A Probabilistic Framework for Entire WSN Localization Using a Mobile Robot

F. Caballero [a] L. Merino [b] P. Gil [a] I. Maza [a] Aníbal Ollero [a]

[a] *University of Seville, Camino de los Descubrimientos s/n, 41092, Seville, Spain*

[b] *Pablo de Olavide University, Crta. de Utrera, km. 1, 41013, Seville, Spain*

Abstract

This paper presents a new method for the localization of a Wireless Sensor Network (WSN) by means of collaboration with a robot within a Network Robot System (NRS). The method employs the signal strength as input, and has two steps: an initial estimation of the position of the nodes is obtained centrally by one robot and is based on particle filtering. It does not require any prior information about the position of the nodes. In the second stage, the nodes refine their position estimates employing a decentralized information filter. The paper shows how the method is able to recover the 3D position of the nodes, and is very suitable for WSN outdoor applications. The paper includes several implementation aspects and experimental results.

Key words: Wireless Sensor Network, Localization, Mobile Robots, Particle Filter, Information Filter

1 INTRODUCTION

In Network Robot Systems (NRS), a team of robots is expected to cooperate with sensors embedded in the environment for tasks like information gathering, tracking, surveillance, etc. Several different kinds of sensor networks can be expected, like surveillance cameras, RFID readers, etc. Latest advances in low-power electronics and wireless communication systems have made possible a new generation of devices able to communicate, sense environmental variables and even process this information, the Wireless Sensor Networks (WSNs). In WSNs, the sensors are cheap and the whole network can consist of hundreds of sensors. In addition, the

[1] This work is partially supported by the URUS project (IST-2006-045062) funded by the European Commission, and the AEROSENS project (DPI-2005-02293) funded by the Spanish Government

recent commercialization of some of these devices has increased the applicability and research efforts in this area.

Collaborative perception between robots and the sensor networks require to anchor all the data gathered to the same reference frame. Therefore, the localization of sensor nodes for sensor network deployment is an essential problem for NRS. It can be also important for the network performance when geographic-based routing methods are used. While most of the WSNs installed indoors are manually localized, localization of all the nodes in outdoor applications is still an open problem because GPS-based solutions are usually not viable due to the cost, the energy consumption and the satellite visibility from each node.

This paper addresses the WSN localization problem in outdoor environments by using a mobile robot. The paper describes a probabilistic framework where the localization of an entire WSN can be estimated by analyzing the interactions of a robot with the network. The approach takes advantage of the good localization capabilities of the robot and its mobility to compute an initial estimation of the node positions. The estimated position of the nodes could be used by the robot to better plan actions for data recovering. Moreover, once an initial estimation is obtained, a second localization stage is launched to refine the position of the nodes in a distributed manner. The received signal strength from neighbor nodes is used to improve their position employing a decentralized scheme based on Information Filtering.

The paper is structured as follows. Firstly, the full approach is outlined in section 2. Section 3 details the proposed method to compute an initial estimation of the position of the nodes. Then, a distributed technique for localization refinement is described in section 4. Finally, some experimental results with a real network are shown.

1.1 Related work

Localization of WSNs is an active field of research and some methods have been proposed. Most of them [2,11,15] are based on a small and well distributed set of nodes with known positions, called beacon-nodes. The position of these nodes is computed by means of a positioning sensor such as GPS or active/passive positioning devices. Sometimes this position is simply pre-computed and stored in the node. This information is propagated to the entire network and, finally, the radio interface of the wireless nodes is used to estimate the distance to the beacon-nodes which is used to estimate the localization of the nodes using triangulation, maximum bounding or other techniques.

A pre-calibrated relation between the Received Signal Strength Indication (*RSSI*) provided by the communication circuitry and the distance is used to obtain a low-accuracy estimation which worsens with the distance from the node to the beacon-nodes, see [12]. Improvements in the radio interface to increase the directivity, as in [19], or the inclusion of new features like time-of-flight have been proposed [18] but, in general, they lead to a power consumption growth and to a radio coverage reduction. In [9] time of arrival and direction of arrival are used to successfully localize a wireless sensor network but a number of requirements such as sensor node processor clock synchronization, special signal source devices and direction of arrival are needed. Unfortunately, even having well localized beacon-nodes and a reliable system to propagate this information, the results will be poor if the beacon-nodes are not appropriately distributed in the network.

