

Fig. 10 analyzes the transitions between states *TrackingR* and *TrackingC* for nodes ID1, ID2 and ID3. Only the changes between *TrackingC* and *TrackingC* are analyzed. Fig. 10(c) shows the values of $J(Action_{3\rightarrow 1})$ for the nodes that are in *TrackingC*, i.e. those that can change to mode *TrackingC*. Fig. 10(d) shows the values of $J(Action_{1\rightarrow 3})$ for the nodes that are in *TrackingC*, i.e. those that can change to *TrackingC*.

While ID1, ID2 and ID3 were in mode *TrackingC* the three nodes contributed to tracking with camera measurements. At the beginning these nodes had few measurements for RSSI-range model training. Their trained models had high covariance σ^2_{tm}. Hence, the three nodes employed the default RSSI-range model. Also, the sensor activation method decided to keep the three nodes in *TrackingC*, see Fig. 10(a). At $k = 4$, the RSSI-range model for node ID1 was trained and had low uncertainty σ^2_{tm}= 13.5. Fig. 10(b) shows in dashed line the threshold below which the trained RSSI-range model is preferred over the default model. Hence at $k = 4$ ID1 started using its trained RSSI-range model. Also, at $k = 4$ $Action^{ID1}_{3\rightarrow 1}$ was selected since it obtained the maximum utility function $J(Action^{ID1}_{3\rightarrow 1})$= 1.49, see Fig. 10(c). The cluster head broadcasted the action in an *ActionAnnounce* packet. ID1 received the packet and at $k = 5$ it changed to *TrackingR* and turned its camera off, see Fig. 10(a). At $k = 7$ ID2 changed to mode *TrackingR*. ID3 remained at *TrackingC* during the whole interval.

The modes of the three nodes remained unchanged until $k = 14$. During this interval, nodes ID1 and ID2 did not take new camera measurements and their RSSI-range model degraded as the target moved. At k=14 the covariance σ^2_{tm} of ID2 grew above the threshold – dashed line in Fig. 10(b). Hence, ID2 started using the default RSSI-range model. At that time activating the camera of ID2 significantly improved the overall tracking accuracy. The sensor activation method selected $Action^{ID2}_{1\rightarrow 3}$ since it obtained the maximum utility, $J(Action^{ID2}_{1\rightarrow 3})$. At $k = 15$ ID2 changed its mode to *TrackingC* and activated its camera. At the end of this experiment in average each camera node remained in mode *Inactive* 45.2% of the time, 38.6% in *Alert*, 7.4% in *TrackingC*, 3.9% in *TrackingR* and 4.9% in *TrackingCR*. The mean tracking error was 32.9 cm, with errors of 21.3 cm and 25.07 cm in X and Y, respectively. The mean power consumed by each node was 97 mW.

The joint execution of the three components generates a behavior in which camera measurements are used to train RSSI-range

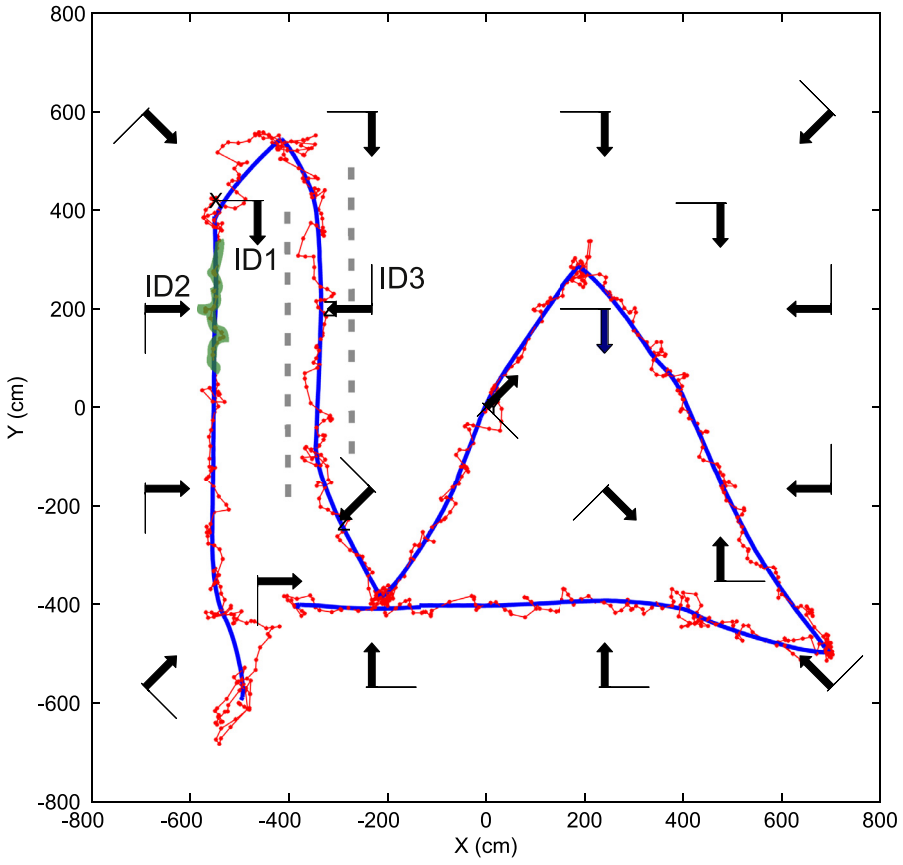


Fig. 9. Result of the proposed scheme in one experiment: estimated location (in red) and ground truth in solid (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

