2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Mechanisms for efficient integration of RSSI in localization and tracking with Wireless Camera Networks

A. de San Bernabé, J.R. Martinez-de Dios, A. Ollero

Abstract— **This paper proposes a scheme that exploits synergies between RSSI and camera measurements in object localization and tracking using Wireless Camera Networks (WCN). It is based on three main mechanisms: a training method that accurately adapts RSSI-range models to the particular environment; a sensor activation/deactivation method that balances the different information contribution and energy consumptions of camera and RSSI measurements; and a distributed Information Filter to integrate the available measurements. The joint use of these mechanisms drastically reduces energy consumption -40%- with no significant degradation w.r.t. existing schemes based on only cameras and shows better robustness to target occlusions. The scheme has been implemented and validated in the indoor *CONET Integrated Testbed*.**

I. Introduction

In the last decade the re-configurability and ease of deployment of Wireless Sensor Networks (WSN) has motivated intense research in applications such as localization and tracking in GPS-denied environments. Wireless Camera Networks (WCN) have gained increasing interest. They can implement multi-camera perception schemes typical of standard camera networks with the flexibility and re-configurability of WSN.

Various tracking methods for WCNs have been developed. However, the great majority of them rely only on cameras ignoring Radio Signal Strength Indicator (RSSI). Fusion of bearing and range measurements -like RSSI- can originate interesting synergies. Both individually provide partial observation but combining one measurement of each type suffices to constrain a location. Moreover, RSSI can be used when the object is occluded in the images or is out of the cameras field of view. Despite these advantages very few methods integrating cameras and RSSI have been developed even taking into account that RSSI can be measured by almost all nodes with negligible additional energy or computer cost. Many RSSI-based localization and tracking methods have been developed. However, RSSI is highly affected by radio reflections and other interactions with the environment, frequently resulting in poor performance.

This paper proposes a new scheme that exploits synergies between RSSI and camera measurements in localization and tracking. It is based on the combination of three main mechanisms: 1) a distributed Extended Information Filter (EIF) to efficiently integrate available -camera and/or RSSI- measurements in the same probabilistic frame; 2) a training method that adapts RSSI-range models to the particular environment and radio reflections and interactions; and 3) a sensor activation/deactivation method that balances the accurate sensing -but high energy consumption- of cameras with RSSI-based localization, which is more energy efficient. The interaction between these mechanisms allows tracking targets with similar errors than schemes based only on cameras but with drastic energy savings, allowing 40% longer batteries lifetimes. The scheme has been validated in the indoor *CONET Integrated Testbed* [1] showing good robustness against target occlusions and camera errors.

The paper is organized as follows. Related work is in the next subsection. The general description of the proposed scheme is in Section II. The distributed EIF, the RSSI-range model training and the sensor activation/deactivation mechanisms are described in Sections III, IV and V, respectively. Section VI presents experimental results and performance and robustness analysis. Conclusions is the final section.

A. Related Work

Most existing localization and tracking methods for WCNs rely on integrating measurements only from cameras using tools such as Kalman Filters [2], Particle Filters [3] and Information Filters, which are more efficient than Kalman Filters in case of a large number of measurements [4].

A high number of RSSI-based localization methods have been developed. Range-based methods, such as multilateration [5] or least squares [6], use RSSI measurements to estimate distance to anchor nodes. Reflections and other interactions with the environment make RSSI models very dependent on the setting, making them unpredictable. *Range-free* methods, such as ROC-RSSI [7] or APIT [8], avoid these drawbacks by relying on geometric considerations. However, their accuracies are usually poorer. Another approach is to learn RSSI characteristics from the environment. Fingerprinting methods, see e.g. [9], compare measurements with a previously obtained RSSI map. They require accurate RSSI maps of the environment and should re-obtain the map if the setting changes. In [10] 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 complementarities. However, the number of Wireless Camera Networks that use RSSI is very low. Miyaki et al. proposed to estimate target location individually using cameras and using RSSI and then, to integrate both estimates using a

This work was supported by the PLANET (European Commission FP7-257649-ICT-2009-5) and CLEAR (DPI2011-28937-C02-01). J.R. Martinez-de Dios thanks project EC-SAFEMOBILE (European Commission ICT-2011-288082). The authors are with the Robotics, Vision and Control Group, University of Seville; Avda. de los Descubrimientos, sn; Seville; Spain adesanbernabe@us.es, jdedios@cartuja.us.es and aollero@cartuja.us.es

sensor fusion method [11] or a Particle Filter [12]. These methods do not fully exploit synergies of bearing and range measurements. Moreover, they were validated in outdoor tests but accuracy and robustness were not analyzed.