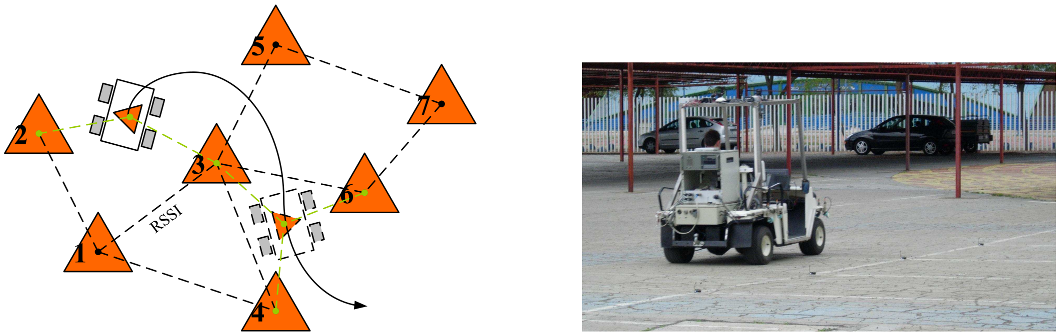
This paper proposes to use a mobile robot equipped with a DGPS device for WSN calibration. A mobile node can substitute a set of fixed beacon-nodes. Some authors have proposed the use of mobile robots for calibration of sensor networks. For instance, in [1] the authors present a similar approach based on particle filters for the calibration of a network of cameras. A particle filter is used to estimate the pose of the cameras, and it employs as data the position on the image plane of a localized robot. It also combines this with a Kalman Filter for position refinement. However, results with only one camera are presented, and no distributed approach is proposed. In [13], the authors consider the use of a mobile robot carrying a calibration pattern for the simultaneous localization, mapping and calibration of a indoors network of cameras. There, a centralized Extended Kalman Filter is employed to fuse all the information and the use of the artificial pattern allows to fully observe the position of the robot relative to the camera.

A technique for WSN localization using mobile robots has been also presented recently in [3]. The authors detail a technique based on potential fields that exploits the position information of a team of Unmanned Aerial Vehicles (UAVs) to localize the nodes. The concept underlying the approach is similar to ours, although the solution adopted is different; moreover, an accurate range sensor is required for computing the distance from the UAVs to the nodes. Here no special device for distance computation is used.

2 TECHNIQUE OVERVIEW

The purpose of the presented technique is to localize all the nodes of the network by using the information gathered by a mobile robot. The robot is able to communicate with the WSN in order to obtain environmental information. These data are sent to the robot by using a radio frequency stage that informs about the signal strength on reception (the *RSSI* value).

The approach, outlined in Fig. 1, uses the received signal strength to estimate the position of the emitter. The technique can be divided into two basic steps. Firstly,



(a) (b)

Fig. 1. (a): Scheme of the approach. The signal strength is used to estimate the position of the nodes of the network. The mobile robot computes centrally an initial estimation employing a separate Particle Filter per node. In the second step, a decentralized Information Filter integrates at each node information received from neighbor nodes and the robot. (b): An example, a ground robot (Romeo) driving through the network

a Particle Filter is used to process the *RSSI* value received from each node to compute an initial estimation of node locations in a static wireless network. The filter takes into account the uncertainty associated with the *RSSI* value and with the robot position (provided by a DGPS device) in order to optimally compute the position of the node. In the second step, the initial estimation of the position of the nodes (represented by mean and standard deviation) is sent to them. A distributed Information Filter is implemented in each node in order to easily improve the localization using the signal strength received from other nodes of the network, including the mobile robot.

The following characteristics differentiate this technique from other approaches:

- A Bayes filter is employed for the estimation of the localization of the nodes. The estimated position of the nodes will be represented by a probability distribution. This allows to take into account the uncertainty on measures involved in the process, mainly the relationship between *RSSI* and distance.
- The mobile robot position is included in the localization process and integrated along time. It allows to reduce one of the endemic problems of the *RSSI*-based localization algorithms: the distortion induced by radio-frequency effects. The chance of measuring the *RSSI* at different robot positions and orientations permits automatic detection of outliers and, hence, an improvement in the distance computation.
- Once an initial solution is computed, the network is able to use the localization information of all the nodes, including the position of the robot, to improve the estimation. This increases the flexibility of the technique.
- There is no triangulation but an estimation process so that the algorithm can consider hypotheses in which the estimated position is spread over a certain area.