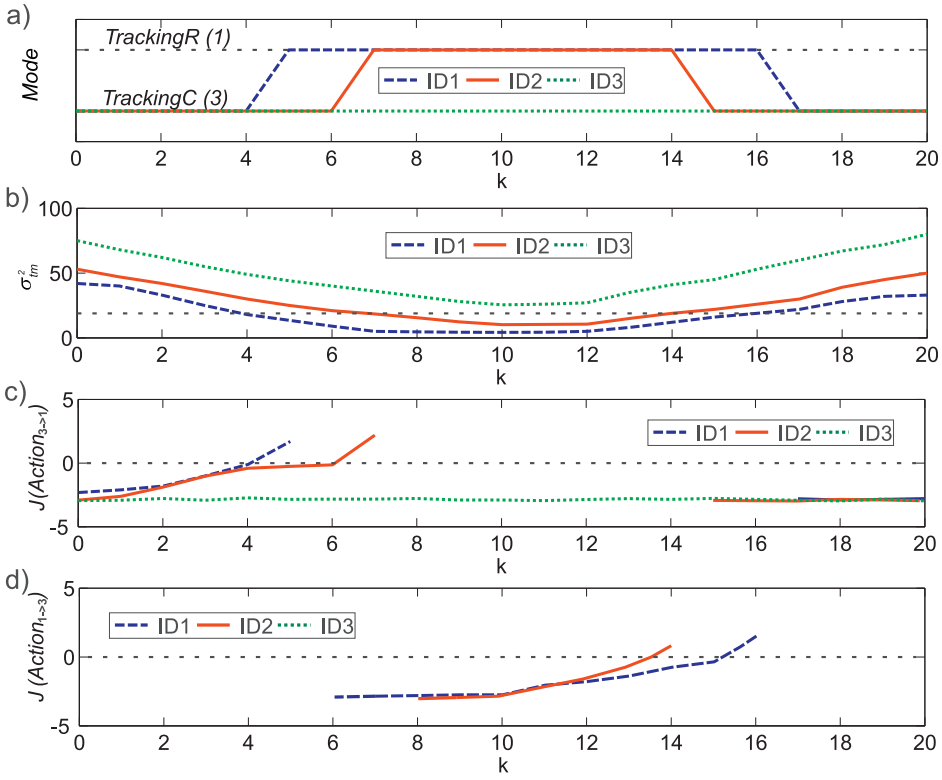


Fig. 10. Performance of the proposed scheme in the experiment shown in Fig. 9 between second 550 ($k = 0$) and second 570 ($k = 20$): (a) modes of nodes ID1, ID2 and ID3; (b) σ^2_{tm} covariance of the trained RSSI-range models for the three camera nodes; (c) values of $J(Action_{3\rightarrow 1})$ for all the camera nodes that are in the *TrackingC mode*; and (d) values of $J(Action_{1\rightarrow 3})$ for all the camera nodes that are in mode *TrackingR*.

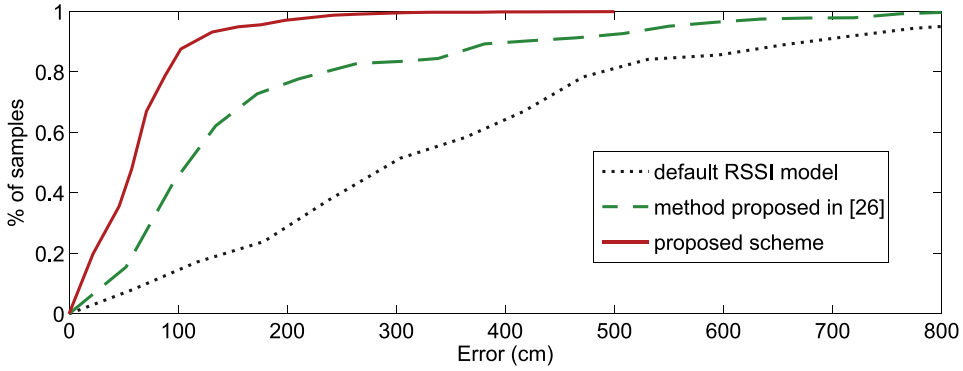


Fig. 11. Cumulative tracking error obtained by RSSI-only tracking using: the default RSSI-range model, the RSSI calibration method in [29] and the training method proposed in Section 5.

models and once trained, the sensor selection method tends to deactivate cameras due to their higher energy consumption. The behavior is self-regulated. When too many cameras are inactive, the accuracy of trained RSSI-range models degrade, involving higher uncertainty in the overall target estimation, which makes the sensor activation method to activate cameras in order to reduce uncertainty.

7.2. Performance evaluation

The proposed scheme was evaluated in simulations and experiments assuming random settings and robot trajectories. In our experiments random settings and robot trajectories means the following. The robot motion was controlled by the Player/Stage random walk functionality, in which it is given pseudo-random velocity commands. The camera nodes were deployed randomly using a uniform distribution in the range of the room size. The camera roll and pith angles of all the cameras were zero and the yaw angles were selected such that the optical axes pointed at the center of the room.

First, we evaluate the accuracy of the training RSSI-range model method proposed in Section 5 in a set of experiments with 20 random settings and robot trajectories. We performed the experiments, logged all RSSI measurements and off-line executed the EIF that integrates the RSSI measurements using three different RSSI-range models: the default RSSI-range model, the RSSI calibration method presented in [29] and the RSSI-range model training method proposed in this paper. Fig. 11 shows the cumulate robot localization errors obtained when using the three RSSI-range models. The RSSI training method proposed in this paper has significantly higher accuracy: the mean error is 55 cm and the error was lower than 90 cm in 80% of the samples. Errors were significantly higher in the other two cases even considering that both RSSI-range models were generated for that specific environment. The adopted training method behaves better because it estimates the target location with cameras and trains the RSSI-range model dynamically considering the local surroundings of the target and of the static node.

M, the number of measurements used in RSSI-range model training, seems an important parameter. We performed tracking simulations using only RSSI measurements adopting the RSSI-range model trained with the last $M \in [2, 20]$ measurements. For each value of M 100 simulations in random settings and target trajectories were performed. Fig. 12 shows the average tracking error obtained. The RSSI-range model training performs better with lower number of measurements. This result can be expected. Target is in motion and with a higher number of measurements the model is not fitted to the target surroundings but to a wider area. We selected $M = 5$ in the experiments shown in the paper. Too low values of M could make RSSI training too sensitive to spurious errors.

Table 2
Evaluation and comparison of the proposed method in series of 40 experiments.

	Scheme1	Scheme2	Scheme3	Scheme4	Proposed
e_{av} **(cm)**	205.6	28.59	32.50	32.95	32.84
CM_{av}	–	11.50	4.59	3.11	2.56
RM_{av}	13.5	–	–	3.11	3.38
PW_{av} **(mW)**	39.28	428.61	162.76	112.57	96.38
PW_{max} **(mW)**	69	719	719	719	719
B_{av}	9691	8256	6288	1313	1335
P_{av}	14.54	12.51	21.91	15.17	14.95

The sensor activation method as presented in Section 6 enables the independent activation/deactivation of each of the sensors of each node. Another approach is to jointly activate/deactivate all the sensors of the selected node, i.e. a simplification of the transition model shown in Fig. 6 in which *TrackingR* and *TrackingC* do not exist. Fig. 13 shows the resulting mean tracking error and power consumption for both approaches in 100 simulations with random settings and target trajectories. The superiority of the independent sensor activation is clear as expected: with similar error it reduces power consumption in 18%.

Next, the performance of the proposed scheme in experiments performed in the *CONET Integrated Testbed* is analyzed and compared with other schemes. 40 different experiments with different settings and robot trajectory were performed, each was executed 5 times using the testbed repeatability. Camera and RSSI measurements of all the camera nodes were logged and off-line processed using different methods. Table 2 summarizes their results: average tracking error (e_{av}), average number of camera measurements integrated per tracking cycle (CM_{av}), average number of RSSI measurements integrated per cycle (RM_{av}), average power consumption by each node (PW_{av}), maximum power consumption of a node (PW_{max}), average number of multiplications of the cluster head per cycle (B_{av}) and the average number of transmitted packets per cycle (P_{av}).