Camera nodes tracking a target are usually organized dynamically in clusters triggered by external events. Tracking a moving target requires dynamically including and excluding nodes in the cluster: nodes out of the cluster can be switched-off, saving energy. Some tracking methods, such as [13], use distance to the target as main node activation criterion. However, proximity does not imply that the camera is capable to acquire information about the target. Most methods activate one camera when the object is estimated to enter its field of view and deactivate the camera when it comes out [2]. This criterion leads to keeping active a high number of cameras, many of which can be unnecessary unless the camera deployment has been optimized [14].

II. General description

Assume that a number of static nodes each equipped with one camera, from now on camera nodes, have been deployed in an environment. Camera nodes are assumed to have sufficient computer capacities to execute simple image processing methods: each can measure the coordinates of the target center in its image plane. Also, each camera node can measure the RSSI of the messages received from the target, assumed tagged, e.g. with a WSN node. The proposed method adopts a semi-decentralized scheme based on clustering. Camera nodes sensing one target organize in a cluster. Thus, several simultaneous targets can be tracked, each with its cluster. Each cluster has one cluster head -in our case the mobile node- that executes the tracking method. Thus, the message flow is kept within the cluster at the surroundings of the target, simplifying transmissions.

This paper proposes a new scheme that employs probabilistic and training mechanisms to exploit synergies between RSSI and camera measurements, see Fig. 1. Lack of accuracy of RSSI models is the main drawback when using RSSI measurements. The proposed scheme includes a mechanism in which each node uses target locations estimated by cameras to train its own RSSI-range model adapted to the local environment surrounding the node and the target. Training is performed dynamically to adapt to the target motion. Thus, it is capable of learning multi-path reflections of the radio signal even in indoors scenarios. In very complex scenarios the accuracy of the trained RSSI is monitored, disabling the mechanism if necessary. As a result, when camera and RSSI measurements are available, this mechanism significantly improves the RSSI accuracy and makes it robust to reflections and interactions with the environment, see Section IV.

Our scheme also includes a mechanism for activation/deactivation of sensors balancing cost (energy) and reward (information gain), see Section V. Its objective is to limit the number of active sensors to those essential reducing energy consumption. This mechanism tends to activate nodes with trained RSSI models. Finally, a distributed *Information Filter* (IF), see Section III, is used to integrate the available camera and/or RSSI measurements.

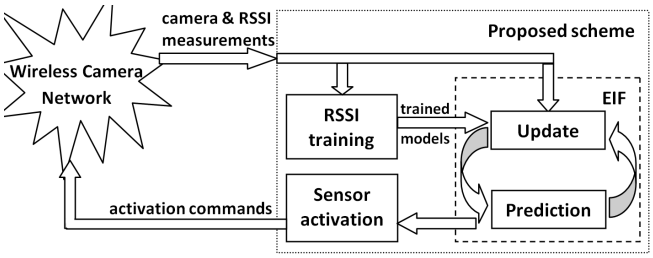


Fig. 1. Schematic illustration of the proposed scheme.

The combination of these mechanisms originates a self-regulated feedback effect: camera measurements are used to calibrate RSSI models and once calibrated, the sensor selection method tends to deactivate cameras due to their higher consumption. Thus, the proposed scheme highly reduces energy consumptions with similar accuracies than schemes based only on cameras. It has superior robustness against target occlusions and similar robustness to camera errors.

III. Measurement integration using Information Filters

Recursive Bayesian Filters (RBFs) are well-founded tools for sensor integration assuming the prediction and measurement models are subject to noise. Information Filters (IFs) are parametric RBFs that employ the so-called canonical representation consisting of an information vector $\xi = \Sigma^{-1}\mu$ and matrix $\Omega = \Sigma^{-1}$. Dual of Kalman Filters (KFs), the update stage of IFs is more efficient than that of KFs. That makes them more convenient in cases with a simple prediction model and high number of measurements, as in our problem, in which many inexpensive camera nodes can be used. Besides, IFs are numerically more stable and more suitable for representing lack of information.

We adopted a state vector q_k typical in localization and tracking problems with 6 components: 3D current target location and 3D target current velocities. More complex models require a priori knowledge, which is often unavailable.

One observation model is required for each type of measurement. The camera observation model is derived from the pin-hole model. Assume that P_k is the location of the target in the global reference frame G at time k. Assume that $p_{i,k}$ is the projection of the target on the image plane of camera node i expressed in the local reference frame of camera, F_i, related to G by transformation matrix T_i. The following observation model for camera i holds:

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

where $t_{i,j}$ is the $j-th$ row of T_i. This observation model is nonlinear. Its linearization leads to the *Extended Information Filter* (EIF), which uses the Jacobian of hc_i, $HC_{i,k}$, at the EIF update stage, see [15].