3 PARTICLE FILTER BASED NODES LOCALIZATION

3.1 Filter overview

The objective of the localization algorithm is to estimate the position of the nodes of the network from the data provided by the node onboard the robot equipped with DGPS. A separate filter is implemented per node. Then, the state to be estimated consists of the position of each node $\mathbf{x} = \begin{pmatrix} X & Y & Z \end{pmatrix}^T$. The information about the state will be obtained from the set of measurements $\mathbf{z}_{1:k}$ received up to time k. This set of measurements consists of pairs of *RSSI* and robot position values $\{\mathbf{x}_k^r, RSSI_k\}$ (the algorithm considers a moving robot, and thus the time subscript for the robot position).

The method is based on Particle Filtering. This technique allows implementing recursive Bayesian filtering by Monte Carlo sampling. The key idea is to represent the posterior density at time k $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ by a set of independent and identically distributed (i.i.d.) random particles $\{\mathbf{x}_k^{(i)}\}$ according to the distribution. Each particle is accompanied by a weight $\omega_k^{(i)}$. Sequential observations and model-based predictions will be used to update the weight and particles respectively. See [4] for more details.

Particle Filters allow Bayesian estimation to be carried out approximately but in a structured and iterative manner, that simplifies the implementation. In general, it is very suitable for non-gaussian stochastic processes with non-linear dynamics and very useful when the posterior $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ has no parametric form or this form is unknown. 3D localization of nodes with non prior information implies a completely unknown posterior, so that Particle Filter seems to be a good solution to address the node localization problem.

Although there are many possible implementations, in the proposed algorithm the prior probability distribution $p(\mathbf{x}_0)$ is used as the importance (or proposal) distribution to draw the initial set of particles at time 0, i.e. $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$. Then, these particles are recursively re-estimated following the algorithm shown in Algorithm 1.

Next subsections describes the main issues in the actual implementation of the algorithm. As the likelihood function is the core of the algorithm, it is described first. Then the updating step, the prior distribution, the prediction step and the resampling procedure are detailed. Finally, some guidelines for computing the mean and standard deviation in the filter are mentioned.

Algorithm 1 $\{\mathbf{x}_k^{(i)}, \omega_k^{(i)}; i = 1,\ldots,L\} \leftarrow$ Particle_filter($\{\mathbf{x}_{k-1}^{(i)} \omega_{k-1}^{(i)}; i = 1,\ldots,L\}, \mathbf{z}_k = \{\mathbf{x}_k^r, RSSI_k\}$)

for $i = 1$ to L **do**
  sample $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$
  Compute $d_k^{(i)} = \|\mathbf{x}_k^{(i)} - \mathbf{x}_k^r\|$
  Determine $\mu(d_k^{(i)})$ and $\sigma(d_k^{(i)})$
  Update weight of particle i $\omega_k^{(i)} = p(RSSI_k|\mathbf{x}_k^{(i)})\omega_{k-1}^{(i)}$ with $p(RSSI|\mathbf{x}_k^{(i)}) = \mathcal{N}(\mu(d_k^{(i)}), \sigma(d_k^{(i)}))$
end for
Normalize weights $\{\omega_t^{(i)}\}, i = 1,\ldots,L$
Compute N_{eff}
if $N_{eff} < N_{th}$ **then**
  Resample with replacement L particles from $\{\mathbf{x}_k^{(i)}, \omega_k^{(i)}; i = 1,\ldots,L\}$, according to the weights $\omega_k^{(i)}$
end if

3.2 *The likelihood function*

The likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ plays a very important role in the estimation process. In this case, this function expresses the probability of obtaining a given *RSSI* value on the node onboard the robot (at position $\mathbf{x}_k^r$) given the position of the emitter node $\mathbf{x}_k$.

Experimental results (Fig. 2) show that there exists a correlation between the distance that separate both nodes and the *RSSI* value, although this correlation decreases with the distance between the two nodes, transmitter and receiver. This is mainly caused by radio-frequency effects such as radio reflection, multi-path or antenna polarization.

The model used here considers that the conditional density $p(\mathbf{z}_k|\mathbf{x}_k)$ can be approximated as a Gaussian distribution for a given distance $d_k = \|\mathbf{x}_k - \mathbf{x}_k^r\|$ as follows:

$$RSSI_k = \mu(d_k) + \mathcal{N}(0, \sigma(d_k)) \tag{1}$$

where the functions $\mu(d_k)$ and $\sigma(d_k)$ are non-linear functions of the distance (which itself is a non-linear function of the state).