The following schemes were compared. *Scheme1* implements a conventional EKF scheme, similar to that described in [55], which integrates RSSI readings from all anchor nodes using the default RSSI-range model. It obtained the lowest energy consumption but had a mean tracking error of 205.6 cm, unsuitable for many applications. The head integrated RSSI readings from each node involving a very high computational effort and number of transmitted packets. *Scheme2* is a WCN-based tracking scheme similar to that described in [18]: all cameras are kept active while they see the object; and an EKF is used to integrate in a centralized way all the measurements from active cameras. *Scheme2* obtained the

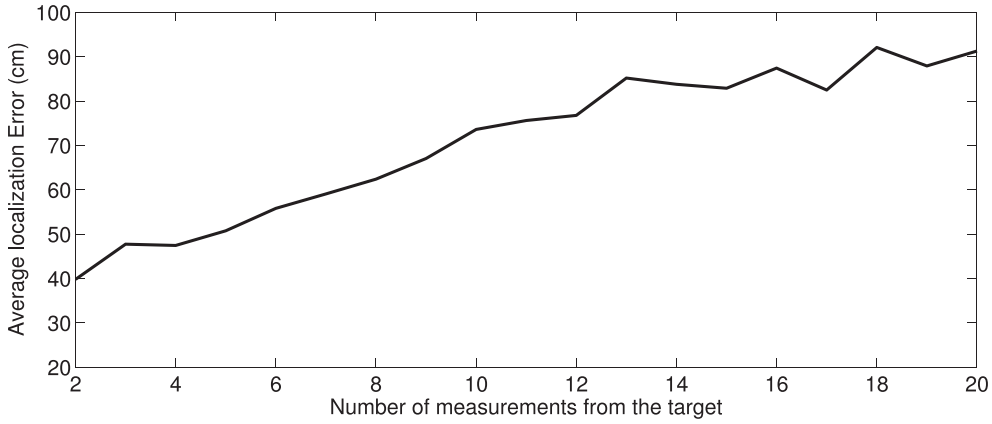


Fig. 12. Mean tracking error VS number of measurements used in RSSI-range model training.

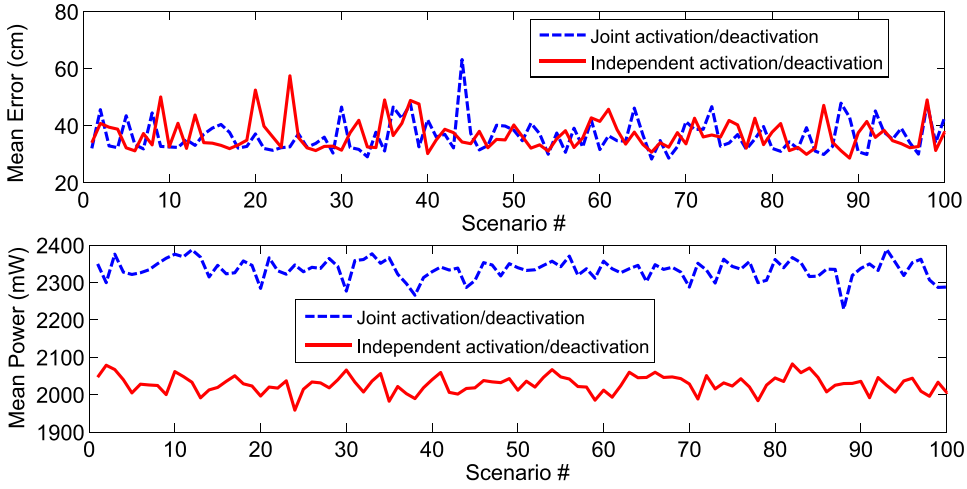


Fig. 13. Comparison between the proposed independent sensor activation/deactivation method and a simplified activation/deactivation method in which *TrackingR* and *TrackingC* do not exist: (top) mean tracking error, (bottom) mean power consumption.

highest accuracy at the expense of requiring an average of 11.5 active cameras along the experiment. The average energy consumption for each camera was 428.61 mW.

Scheme3 is the WCN-based tracking scheme proposed in [56]. It efficiently integrates camera measurements using a distributed EIF and node inclusion/exclusion tools. It dynamically deactivates the camera nodes that do not provide informative measurements and only an average of 40% of the cameras that sensed the object were kept active achieving a reduction of 62% in energy consumption w.r.t. *Scheme2*.

Scheme4 integrates camera and RSSI measurements using the RSSI-range model training method proposed in Section 5 and the joint camera node activation instead of the independent sensor activation method. Joint activation uses a simplification of the independent sensor activation transition model shown in Fig. 6 in which *TrackingR* and *TrackingC* do not exist. In this case an average of only 3.11 camera nodes (which provided simultaneously camera and RSSI measurements) were active in the experiments, which consumed 26.6% less than *Scheme3* with almost the same accuracy.

The proposed scheme is the same as *Scheme4* but using the independent sensor selection presented in Section 6. The proposed scheme only required an average of only 2.56 active cameras and 3.38 nodes in mode *TrackingR*. It required almost the same computational effort than *Scheme4* but its energy consumption was 15% lower. It also consumed less than schemes based on only cameras: 40% less than *Scheme3* and 78% less than *Scheme2*. In the proposed scheme batteries lasted 4.4 times longer than in *Scheme2*.

Fig. 14 compares the performance of the proposed scheme against a scheme that comprises the same components but uses entropy as uncertainty metric in a set of experiments with different settings and robot trajectories. The performance in both cases is very similar: the scheme that uses entropy in average had 12% lower tracking error and 7% higher energy consumption. However, using entropy required 370% more operations. We can reach to several conclusions. First, if we consider only tracking error and energy consumption there is not a clear advantage in using entropy. α, the weighting factor used in Section 6, imposes a trade-off between tracking error and energy consumption. The same value of α involves a different trade-off when using entropy or when using the trace of the information matrix. With the value of α used using entropy tends to activate more cameras than using the trace, which results in lower tracking error but also higher energy consumption. Second, in contrast to entropy, the trace of the information matrix discards the off-diagonal elements in the state estimation, which could have impact if the states are highly correlated. This is not the case: the states of the state vector are poorly correlated and discarding the off-diagonal elements of the information matrix has low influence in absolute terms. Third, in our scheme using the trace enables distributed computation, which has clear advantages in burden and scalability.