Our method includes a mechanism in which each node trains its own RSSI model. If the training RSSI model mechanism is active the EIF uses the model trained as described in Section IV. Otherwise, the EIF uses the following widely accepted model [16] -we will call it default RSSI model:

$$RSSI_{i,k} = hr_i(P_k) = a \log d_{i,k} + b, \tag{2}$$

where $d_{i,k}$ is the distance between the target and node i at time k and a and b are model parameters. The default RSSI model is given by a_D, b_D and the model variance σ_D^2. This default model, taken from works such as [16], is not suitable for indoors and does not address multipath reflections. However, it is used very scarcely, only when the training-based RSSI-range model is not available or not accurate, see Section IV. The RSSI model in (2) is nonlinear and the EIF uses its Jacobian:

$$Hr_{i,k} = \frac{\partial hr_i}{\partial d_{i,k}} \frac{\partial d_{i,k}}{\partial q_k} \tag{3}$$

For robustness and scalability the proposed EIF uses a distributed implementation. At each time k the cluster head executes the EIF prediction stage computing $\overline{\Omega}_k$, $\overline{\xi}_k$ and $\overline{\mu}_k$ and broadcasts $\overline{\mu}_k$ within the cluster. Each node i in the cluster takes measurements from its active sensors, integrates its local measurements computing $\Omega_{i,k}$ and $\xi_{i,k}$ -its contribution to the overall EIF update stage- and transmits $\Omega_{i,k}$ and $\xi_{i,k}$ to the cluster head. Then, the cluster head computes Ω_k summing $\overline{\Omega}_k$ and the contribution of each node $\Omega_{i,k}$. It also computes ξ_k summing $\overline{\xi}_k$ and the contribution of each node $\xi_{i,k}$. The computing time of the distributed EIF is roughly constant regardless the number of cluster nodes. RSSI between the target (cluster head) and the nodes is measured from the messages they interchange without additional energy consumption, computer burden or delays.

IV. RSSI-Range Model Training

When measuring the RSSI between two nodes the local environment around both nodes critically affects the observation model. In fact, if we make RSSI measurements without changing their local environments, we can notice that all the values are very similar: it is reflections and other interactions with the environment and not measurement errors themselves what originate RSSI difficulties for localization and tracking. Our method includes a mechanism in which each node uses the current target location to dynamically adapt its own RSSI-range model to the particularities of the environment. The resulting trained models are adapted to the local environments of the emitter and receiver -and its reflections and interactions with the environment- and present good accuracies making them suitable for tracking problems.

Let P_k be the target location at time k. Let $RSSI_{i,k}$ be the RSSI measured by node i from the messages received from the target. Assuming each node knows its location, node i can collect a set of the last M measurements $\{(RSSI_{i,j}, d_{i,j})\}$ $j = k-M+1 : k$, where $d_{i,j}$ is the distance between the node i and P_j. The objective is to obtain a RSSI-range model for node i adapted to the local environment of the target at time k. These local models can be assumed linear, $RSSI_{i,k} = a_i d_{i,k} + b_i$, and can be efficiently trained using a simple linear regression:

$$a_i = \frac{\sum_j RSSI_{i,j} d_{i,j} - \overline{RSSI_i} \sum_j d_{i,j}}{\sum_j (d_{i,j})^2 - \overline{d_i} \sum_j d_{i,j}}, \tag{4}$$

$$b_i = \overline{RSSI_i} - (a\overline{d_i}), \tag{5}$$

where $\overline{rssi_i}$ and $\overline{d_i}$ stand for the mean of $RSSI_{i,j}$ and $d_{i,j}$. To cope with the target motion, node i trains its model with the last M pairs collected. M is typically selected with low values. If the target moves, high M can involve RSSI measurements with different target local environments.

Of course, in our system P_k, the actual location of the target, is not known. Instead, we use estimates of the target location obtained by an auxiliary EIF as that presented in Section III but that only integrates camera measurements. If not addressed carefully this approach may introduce cameras inaccuracies into the RSSI training.

Assume $d'_{i,k} = d_{i,k} + u_i$, 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. Assume that $RSSI_{i,k} = a_i d_{i,k} + b_i$ 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 the following expression holds:

$$RSSI'_{i,k} = a_i(d_{i,k}+u_i)+b_i+v_i = RSSI_{i,k}+a_i u_i+v_i \tag{6}$$

Assuming 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 trained model can be expressed as:

$$\sigma^2_{tm,i} = a_i^2 \sigma^2_{u_i} + \sigma^2_{v_i} \tag{7}$$

The variance of the trained RSSI 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 models with higher a are more sensitive to target location errors. If a trained model has $1/a = 0$, it is approximated by $1/a = \epsilon$ and b is recomputed using (5) with the new a.

Figure 2-top shows the default RSSI model computed by fitting (2) with RSSI measurements (in red color) from every pair of nodes deployed in the *CONET Integrated Testbed*. Figure 2-bottom shows two trained RSSI models for node i computed with measurements between node i and the target at two different times along its path. Differences in fitting error between both models are evident. Trained RSSI models are specific for each node and valid only for local surroundings around the target but they are significantly more accurate than the RSSI default model and is capable of capturing existing reflection and interactions with the environment.