These functions are estimated offline from a training data set. A couple of nodes have been distanced from 0 to 30 meters and the *RSSI* has been recorded for each distance. This experiment has been repeated with several antenna polarizations. A least squares process was used to compute the $\mu(d_k)$ and $\sigma(d_k)$ functions that best

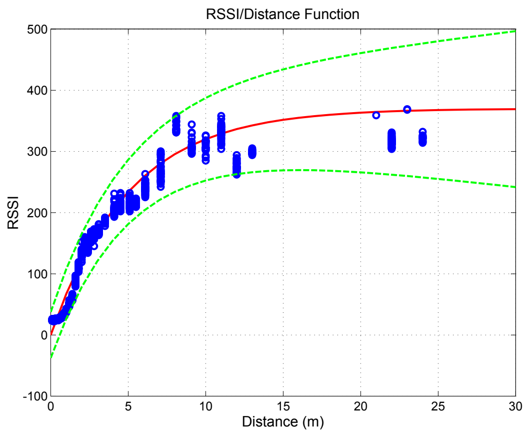


Fig. 2. RSSI-Distance functions, $\mu(d_k)$ and $\sigma(d_k)$. These functions relate the distance between two nodes and the RSSI received in mean and std. deviation. It has been experimentally computed using a large set of RSSI/distance couples. The RSSI representation is the one used in the Mica2 nodes, 0 is the maximum signal strength and 375 the minimum. Dots: A sub-set of the experimental set of data. Solid line: Estimated mean $\mu(d_k)$. Dashed lines: standard deviation confidence interval based on $\sigma(d_k)$.

fit the set of data. The computed equations are the following:

$$\mu(d_k) = 360 \cdot (1 - e^{-0.2 \cdot d_k}), \qquad \sigma(d_k) = 2.11 \cdot d_k + 25.36 \tag{2}$$

These equations are also shown in Fig. 2. As expected, it can be seen that the standard deviation increases with the distance d_k.

Note that the empirical model defined by (1) and (2) consider not only the antenna properties, but also the filtering and data conversions carried out by the communications circuitry. For that reason, the model does not match with the classic logarithmic free space propagation equations. Nevertheless, the experiment data agree with those obtained in [14], where the authors also identify quasi-gaussian distributions in the relations RSSI/distance for a fixed distance.

This experiment is only carried out for a couple of nodes of the network. Unfortunately, the nodes on a WSN are similar but not exactly the same, and therefore the previous relations should be computed for all the nodes of the network. To avoid this problem, the computed standard deviation has been intentionally overestimated in order to include as much nodes as possible. As a result, this overestimation increases the time needed to converge to a correct solution in the Particle Filter. Moreover, this overestimation does not solve the problem of biased measurements that could produce nodes which RSSI/Distance relation differs significantly from the one of the nodes used for the calibration. However, in the experimental results, with 25 nodes, no divergence was observed in the estimation due to biased measurements.

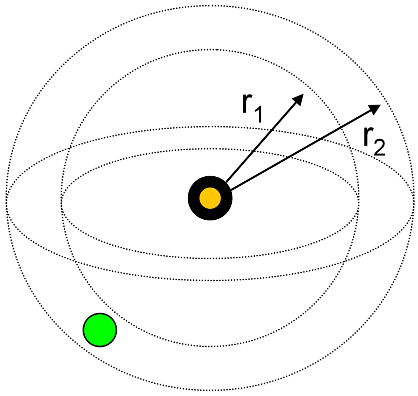


Fig. 3. Prior distribution. The initial samples are drawn from an uniform distribution over a spherical annulus. The inner (r_1) and outer (r_2) radius are a function of the estimated distance from the RSSI and its variance.

3.3 Updating

Once obtained, the functions $\mu(d_k)$ and $\sigma(d_k)$ are used online in the estimation process. Each time a new measure is received, the weights of the particles are updated considering the likelihood of the received data (lines 3, 4 and 5 of Algorithm 1).

The procedure is as follows. For each particle, the distance $d_k^{(i)} = \|\mathbf{x}_k^{(i)} - \mathbf{x}_k^r\|$ is obtained. From this distance, the mean and variance of the conditional distribution $p(\mathbf{z}_k|\mathbf{x}_k^{(i)})$ are obtained, so that $p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) = \mathcal{N}(\mu(d_k^{(i)}), \sigma(d_k^{(i)}))$.

The probability of the actual RSSI value under this distribution is finally employed to update the weight of the particle $\omega_k^{(i)}$.

$$\omega_k^{(i)} = \frac{1}{\sigma(d_k^{(i)})\sqrt{2\pi}} \exp(-\frac{(RSSI_k - \mu(d_k^{(i)}))^2}{2\sigma(d_k^{(i)})^2})\omega_{k-1}^{(i)} \quad (3)$$

After each update stage, the weights are normalized to have a sum equal to one (line 7 of Algorithm 1).