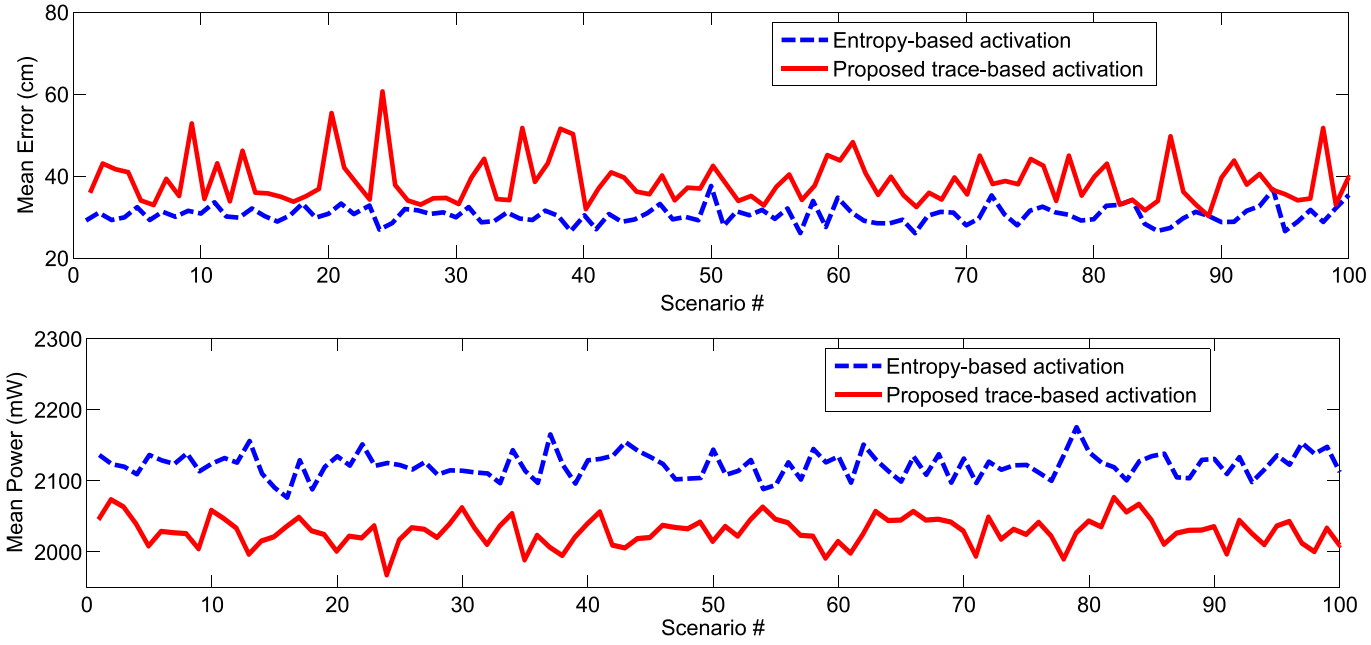


Fig. 14. Comparison between the proposed scheme and a scheme that comprises the same components but uses entropy as uncertainty metric: (top) mean tracking error and (bottom) mean energy consumption.

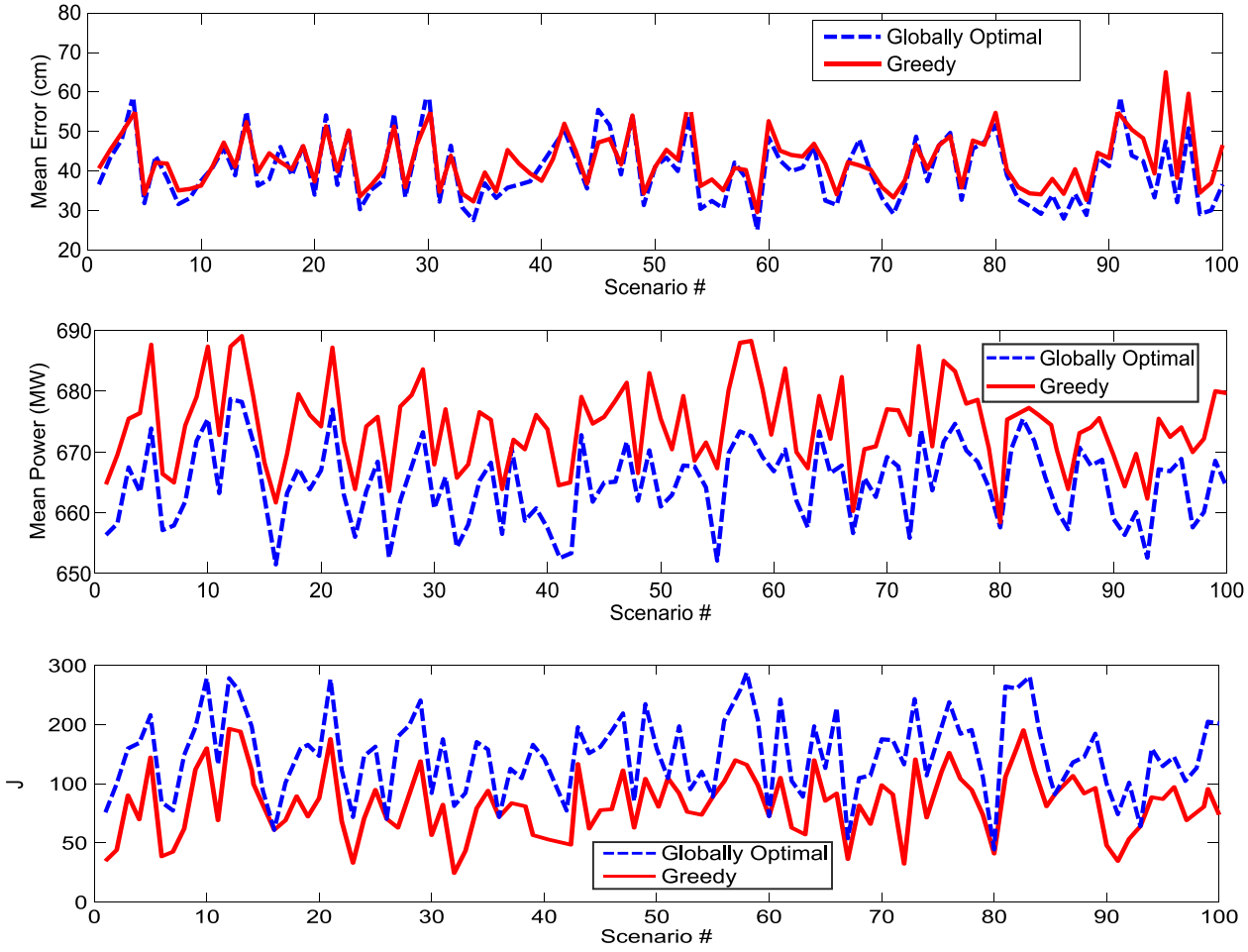


Fig. 15. Comparison of the impact of greedy versus globally optimal activation/deactivation in the proposed scheme: (top) mean tracking error, (center) mean power consumption, and (bottom) value of the sum of J along the full experiment.