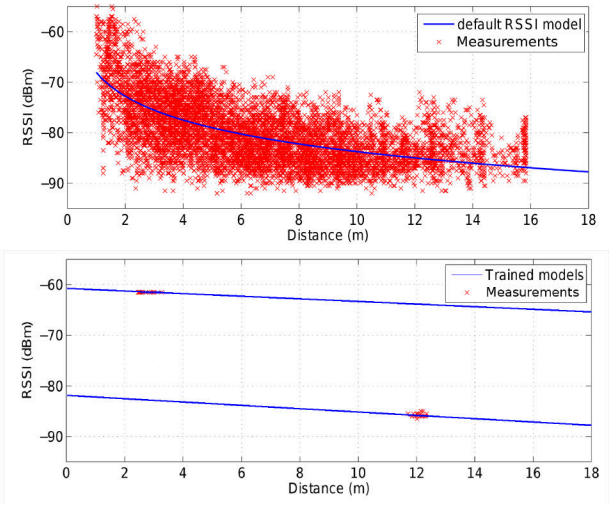


Fig. 2. RSSI models obtained in an experiment: top) default logarithmic model, bottom) trained linear models valid only locally at two times along the robot path.

In our training mechanism each node decides the RSSI model that should be used for integrating its measurements in the filter described in Section III. The auxiliary EIF estimates the target location and its covariance. Thus, each node computes $\sigma^2_{tm,i}$ using (7) to predict the accuracy of its trained model. For instance, it can decide to use the default RSSI model instead its trained RSSI model if $\sigma^2_{tm,i} > \sigma^2_D$, see Algorithm 1. Thus, the malfunctioning of the auxiliary EIF or badly trained RSSI models can be easily detected avoiding to include these errors in the trained RSSI model. The effects of errors in the RSSI training mechanism and its robustness to different types of camera errors are analyzed in Section VI.

With good target location estimates this mechanism allows RSSI measurements with accuracies comparable to camera measurements but spending significantly lower energy. This can be exploited when this mechanism is combined with sensor selection methods based on balancing energy VS information gain, as that described in Section V.

Algorithm 1 RSSI-range model training for node i

Require: $\{(RSSI_{i,j}, d_{i,j})\}, j = k - M + 1 : k$

$a_i = \frac{\sum_j RSSI_{i,j} d_{i,j} - \overline{RSSI_i} \sum_j d_{i,j}}{\sum_j (d_{i,j})^2 - \overline{d_i} \sum_j d_{i,j}}$

$b_i = \overline{RSSI_i} - (a\overline{d_i})$

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

if $\sigma^2_{tm,i} > \sigma^2_D$ **then**

    Use default RSSI model

else

    Use trained model a_i, b_i

end if

V. Sensor Activation/Deactivation

In this method the cluster head decides which sensors should be integrated in the target tracking. Nodes with non-selected sensors keep inactive saving energy.

Assume that each node sensor can be commanded with two actions: activate or deactivate. Let A be the universe of possible actions. Given a certain state q_k at time k, each action $a_k \in A$ involves a cost $c(q_k, a_k)$ and a reward on how the target state is updated, $r(q_k, a_k)$. We use a greedy approach, which aims to decide the next most suitable action, see Algorithm 2. POMDPs can consider longer term goals but scale badly and involve intense computer burden. At each time k the method selects action $a_k \in A$ balancing expected rewards and costs. The cost for action a_k is the difference in energy consumed before and after performing the action. The reward is the expected information gain about the target state caused by the new observations, which can be expressed as follows:

$$r(q_k, a_k) = H(q_k) - H(q^*_{k+1}), \tag{8}$$

where $H(q_k)$ is the entropy of $p(q_k)$, the prior state distribution and $H(q^*_{k+1})$ is the entropy of $p(q^*_{k+1})$, the predicted state distribution with the new measurements. Assuming Gaussian distributions, $H(q_k) = -0.5 \log(\Omega_k)$, allowing easy integration in the EIF described in Section III.

Let A^+ be the set of actions which reward improves its cost, i.e. $r(q_k, a_k) - c(q_k, a_k) > 0$. At each time k the most suitable action $\hat{a}_k$ is selected as:

$$\hat{a}_k = \arg\max_{a_k \in A^+} (r(q_k, a_k) - c(q_k, a_k)) \tag{9}$$

If A^+ is empty, no action is performed. The method selects $\hat{a}_k$ without having the new measurements but using expectations of future information gain. However, it can largely reduce the energy consumption with no significant accuracy degradation. We use two variations of the method.