3.4 Initializing the filter. The prior model

The filter associated with a specific node is initiated when the first message is received in the mobile robot (it should be recalled that there is a separate filter per node). In this case, the *RSSI* distance functions of (2) are used inversely as in the estimation process. From the *RSSI* values, an initial distance is estimated, and also a corresponding variance on the distance.

The prior considered is then a uniform distribution on a spherical annulus, in which the inner and outer radius depend on the estimated mean and variance (see Fig. 3). As the number of particles is limited, not all the messages received initiate the filter. Only when a *RSSI* value corresponding to a variance below a threshold is received, the filter is initiated, in order to have a good resolution (particles per volume unit) with a limited number of particles. In the experiments show in this paper, this threshold has been set to $RSSI = 300$, that (according to Fig. 2) corresponds to distances shorter than 8 meters approximately.

3.5 Prediction

The nodes of the WSN are static, so the prediction step might be ommitted (that is, with probability 1 each node is in the same position at time k and $k-1$). However, as the resolution of particles over the state space is limited, a random move is added to the particles, in order to search locally over the area around the position of the previous time step. Therefore, the prediction model is:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathscr{N}(\mathbf{x}_{k-1}, \Sigma_{k-1}) \tag{4}$$

The value of Σ depends on the spatial distribution of particles, mainly the density of particles per volume unit. The idea behind is to make the particles move around their position in order to approximately cover the half of the distance to the neighbor particles. This way, the particle will integrate the mean value of the weight associated to its area, not only the given position.

3.6 Resampling

In general, the mission of the resampling step is to spatially distribute the particles in order to increase the sampling of the posterior in the areas where the likelihood is high. Thus, the resampling step will duplicate particles with high weights and will eliminate those with very low weights.

If no resampling is carried out in the Particle Filter, it will slowly converge to an only one particle with a weight close to 1, while the rest of particles will be weighted by 0. There are two problems related with this behavior: first, the estimated std. dev. becomes clearly sub-estimated, leading to the filter divergence, and, second, the spatial resolution of the filter is strongly limited by the number of particles, it leads to poor estimations. However, resampling reduces the diversity of the particle set [16].

Therefore, a resampling step (line 10 of Algorithm 1) is included in the filter to

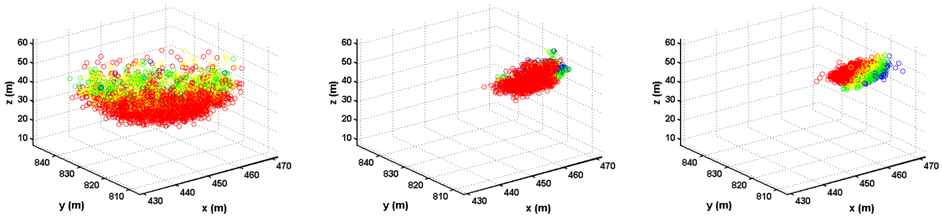


Fig. 4. An example of particles evolution at three stages. In the first one, it can be identified the sphere-like shape. As more messages are received, the particles concentrate into a unimodal distribution. The color represents the weight of the particle: Red between 0 and 0.25, Yellow between 0.25 and 0.5, Green between 0.5 and 0.75, Blue between 0.75 and 1.

increase the accuracy of the estimated position and to reduce the required particles. Two considerations are taken into account in this resampling step: first, resampling only takes place when the effective number of particles N_{eff} is below a threshold. The effective number is computed as follows:

$$N_{eff} = \left[\sum_{i=1}^{L} (\omega_k^{(i)})^2 \right]^{-1} \tag{5}$$

The threshold is set to the 10% of the number of particles, so $N_{th} = 0.1L$.

Second, the algorithm employed for sampling the particles space is a low variance sampler, particularly the algorithm described in [16] (p. 110). This method reduces the loss of diversity on the particle set in the resampling step.

3.7 *Estimation of mean and standard deviation*

The filter mean and standard deviation at time k can be computed as follows:

$$\mu_k = \sum_{i=1}^{L} [\mathbf{x}_k^{(i)} \omega_k^{(i)}], \qquad \sigma_k^2 = \sum_{i=1}^{L} [(\mathbf{x}_k^{(i)} - \mu_k)^2 \omega_k^{(i)}] \tag{6}$$

One of the benefits of the Particle Filter is that allows to face multi-modal or non-parametric hypothesis. While the posterior distribution will depend on the measures during the transient state, the filter approximately converge to a Normal distribution in the position of the node. Figure 4 shows an example of the evolution of the particles for one node. It has been considered that the filter converges when σ_k is below a certain threshold during a period of time. In the implementation the threshold was set to $3m$ during at least 20 messages.