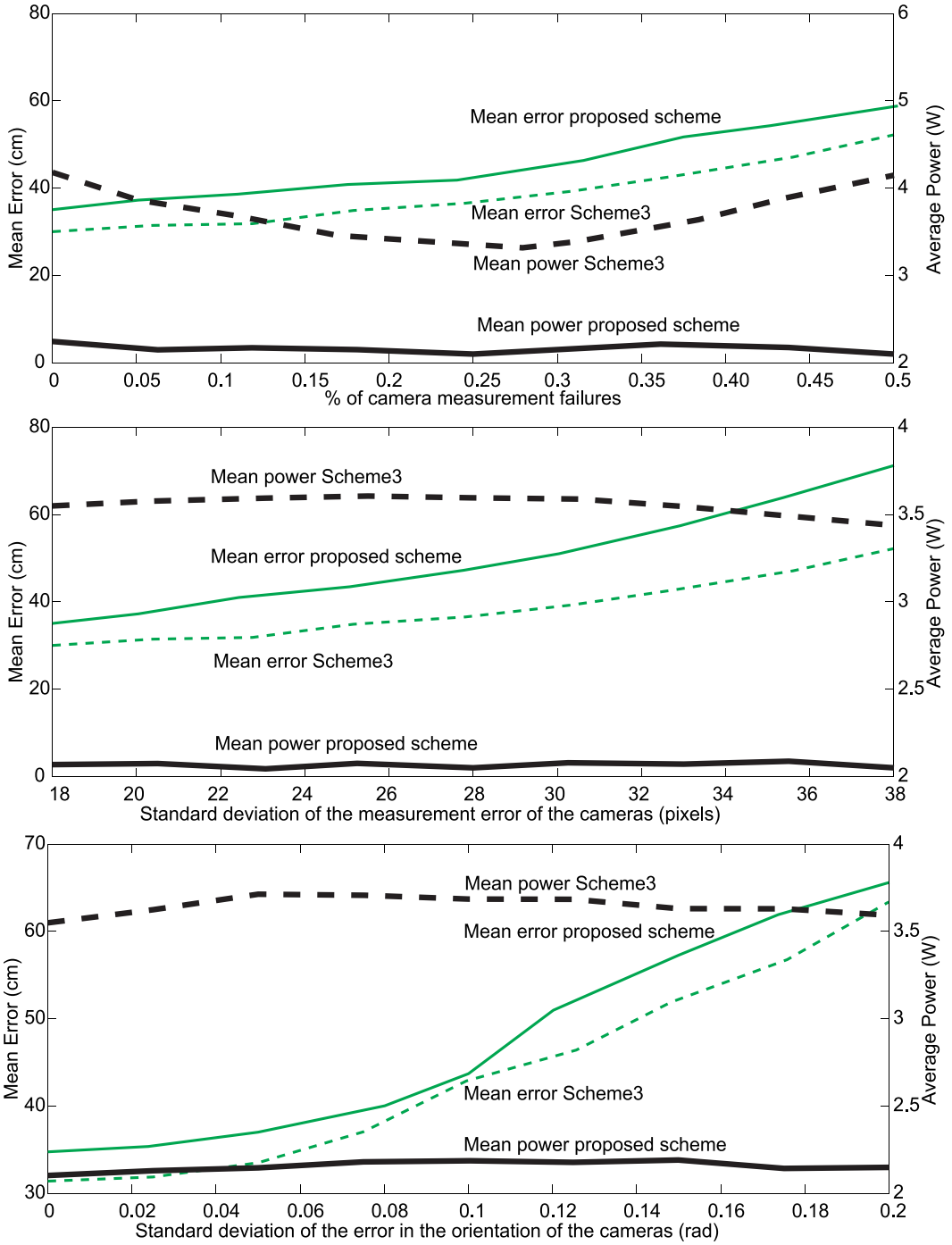


Fig. 16. Robustness comparison between *Scheme3* and the proposed scheme against camera failures (top), target segmentation errors (center), camera orientation errors (bottom). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Finally, Fig. 15 analyzes the impact of the greedy sensor activation. The proposed scheme has been compared to a scheme where the activation/deactivation decisions have been taken to globally optimize the sum of J along the full experiment. The rest of the components were exactly the same in both cases. 100 experiments with different target trajectories and different settings – each with 7 camera nodes – were performed. The experiments were carried out registering all the measurements and then, executing both schemes offline. The scheme with the globally optimal activation/deactivation was executed using the brute force approach. Fig. 15 shows that the difference in performance was very low. The proposed scheme with greedy activation/deactivation consumed only 4% more energy and its mean error was only 3.5% higher than with the globally optimal activation/deactivation. Of course, the globally optimal activation/deactivation cannot be executed in real time in COTS nodes and scales badly with the problem size.

7.3. Robustness analyses

Next we analyze the robustness of the proposed scheme against the most common sources of error in cameras: camera failures, camera orientation errors and errors in target segmentation in the

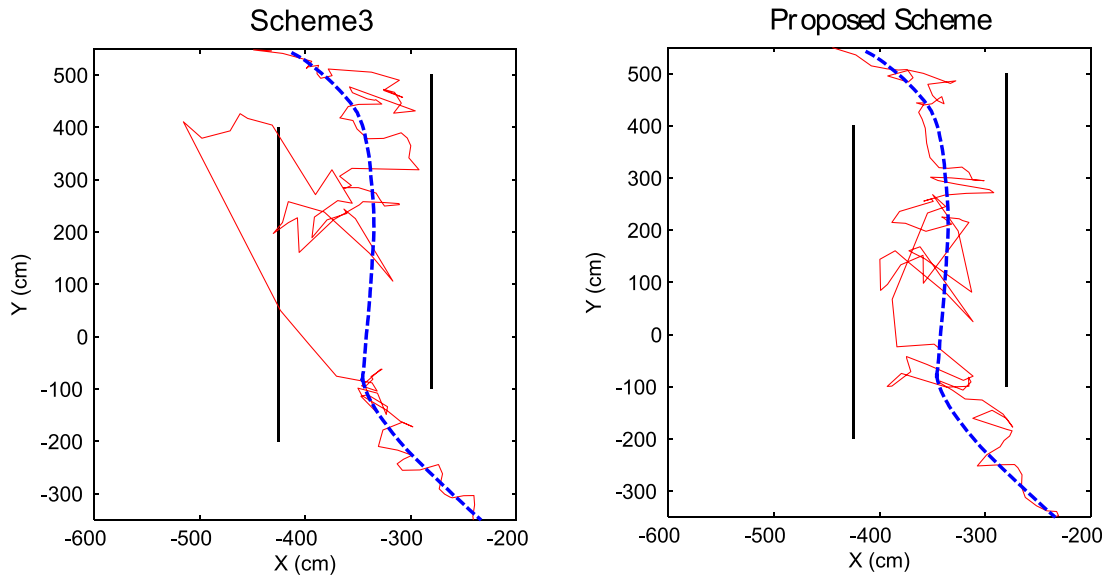


Fig. 17. Robustness against occlusions between *Scheme3* and the proposed scheme. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

images. Fig. 16 compares the mean error (blue color) and energy consumed (red) by the proposed scheme (solid line) and the aforementioned *Scheme3* (dashed), which uses only cameras.