A. Full camera node activation

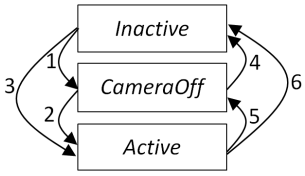
Each camera node can be *Active* or *Inactive*. When *Active* the node produces camera and RSSI measurements while in the *Inactive* mode the camera node is off. Only one action is possible for each node: to activate it if the node is currently inactive, or to deactivate it if it is active. The reward when activating a node is the information gain obtained by the new camera and RSSI measurements. The cost is $c1 + c2$, being $c1$ and $c2$ the energies consumed respectively by the node and by its camera module. In some cases it is interesting to consider the costs with different values depending on the resources currently available for the node. For simplicity we assume that the costs are the same for all camera nodes.

B. Sensor activation

In this case each camera node can be in three possible modes: *Active*, *Inactive* and *CameraOff*. In the *CameraOff* mode the node produces RSSI measurements but the camera module is off. This scheme is motivated by the high differences in the energy consumed by cameras and nodes.

Figure 3-left depicts the mode transition model adopted. Two possible actions can be performed by a node at each mode. A node at mode *CameraOff* can be commanded to

activate its camera ($Action2$) or can be fully deactivated ($Action4$). Each action involves different costs and rewards. Figure 3-right shows the cost of each action using $c1$ and $c2$ as defined above. Costs when deactivating are negative meaning energy savings. As in the full camera node activation scheme, costs are considered the same for all the nodes.



Action	Cost
1	c1
2	c2
3	c1+c2
4	-c1
5	-c2
6	-c1-c2

Fig. 3. Adopted transition model and action costs.

This approach allows more flexibility and can better use the RSSI training mechanism. Nodes in *CameraOff* mode with trained RSSI models produce accurate RSSI measurements originating further reductions in energy consumption, as is shown in the examples in Section VI.

Algorithm 2 Sensor activation mechanism

Require: q_k
1: $A^+ = \emptyset$
2: **for all** $a_k \in A$ **do**
3: $r(q_k, a_k) = H(q_k) - H(q^*_{k+1})$
4: **if** $r(q_k, a_k) - c(q_k, a_k) > 0$ **then**
5: $A^+ \leftarrow a_k$
6: **end if**
7: **end for**
8: $\hat{a}_k = \arg\max_{a_k \in A^+}(r(q_k, a_k) - c(q_k, a_k))$
9: Perform action $\hat{a}_k$

VI. Experiments

The implemented version of the scheme executes the three mechanisms (RSSI-model training in Algorithm 1, EIF and the sensor activation in Algorithm 2) one after the other at each time step. It has been evaluated in series of experiments performed in the *CONET Integrated Testbed* [1]. The objective was to track a moving robot, which ground truth location was computed using the AMCL method [17]. A total of 21 camera nodes were deployed on the room floor, each implemented with a *CMUcam3* camera connected to one *Crossbow TelosB*, see Fig. 4. Each *CMUcam3*, internally calibrated with the model in [18], captures 352x288 RGB images and executes efficient color and motion segmentation methods. The *CMUcam3* sends to its node the coordinates of the center of the region segmented in each image. Camera nodes also measure the RSSI of the messages received from the robot, tagged with another *TelosB*. Nodes implement the Flooding Time Synchronization Protocol (FTSP) [19] to ensure synchronization errors of few ms within the cluster.

Instead of using a default RSSI 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, depicted in Fig. 2-top, as our default RSSI 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 activation/deactivation method we adopted the energy consumption given by the manufacturers. *CMUcam3* consumed 650 mW when active and 0 when switched off. *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. Costs $c1$ and $c2$ are taken proportional to the consumptions of active TelosB and the *CMUcam3*.

Fig. 4. Left) Picture of the *CONET Integrated Testbed* taken in the experiments. Right) Camera node used in the experiments.

A. Performance evaluation

Figure 5 shows an scheme of the environment with the 21 camera nodes. The cameras local frames can be seen: Z represents the optical axis. Figure 5 also shows the result of one experiment with the proposed methods comprising the three mechanisms: the EIF, the RSSI model training and the sensor selection method described in Section V-B. The estimated target location is represented in red and the ground truth location is in blue. The average error in this experiment was 32.9 cm, with errors of 21.3 cm and 25.07 cm in X and Y, respectively. The mean overall power consumed by all devices along this experiment was 2019 mW.

Its performance was compared with other methods. The same experiment was executed 40 times. Camera and RSSI measurements of all the camera nodes were logged and off-line processed using different methods. Table I summarizes the main results. All the methods used the same parameters including observation and prediction covariances and $c1$ and $c2$ costs. *Method1* implements an EIF using only camera measurements and without any mechanisms: all cameras are kept active while seeing the object. It obtained the highest accuracy at the expense of requiring an average of 11.5 active cameras along the experiment, which consumed an average of 9001 mW. *Method2* is *Method1* with the camera selection method proposed in [20]. It dynamically deactivated the camera nodes that do not provide informative measurements and only an average of 40% of the cameras that sense the object were kept active. It achieved a reduction of 60% in energy consumption.