If the filter converges at time k_0, the belief on the position of the node can be modeled as a Normal distribution such as $\mathcal{N}(\mu_{k_0}, \sigma_{k_0})$. This allows switching to another filter implementation like Extended Kalman Filter (EKF) or Extended Information Filter (EIF) which will efficiently take into account the gaussian nature of the posterior distribution. Next Section focus on the use of Information Filtering for refining the initial estimation given by the Particle Filter approach.

4 DECENTRALIZED NODES POSITION REFINEMENT EMPLOYING INFORMATION FILTERS

4.1 Description

Once the particle filter on the mobile robot has obtained a unimodal distribution for the position of one node, the mean μ and covariance matrix Σ are transmitted to it. Then, the node can locally refine its own position by using the information received from the mobile robot and also from its neighbor nodes (which also have an initial estimation of their positions). The messages exchanged among the nodes will be always accompanied by the estimated position of the emitter. This information, joined to the *RSSI* at the receiver, introduces a constraint on the possible positions of emitter and receiver. By using measures from several neighbors or the mobile robot, the position of the node can be further refined (see Fig. 5). Moreover, if a node also maintains an estimation of the position of the neighbor nodes in its communication range, this can be used for geographic routing of data.

The main issue for a decentralized estimation that runs on the nodes are the memory restrictions that these nodes have, with a storage space of a few kilobytes.

4.2 Local filters

The information from neighbor nodes is integrated employing an Information Filter (IF) [16]. The IF is a Gaussian Filter that employs the so-called *canonical representation* for the Gaussian distribution. The fundamental elements are the *information matrix* $\Omega = \Sigma^{-1}$ and the *information vector* $\xi = \Sigma^{-1}\mu$. The properties of the Information Filter allow easy decentralized data fusion at low computational cost thanks to an efficient updating stage and to the naturally sparse characterization of the information matrix respectively [16,5]. The sparseness of the information matrix has been employed, for instance, in batch algorithms for the full SLAM problem, as in [17]. Here, it will be employed to devise online algorithms for decentralized estimations that are efficient in term of memory requirements.

In the most general case, each node maintains a local estimation of its position and

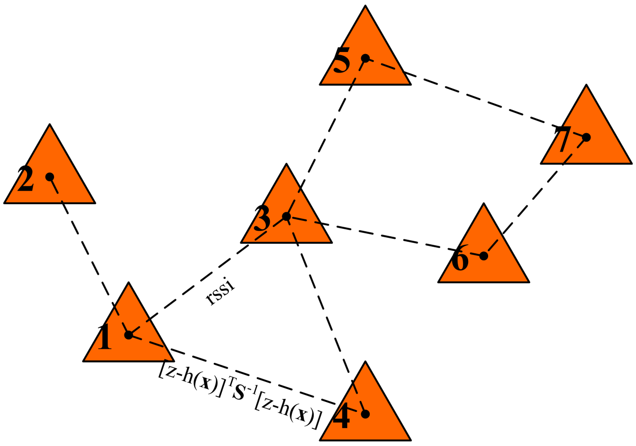


Fig. 5. The *RSSI* values received from neighbor nodes can be used as constraints over the positions of the different nodes of the network. These constrains are integrated by the nodes by using a decentralized Information Filter.

the position of its surrounding nodes. The state is maintained in information form. It is considered that nodes only have information about their own position at time 0. Thus, for node 1 (denoted by ξ^1 and Ω^1), the initial state is given by:

$$\Omega_0^1 = \begin{pmatrix} \Omega_{11}^1 & \mathbf{0} \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \dots & \mathbf{0} \\ \vdots & \vdots \ddots & \vdots \\ \mathbf{0} & \mathbf{0} \dots & \mathbf{0} \end{pmatrix} \xi_0^1 = \begin{pmatrix} \xi_1^1 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} \tag{7}$$

The state to be estimated (the node position and that of its neighbors) is static, so no prediction step is performed. Each node updates its map with the information received from its neighbors. In the general case, the received messages consists of the estimated position of the emitter. For instance, if the sender is node i, it sends its own position estimation as a pair ξ_i^i and Ω_{ii}^i. The receiver also computes the $RSSI_{1i}$ value of the incoming message.