Fig. 16 (top) analyzes the robustness of the proposed scheme and of *Scheme3* assuming that each camera can fail randomly with a probability in the range [0, 50] %. The mean tracking error behaves similarly in the proposed scheme and in *Scheme3*. Our scheme consumes an average of 41% lower than *Scheme3*. Fig. 16(center) analyzes robustness against errors in image segmentation. In preliminary tests we noticed that the measured center of the object had errors with a standard deviation of 18 pixels. We added additional random errors so that the total error had standard deviations in the range [18, 38] pixels. Both schemes had similar performance with low image segmentation error level but our scheme performed 15% worse with high error level. Again, the energy consumption of the proposed scheme was 42% lower than that of *Scheme3*. Fig. 16(bottom) analyzes robustness assuming that the orientation angles of cameras (roll, pitch and yaw) contain random errors with standard deviations in the range [0, 0.2] rad. The mean error increased similarly in both schemes and the energy consumption in the proposed scheme was 40% lower than in *Scheme3*.

One advantage of the proposed scheme is robustness against camera occlusions. Of course, any system solely based on camera measurements will never be robust to complete occlusions for relatively large areas. The proposed scheme handles occlusions by integrating RSSI measurements. We repeated the experiments deploying two walls in the environment such that no camera could see the target during an interval in its path. These walls are represented in Fig. 9 with dashed lines. Fig. 17 shows the estimated path (red line) and ground truth (blue). During the occlusion the lack of camera measurements prevented the EIF in *Scheme3* from updating the state vector, degrading its accuracy – mean error of 176 cm. The proposed scheme had a mean error of 65 cm. Occlusions originated high uncertainties in the RSSI training method and the default RSSI-range model was preferred by most camera nodes. The integration of RSSI measurements even using the default model was a significant advantage when no camera measurements were available.

The proposed scheme, consuming 40% less, is as robust as *Scheme3* against camera errors and significantly more robust to target occlusions.

8. Conclusions

Research in WCN-based systems for cluster-based target localization and tracking in ubiquitous computing environments have attracted high interest in the recent years. Our scheme assumes that targets are tagged with emitting nodes. This is actually the case in many applications in which people carrying smartphones and portable computing devices – Wi-Fi emitting nodes – are tracked in environments endowed with Wi-Fi cameras – wireless camera nodes.

Most WCN node models can measure the RSSI of an incoming packet with negligible delay, energy consumption and computational cost. The interactions of the radio signal with the environment – particularly indoors – originate disturbances that are very difficult to reflect in RSSI-range models. Thus, most RSSI-based localization and tracking systems use RSSI-range models that assume free-space radio transmission, which result in poor localization and tracking performance. This is the main reason why most WCN-based tracking systems rely exclusively on camera measurements. Very few WCN-based tracking systems combine RSSI with camera measurements and these systems use RSSI measurements only when the camera measurements are not available, for instance in case of target occlusions.

Cluster-based tracking schemes requires methods for including/excluding nodes in the cluster, which has direct impact on nodes energy consumption. Existing WCN-based tracking systems include nodes using criteria based on distance or depending if the target is within the camera field of view or not. In case of nodes equipped with different sensors, these node inclusion/exclusion methods do not have enough flexibility to fully exploit the synergies between the different types of measurements. Besides, low-cost constraints of COTS WCN nodes impose critical energy, bandwidth and computational efficiency requirements. In most existing WCN-based tracking systems measurement integration is computed by the cluster head resulting in inhomogeneous resource consumption and lack of scalability.

This paper proposes an scheme to exploit synergies between RSSI and camera measurements for localization and tracking. Our objective is to give a response to the three aforementioned issues – lack of precise RSSI-range models, sensor activation/deactivation and measurement integration – while reducing the consumption of the resources available at WCN nodes: energy in order to

enlarge battery lifetime and computational effort in order to enable the use of simpler and cheaper WCN nodes.

The scheme is based on two main components: a training method in which each node dynamically adapts its RSSI-range model considering the target current location; and a sensor selection method that activates/deactivates individual sensors of each camera node balancing the different accuracies and energy consumptions of camera and RSSI measurements. It also encompasses a distributed EIF for the integration of heterogeneous measurements. The joint use of these components generates a self-regulated behavior in which the camera measurements are used to train RSSI-range models and once trained, the sensor selection method tends to deactivate cameras due to their higher consumption. As a result the proposed scheme strongly reduces energy consumption – 40% – with almost no performance degradation w.r.t. schemes solely based on cameras. Besides, the experiments confirmed that its performance and robustness are comparable to methods based only on cameras but our method is significantly more robust to target occlusions.

The association between visual and RSSI measurements is the scope of this paper. Reported measurement association methods or combination of them could be used. Voting methods based on local associations performed using the minimum Mahalanobis distance seems a good approach in our scheme. Even without using visual-radio data association the proposed scheme can be useful: (1) if the target has visual markers or can be identified using computer vision, e.g. face recognition, in that case a visual identifier can be associated to the radio identifier; and (2) in applications where few targets can be in the environment at the same time, e.g. security in restricted areas.

In this work the costs taken for the sensor selection method were static. It can be interesting to make them dependent on the number of currently active nodes in order to avoid a big tracking cluster. The extension of camera-based sensor training to others sensors is object of current research.

Acknowledgment

This work was partially supported by projects AEROARMS, funded by the European Commission under H2020-EU.2.1.1.5 with Project Reference 644271 and “SILCAE: Integrated system for identification, localization and monitoring of personnel in working centers”, funded by CLEVER S.L. J.R. Martinez-de Dios would like to thank project ARCOW funded by the European Commission. The authors would like to express their gratitude to Prof. Ruben Martin-Clemente and to Ph.D. Arturo Torres-Gonzlez for their help in the analysis and response to reviewers comments.