Traditional integration of RSSI in camera-based systems does not improve performance w.r.t. using only camera mea-

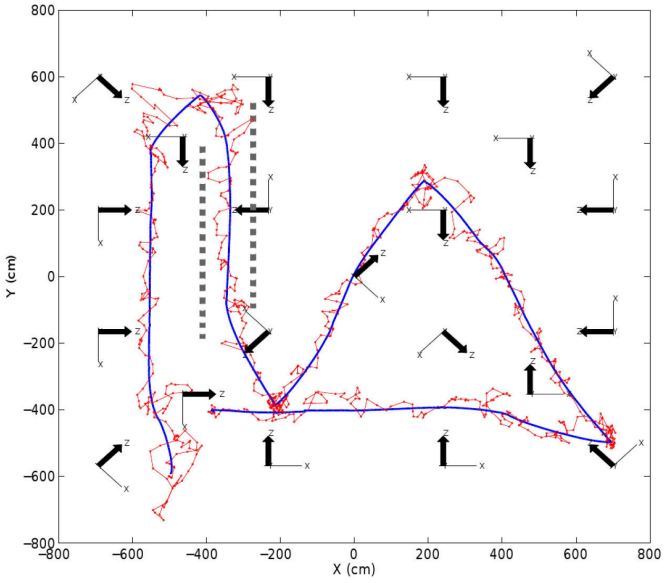


Fig. 5. Result of the proposed scheme: estimated location (in red) and ground truth in solid (blue).

TABLE I

EVALUATION AND COMPARISON OF THE PROPOSED METHOD IN SERIES OF 40 EXPERIMENTS.

	Method1	Method2	Method3	Proposed method
Mean error (cm)	28.59	32.50	32.95	33.54
Std (cm)	21.41	23.63	23.31	24.91
Average number of active cameras	11.50	4.59	3.11	2.56
Average number of RSSI measurements	-	-	3.11	3.38
Average power (mW)	9001	3418	2364	2024

surements, as is analyzed later. *Method3* integrates camera and RSSI measurements using the RSSI model training method proposed in Section IV and the sensor selection method in Section V-A. In this case an average of only 3.11 camera nodes (which provide simultaneously camera and RSSI measurements) were active in the experiment, which consumed 30% lower than *Method2* with almost the same accuracy.

The proposed scheme is the same as *Method3* but using the sensor selection described in Section V-B. While in *Method3* active camera nodes provide camera and RSSI measurements, in the proposed method they can also be in the *CameraOff* mode and only RSSI measurements are taken keeping cameras off. The proposed method required an average of only 2.56 active cameras and 0.86 nodes in the *CameraOff* mode. While having almost the same errors, our method consumed 14% less than *Method3*. It consumed less than methods based on only cameras: 40% less than *Method2* and 77% less than *Method1*. In the proposed method batteries last more than 4 times longer than in *Method1*.

In the proposed method the joint execution of the three mechanisms generates a behavior in which camera measurements are used to train RSSI models and once trained, the sensor selection method tends to deactivate cameras due to their higher consumption. The behavior is self-regulated. When too many cameras are inactive, the accuracies of RSSI models degrade, involving higher uncertainty in the overall target estimation, which makes the sensor selection method to activate cameras again to reduce uncertainty.

B. Robustness analyses

Now we analyze the accuracy of the training RSSI model method proposed in Section IV. We employed the same setting in the *CONET Integrated Testbed* and tracked the robot using an EIF that integrates only RSSI measurements. We compared the performance of the EIF with three different RSSI models: the default RSSI model, the RSSI calibration method presented in [10] and the RSSI model training method proposed in this paper. The experiment was repeated 40 times. Figure 6 shows the cumulate robot localization errors obtained when using the three RSSI models. The 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 models were generated for that specific environment.

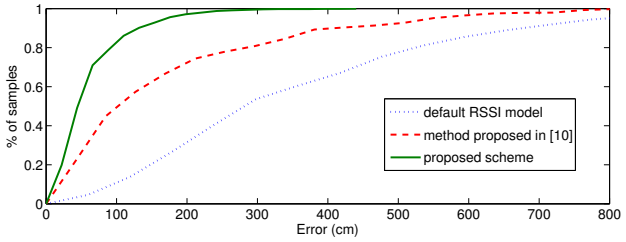


Fig. 6. Cumulative robot localization errors obtained by RSSI-only tracking using: the default RSSI model, the RSSI calibration method in [10] and the training method proposed in Section IV.