As shown in Section 3.2 and illustrated in Fig. 2, the measurement function $RSSI_t = h(\mathbf{x})$ is non-linear on the state, so the Extended Information Filter (EIF) is employed. The updating equations for the EIF are:

$$\Omega_t^1 = \Omega_{t-1}^1 + \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{M}_t \tag{8}$$

$$\xi_t^1 = \xi_{t-1}^1 + \mathbf{M}_t^T \mathbf{S}_t^{-1} [\mathbf{z}_t - h(\mu_t) + \mathbf{M}_t \mu_t] \tag{9}$$

The updating equations of the EIF require an estimation of the variance σ_{rssi}^2 of the received RSSI value in information form $\mathbf{S}_t^{-1} = \frac{1}{\sigma_{rssi}^2}$. This value is estimated by using the expression (2) evaluated at the mean distance between nodes. Also, the Jacobian $\mathbf{M}$ of the measurement equation is required, which is obtained by linearizing the relation (2) around the current estimated mean of the state $\mathbf{x}$. The *RSSI* is a scalar value that depends on the distance between the two nodes $d_{1i} = \|\mathbf{x}_1 - \mathbf{x}_i\| = \sqrt{(\mathbf{x}_1 - \mathbf{x}_i)^T(\mathbf{x}_1 - \mathbf{x}_i)}$:

$$RSSI_{1i} = h(d_{1i}) = h(\|\mathbf{x}_1 - \mathbf{x}_i\|) \tag{10}$$

Then,

$$\mathbf{M} = \frac{\partial RSSI_{1i}}{\partial \mathbf{x}} = \frac{\partial h}{\partial d_{1i}} \frac{\partial d_{1i}}{\partial \mathbf{x}} = \underbrace{\frac{\partial h}{\partial d_{1i}} \frac{2}{2d_{1i}}}_{M} \Big((\mathbf{x}_1 - \mathbf{x}_i)^T \;\; \mathbf{0} \; \cdots \; -(\mathbf{x}_1 - \mathbf{x}_i)^T \; \cdots \; \mathbf{0} \Big) \tag{11}$$

where M is a scalar dependant on the mean distance between the nodes. It can be seen, from the form of $\mathbf{M}$, that the updating equations only affect the part of the information vector and matrix related to nodes 1 and i. Removing the time indexes, the final updating equations when receiving information from node i are:

$$\Omega_t^1 = \underbrace{\begin{pmatrix} \Omega_{11}^1 & \Omega_{1i}^1 \\ \Omega_{i1}^1 & \Omega_{ii}^1 \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Omega_{ii}^i \end{pmatrix}}_{fusion} + \frac{M^2}{\sigma_{rssi}^2} \begin{pmatrix} (\mathbf{x}_1 - \mathbf{x}_i)(\mathbf{x}_1 - \mathbf{x}_i)^T & -(\mathbf{x}_1 - \mathbf{x}_i)(\mathbf{x}_1 - \mathbf{x}_i)^T \\ -(\mathbf{x}_1 - \mathbf{x}_i)(\mathbf{x}_1 - \mathbf{x}_i)^T & (\mathbf{x}_1 - \mathbf{x}_i)(\mathbf{x}_1 - \mathbf{x}_i)^T \end{pmatrix} \tag{12}$$

$$\xi_t = \underbrace{\begin{pmatrix} \xi_1^1 \\ \xi_i^1 \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \xi_i^i \end{pmatrix}}_{fusion} + \frac{M}{\sigma_{rssi}^2} \begin{pmatrix} \mathbf{x}_1 - \mathbf{x}_i \\ -(\mathbf{x}_1 - \mathbf{x}_i) \end{pmatrix} [RSSI - h(\mathbf{x}) + \mathbf{M}\mathbf{x}] \tag{13}$$

where $\mathbf{x}_1$ and $\mathbf{x}_i$ are evaluated at the current means μ_1 and μ_i. As shown in [17], this estimation procedure is equivalent to obtaining the optimal position of the nodes under the restrictions on their positions $\mathbf{x}$ induced by the *RSSI* values, which are of the form $[\mathbf{z}_t - h(\mathbf{x})]\mathbf{S}_t^{-1}[\mathbf{z}_t - h(\mathbf{x})]^T$ (see Fig. 5).