References

[1] A. Jiménez-González, J.R. Martínez-de Dios, A. Ollero, An integrated testbed for cooperative perception with heterogeneous mobile and static sensors, Sensors 11 (12) (2011) 11516–11543, doi:10.3390/s111211516.
[2] W.-P. Chen, J.C. Hou, L. Sha, Dynamic clustering for acoustic target tracking in wireless sensor networks, IEEE Trans. Mob. Comput. 3 (3) (2004) 258–271, doi:10.1109/TMC.2004.22.
[3] J. Shin, L.J. Guibas, F. Zhao, A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks, in: Proceedings of the International Conference on Information Processing in Sensor Networks, Springer, 2003, pp. 223–238, doi:10.1007/3-540-36978-3_15.
[4] H. Zha, J. Metzner, G. Kesidis, et al., Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks, in: Proceedings of the 2004 IEEE International Conference on Communications, 7, IEEE, 2004, pp. 3807–3811, doi:10.1109/ICC.2004.1313265.
[5] A. de San Bernabe, J.R. Martínez-de Dios, A. Ollero, Mechanisms for efficient integration of RSSI in localization and tracking with wireless camera networks, in: Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2013, pp. 390–397, doi:10.1109/IROS.2013.6696381.
[6] R. Want, A. Hopper, V. Falcão, J. Gibbons, The active badge location system, ACM Trans. Inf. Syst. 10 (1) (1992) 91–102, doi:10.1145/128756.128759.
[7] E. Aitenbichler, M. Muhlhauser, An IR local positioning system for smart items and devices, in: Proceedings of the Twenty Third International Conference on Distributed Computing Systems, 2003, pp. 334–339, doi:10.1109/ICDCSW.2003.1203576.
[8] A. Ward, A. Jones, A. Hopper, A new location technique for the active office, IEEE Pers. Commun. 4 (5) (1997) 42–47, doi:10.1109/98.626982.
[9] H. Piontek, M. Seyffer, J. Kaiser, Improving the accuracy of ultrasound-based localisation systems, Pers. Ubiquitous Comput. 11 (6) (2007) 439–449, doi:10.1007/s00779-006-0096-1.
[10] P. Bahl, V.N. Padmanabhan, Radar: an in-building RF-based user location and tracking system, in: Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM, 2000, pp. 775–784, doi:10.1109/INFCOM.2000.832252.
[11] S. Ingram, D. Harmer, M. Quinlan, Ultra wideband indoor positioning systems and their use in emergencies, in: Proceedings of the Position Location and Navigation Symposium, PLANS, 2004, pp. 706–715, doi:10.1109/PLANS.2004.1309063.
[12] F. Raab, E. Blood, T. Steiner, H. Jones, Magnetic position and orientation tracking system, IEEE Trans. Aerosp. Electron. Syst. 15 (5) (1979) 709–718, doi:10.1109/TAES.1979.308860.
[13] P. Kumar, L. Reddy, S. Varma, Distance measurement and error estimation scheme for RSSI based localization in wireless sensor networks, in: Proceedings of the Fifth Conference on Wireless Communication and Sensor Networks (WCSN), 2009, pp. 1–4, doi:10.1109/WCSN.2009.5434802.
[14] I. Guvenc, C.-C. Chong, A survey on TOA based wireless localization and NLOS mitigation techniques, IEEE Commun. Surv. Tutor. 11 (3) (2009) 107–124, doi:10.1109/SURV.2009.090308.
[15] N.B. Priyantha, A.K. Miu, H. Balakrishnan, S. Teller, The cricket compass for context-aware mobile applications, in: Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking, ACM, 2001, pp. 1–14, doi:10.1145/381677.381679.
[16] D. Niculescu, B. Nath, Ad hoc positioning system (APS) using AOA, in: Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, INFOCOM 2003, 3, IEEE, 2003, pp. 1734–1743, doi:10.1109/INFCOM.2003.1209196.
[17] J. Fernández-Berni, R.A. Carmona-Galán, A. Rodríguez-Vázquez, et al., Low-Power Smart Imagers for Vision-Enabled Sensor Networks, Springer, 2012.
[18] H. Medeiros, J. Park, A. Kak, Distributed object tracking using a cluster-based Kalman filter in wireless camera networks, IEEE J. Sel. Top. Signal Process. 2 (4) (2008) 448–463, doi:10.1109/JSTSP.2008.2001310.
[19] R. Goshorn, J. Goshorn, D. Goshorn, H. Aghajan, Architecture for cluster-based automated surveillance network for detecting and tracking multiple persons, in: Proceedings of the First ACM/IEEE International Conference on Distributed Smart Cameras, ICDSC'07, IEEE, 2007, pp. 219–226, doi:10.1109/ICDSC.2007.4357527.
[20] A.O. Ercan, A. El Gamal, L.J. Guibas, Object tracking in the presence of occlusions via a camera network, in: Proceedings of the Sixth International Conference on Information Processing in Sensor Networks, 2007, pp. 509–518, doi:10.1145/1236360.1236425.
[21] B. Song, A.K. Roy-Chowdhury, Stochastic adaptive tracking in a camera network, in: Proceedings of the Eleventh IEEE International Conference on Computer Vision, ICCV'2007, 2007, pp. 1–8, doi:10.1109/ICCV.2007.4408937.
[22] A. de San Bernabe, J.R. Martinez-de Dios, A. Ollero, Efficient cluster-based tracking mechanisms for camera-based wireless sensor networks, IEEE Trans. Mob. Comput. 14 (9) (2015) 1820–1832, doi:10.1109/TMC.2014.2374164.
[23] C. Soto, B. Song, A.K. Roy-Chowdhury, Distributed multi-target tracking in a self-configuring camera network, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, CVPR'2009, 2009, pp. 1486–1493, doi:10.1109/CVPR.2009.5206773.
[24] X. Wang, O. Bischoff, R. Laur, S. Paul, Localization in wireless ad-hoc sensor networks using multilateration with RSSI for logistic applications, Procedia Chem. 1 (1) (2009) 461–464, doi:10.1016/j.proche.2009.07.115.
[25] A. Savvides, C.-C. Han, M.B. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, in: Proceedings of the Seventh International Conference on Mobile Computing and Networking, 2001, 2001, pp. 166–179, doi:10.1145/381677.381693.
[26] C. Liu, K. Wu, T. He, Sensor localization with ring overlapping based on comparison of received signal strength indicator, in: Proceedings of the International Conference on Mobile Ad-hoc and Sensor Systems, 2004, pp. 516–518, doi:10.1109/MAHSS.2004.1392193.
[27] T. He, C. Huang, B.M. Blum, J.A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: Proceedings of the Ninth International Conference on Mobile Computing and Networking, 2003, pp. 81–95, doi:10.1145/938985.938995.
[28] V. Honkavirta, T. Perala, S. Ali-Loytty, R. Piché, A comparative survey of WLAN location fingerprinting methods, in: Proceedings of the Sixth Workshop on Positioning, Navigation and Communication, 2009, pp. 243–251, doi:10.1109/WPNC.2009.4907834.
[29] A. De San Bernabé, J.R. Martínez-de Dios, A.O. Baturone, A WSN-based tool for urban and industrial fire-fighting, Sensors 12 (11) (2012) 15009–15035, doi:10.3390/s121115009.
[30] P. Desai, K.S. Rattan, Indoor localization and surveillance using wireless sensor network and Pan/Tilt camera, in: Proceedings of the IEEE 2009 National Aerospace and Electronics Conference (NAECON), 2009, pp. 1–6, doi:10.1109/NAECON.2009.5426659.