Our RSSI training mechanism behaves better because it estimates target location with cameras and trains the RSSI model dynamically considering the local surroundings of the target and of the static node. As presented in Section IV the proposed scheme includes self-corrective mechanisms that improve robustness against camera errors. In the next experiments we analyze the robustness of the RSSI trailing method against the most common errors in the cameras. Three cases that compromise the accuracy of the auxiliary EIF are analyzed: camera failures, camera pointing errors and errors in target segmentation in the images. These errors were simulated in these robustness experiments. Figure 7 compares the mean error (blue color) and energy consumed (red) by the proposed method (solid line) and the aforementioned *Method2* (dashed), which uses only cameras.

Figure 7-top analyzes robustness assuming each camera can fail randomly with a probability in the range [0, 50] %. The mean tracking error behaves similarly in the proposed method and in *Method2*. Our method consumes an average

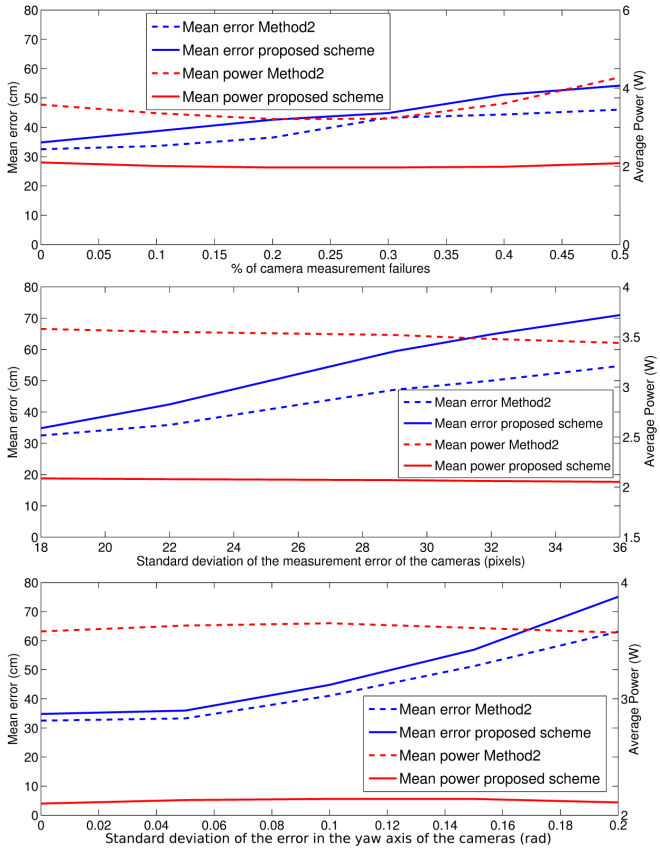


Fig. 7. Robustness against different types of camera errors between *Method2* and the proposed method: top) camera failures, center) target segmentation errors and bottom) camera pointing errors.

of 43% lower than *Method2*. Figure 7-center analyzes robustness against errors in the image segmentation methods. In preliminary tests we noticed that the measured center of the object had errors with a standard deviation of 18 pixels. In these experiments we added additional random errors so that the total error had standard deviations in the range [18, 36] pixels. Both methods had similar performance with low image segmentation errors but our method performed 22% worse with high image segmentation errors. Again, the proposed method saved a mean of 41% energy w.r.t. *Method2*. Figure 7-bottom analyzes robustness assuming that the all cameras pointing contain random yaw errors with standard deviations in the range [0, 0.2] rad. The mean error increases in a similar way in both methods and energy consumption in the proposed method is 42% lower than in *Method2*.

One further advantage of the proposed method is robustness against camera occlusions. 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. 5 with dashed lines. We focus on their performance during the occlusion. Their operation before and after is as described above. Figure 8 shows the estimated path (red line) and ground truth (blue). In the occluded section the lack of camera measurements prevented the EIF in *Method2* from updating the state vector resulting in a mean error of 188 cm. The proposed method had a mean error of 68 cm. Occlusions originated high uncertainties in the RSSI training method and the default RSSI model was selected. However, the integration of RSSI measurements even using the default RSSI model was a significant advantage when no camera measurements were available.

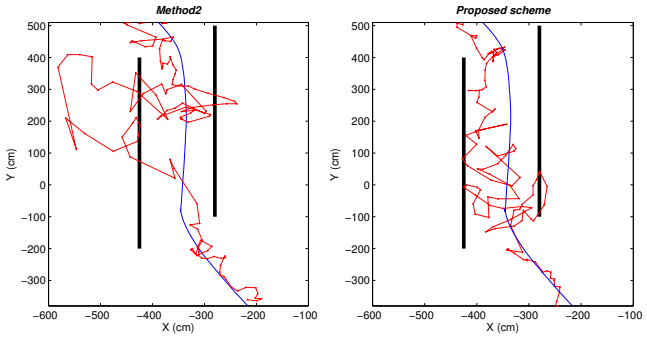


Fig. 8. Comparison of the robustness against occlusions between *Method2* and the proposed scheme.