[31] T. Miyaki, T. Yamasaki, K. Aizawa, Multi-sensor fusion tracking using visual information and Wi-Fi location estimation, in: Proceedings of the First International Conference on Distributed Smart Cameras, ICDSC'07, 2007a, pp. 275–282, doi:10.1109/ICDSC.2007.4357534.
[32] T. Miyaki, T. Yamasaki, K. Aizawa, Tracking persons using particle filter fusing visual and Wi-Fi localizations for widely distributed camera, in: Proceedings of the International Conference on Image Processing, ICIP 2007, 3, 2007b, pp. 225–228, doi:10.1109/ICIP.2007.4379287.
[33] A. Dore, A.F. Cattoni, C.S. Regazzoni, A particle filter based fusion framework for video-radio tracking in smart spaces, in: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS'2007, 2007, pp. 99–104, doi:10.1109/AVSS.2007.4425293.
[34] W. Zhang, G. Cao, DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks, Trans. Wirel. Commun. 3 (5) (2004) 1689–1701, doi:10.1109/TWC.2004.833443.
[35] A. Ercan, D. Yang, A. El Gamal, L. Guibas, Optimal placement and selection of camera network nodes for target localization, Distrib. Comput. Sens. Syst. (2006) 389–404, doi:10.1007/11776178_24.
[36] F. Zhao, J. Shin, J. Reich, Global node selection for localization in a distributed sensor network, IEEE Trans. Aerosp. Electron. Syst. 42 (2006a) 113–135, doi:10.1109/TAES.2006.1603409.
[37] F. Zhao, J. Shin, J. Reich, Local node selection for localization in a distributed sensor network, IEEE Trans. Aerosp. Electron. Syst. 42 (2006b) 136–146, doi:10.1109/TAES.2006.1603410.
[38] F. Zhao, J. Shin, J. Reich, Information-driven dynamic sensor collaboration, IEEE Signal Process. Mag. (2002) 61–72, doi:10.1109/79.985685.
[39] P. Pahalawatta, P. Pappas, A. Katsaggelos, Optimal sensor selection for video-based target tracking in a wireless sensor network, in: Proceedings of the International Conference on Image Processing (ICIP 2004), 2004, doi:10.1109/ICIP.2004.1421762.
[40] H. Wang, K. Yao, G. Pottie, D. Estrin, Entropy-based sensor selection heuristic for target localization, in: Proceedings of the Third International Symposium on Information Processing in Sensor Networks, 2004.
[41] E. Ertin, J. Fisher, L. Potte, Maximum mutual information principle for dynamic sensor query problems, in: Proceedings of the Second International Workshop on Information Processing in Sensor Networks (IPSN 2003), 2003, doi:10.1007/3-540-36978-3_27.
[42] X. Shen, S. Liu, P.K. Varshney, Sensor selection for nonlinear systems in large sensor networks, IEEE Trans. Aerosp. Electron. Syst. 50 (4) (2014) 2664–2678, doi:10.1109/TAES.2014.130455.
[43] T. Onel, C. Ersoy, H. Delic, Information content-based sensor selection and transmission power adjustment for collaborative target tracking, IEEE Trans. Mob. Comput. 8 (8) (2009) 1103–1116, doi:10.1109/TMC.2009.12.
[44] A. Smailagic, D. Kogan, Location sensing and privacy in a context-aware computing environment, IEEE Wirel. Commun. 9 (5) (2002) 10–17, doi:10.1109/MWC.2002.1043849.
[45] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, J. Sievänen, A probabilistic approach to WLAN user location estimation, Int. J. Wirel. Inf. Netw. 9 (3) (2002) 155–164, doi:10.1023/A:1016003126882.
[46] F. Vanheel, J. Verhaevert, E. Laermans, I. Moerman, P. Demeester, Automated linear regression tools improve RSSI WSN localization in multipath indoor environment, EURASIP J. Wirel. Commun. Netw. 2011 (1) (2011) 1–27, doi:10.1186/1687-1499-2011-38.
[47] S. Thrun, W. Burgard, D. Fox, et al., Probabilistic Robotics, 1, MIT Press, Cambridge, MA, 2005.
[48] L. Mihaylova, D. Angelova, D. Bull, N. Canagarajah, Localization of mobile nodes in wireless networks with correlated in time measurement noise, IEEE Trans. Mob. Comput. 10 (1) (2011) 44–53, doi:10.1109/TMC.2010.132.
[49] M.C. Vuran, Ö. B. Akan, I.F. Akyildiz, Spatio-temporal correlation: theory and applications for wireless sensor networks, Comput. Netw. 45 (3) (2004) 245–259, doi:10.1016/j.comnet.2004.03.007.
[50] S. Liu, E. Masazade, M. Fardad, P.K. Varshney, Sensor selection with correlated measurements for target tracking in wireless sensor networks, in: Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2015, pp. 4030–4034, doi:10.1109/ICASSP.2015.7178728.
[51] D. Fox, W. Burgard, F. Dellaert, S. Thrun, Monte Carlo localization: efficient position estimation for mobile robots, in: Proceedings of the National Conference on Artificial Intelligence, 1999, pp. 343–349. 10.1.1.2.342
[52] J. Heikkila, O. Silven, A four-step camera calibration procedure with implicit image correction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition., 1997, pp. 1106–1112, doi:10.1109/CVPR.1997.609468.
[53] A. Rowe, Cmucam3 datasheet, 2012.
[54] M. Corporation, Telos (rev b) datasheet, 2004.
[55] M. Ben Kilani, A.J. Raymond, F. Gagnon, G. Gagnon, P. Lavoie, RSSI-based indoor tracking using the extended Kalman filter and circularly polarized antennas, in: Proceedings of the 2014 Eleventh Workshop on Positioning, Navigation and Communication (WPNC), IEEE, 2014, pp. 1–6, doi:10.1109/WPNC.2014.6843305.
[56] A. de San Bernabe, J.R Martinez-de Dios, A. Ollero, Entropy-aware cluster-based object tracking for camera wireless sensor networks, in: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 3985–3992, doi:10.1109/IROS.2012.6385805.