The proposed method, consuming 40% less, has robustness against camera errors comparable to that of *Method2* and is significantly more robust to target occlusions.

VII. Conclusions

This paper presents a scheme that employs three mechanisms for the efficient integration of RSSI measurements in target localization and tracking with Wireless Camera Networks. Besides a distributed EIF, it includes a training method in which each node dynamically adapts its RSSI-range model considering the target current location and a sensor selection method that balances the different accuracies and energy consumptions of cameras and RSSI.

The joint use of these mechanisms generates a self-regulated behavior in which camera measurements are used to train RSSI 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 based on only cameras.

These mechanisms have been implemented with *TelosB* nodes connected to *CMUcam3* modules and validated in the indoor *CONET Integrated Testbed*. The experiments confirmed its performance: its robustness against camera errors is comparable to that of *Method2* and is significantly more robust to target occlusions.

The work presented opens wide fields for future application and research. 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 active sensors in order to avoid big clusters. Design of WCN deployment to improve tracking is also object of current research.

REFERENCES

[1] A. Jiménez-González, J. R. Martínez-de Dios, and A. Ollero, "An integrated testbed for cooperative perception with heterogeneous mobile and static sensors," *Sensors*, vol. 11, no. 12, pp. 11 516–11 543, 2011.

[2] H. Medeiros, J. Park, and A. Kak, "Distributed object tracking using a cluster-based kalman filter in wireless camera networks," *Journal of Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 448–463, 2008.

[3] A. O. Ercan, A. El Gamal, and L. J. Guibas, "Object tracking in the presence of occlusions via a camera network," in *Proc. 6th Intl. Conf. on Information processing in sensor networks*, 2007, pp. 509–518.

[4] A. Jiménez-González, J. R. Martínez-de Dios, A. De San Bernabé, and A. Ollero, "Wsn-based visual object tracking using extended information filters," in *Proc. Intl. Workshop on Networks of Cooperating Objects*, 2010.

[5] X. Wang, O. Bischoff, R. Laur, and S. Paul, "Localization in wireless ad-hoc sensor networks using multilateration with rssi for logistic applications," *Procedia Chemistry*, vol. 1, no. 1, pp. 461–464, 2009.

[6] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. 7th Intl. Conf. on Mobile Computing and Networking*, vol. 2001, 2001, pp. 166–179.

[7] C. Liu, K. Wu, and T. He, "Sensor localization with ring overlapping based on comparison of received signal strength indicator," in *Intl. Conf. on Mobile Ad-hoc and Sensor Systems*, 2004, pp. 516–518.

[8] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. 9th Intl. Conf. on Mobile computing and networking*, 2003, pp. 81–95.

[9] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piché, "A comparative survey of wlan location fingerprinting methods," in *6th Workshop on Positioning, Navigation and Communication*, 2009, pp. 243–251.

[10] A. De San Bernabé, J. R. Martínez-de Dios, and A. O. Baturone, "A wsn-based tool for urban and industrial fire-fighting," *Sensors*, vol. 12, no. 11, pp. 15 009–15 035, 2012.

[11] T. Miyaki, T. Yamasaki, and K. Aizawa, "Multi-sensor fusion tracking using visual information and wi-fl location estimation," in *1st Intl. Conf. on Distributed Smart Cameras. ICDSC'07.*, 2007, pp. 275–282.

[12] ——, "Tracking persons using particle filter fusing visual and wi-fi localizations for widely distributed camera," in *Intl. Conf. on Image Processing, 2007. ICIP 2007.*, vol. 3, 2007, pp. 225–228.

[13] W. Zhang and G. Cao, "Dctc: dynamic convoy tree-based collaboration for target tracking in sensor networks," *Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1689–1701, 2004.

[14] A. Ercan, D. Yang, A. El Gamal, and L. Guibas, "Optimal placement and selection of camera network nodes for target localization," *Distributed Computing in Sensor Systems*, pp. 389–404, 2006.

[15] S. Thrun, W. Burgard, D. Fox, *et al.*, *Probabilistic robotics*. MIT press Cambridge, MA, 2005, vol. 1.

[16] P. Kumar, L. Reddy, and S. Varma, "Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks," in *5th Conf. on Wireless Communication and Sensor Networks (WCSN)*, 2009, pp. 1–4.

[17] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *Proc. National Conf. on Artificial Intelligence*, 1999, pp. 343–349.

[18] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition.*, 1997, pp. 1106–1112.

[19] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proc. 2nd Intl. Conf. on Embedded networked Sensor Systems*, 2004, pp. 39–49.

[20] A. de San Bernabe, J. Martinez-de Dios, and A. Ollero, "Entropy-aware cluster-based object tracking for camera wireless sensor networks," in *Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 3985–3992.