There are several issues to be pointed out. First, the application of all the previous equations maintains a sparse structure for the information matrix about the position

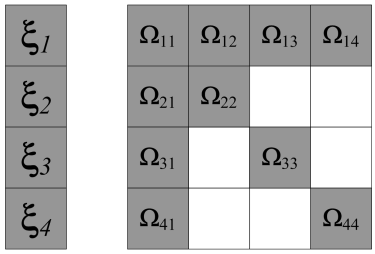


Fig. 6. Structure of the information matrix for the case considered.

of all the neighbor nodes that have communicated with node 1 (see Fig. 6). Then, it can be seen that the memory requirements are linear with the number of neighbor nodes. This is key issue for the implementation of the algorithm in sensor nodes.

Also, measurements induce relative relations, and therefore the state is not fully observable. However, the nodes are already observed after the initial estimation by using the particle filter. Also, the measurements from the robot, which position uncertainty is independent from that of the nodes, allows anchoring the nodes to a common reference frame. It is important to remark that the nodes do not maintain an estimation of the position of the node on board the mobile robot. Thus, when receiving information from the robot, the information about the robot position is marginalized out.

Finally, the received local estimation about node i is fused with the current one. As shown in eqs. (12) and (13), for a decentralized EIF, the fusion step is a simple addition of the local information. This step can be modified to incorporate more information, as described in the next section.

4.3 Decentralized estimation

The main idea is to estimate at each node the position of other nodes of the network in a decentralized manner. The previous equations involve a decentralized fusion of the marginal information ξ_{ii} and Ω_{ii} about the position of the emitter. However, the emitter could include in the message its local estimation about the position of all its surrounding nodes. This way, more information could be used for position refinement in the fusion step. Also, each node would finally had knowledge about the positions of all the nodes of the network.

The problem derived from this scheme is that the sparsity is lost (see Fig. 7). Therefore, the storage requirements increases (which is the main limitation for this kind of nodes), and the size of the message needed to transmit the information as well.

Besides, the updating equations require maintaining an estimation of the mean μ of $\mathbf{x}$ for the computation of the updating stage (12) and (13). This would require to solve the system $\xi = \Omega\mu$. If the sparsity of the system is maintained, efficient

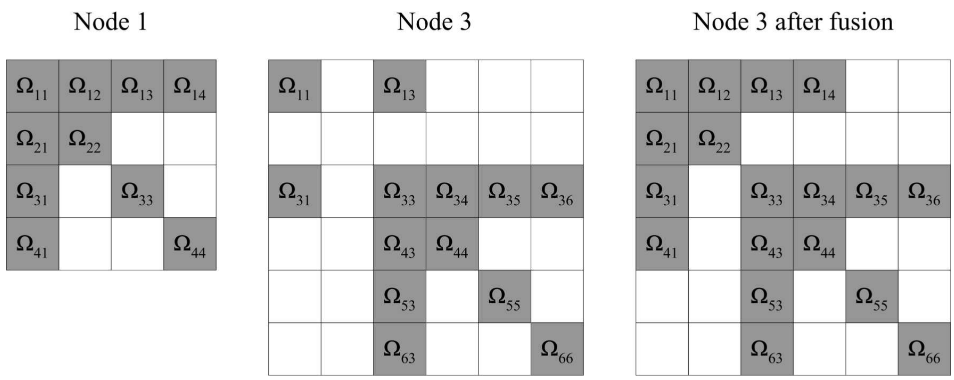


Fig. 7. Left and centre: local estimations of nodes 1 and 3 for the situation of Fig. 5. Right: estimation at node 3 after fusing all information from node 1. It can be seen that exchanging the full map leads to a higher density in the information matrix.

algorithms can be used.

The option considered here is to send with each message not only the marginal information about the emitter, but also its local estimation about the position of the receiver. This way, the fusion equations are changed by:

$$\Omega_t^1 = \begin{pmatrix} \Omega_{11}^1 & \Omega_{1i}^1 \\ \Omega_{i1}^1 & \Omega_{ii}^1 \end{pmatrix} + \begin{pmatrix} \Omega_{11}^i & \Omega_{1i}^i \\ \Omega_{i1}^i & \Omega_{ii}^i \end{pmatrix} \tag{14}$$

and:

$$\xi_t^1 = \begin{pmatrix} \xi_1^1 \\ \xi_i^1 \end{pmatrix} + \begin{pmatrix} \xi_1^i \\ \xi_i^i \end{pmatrix} \tag{15}$$

where the sums only affect the part of the state corresponding to nodes 1 and i, and thus maintain the structure of the information matrix. In order to do that, each time information is sent to a node, the marginal information about emitter and receiver is extracted. The marginal of a multivariate Gaussian can be computed in closed form [16]. In the particular case of matrices with the structure of Fig. 6, the computation requirements only involve inversions of 3×3 matrices and local operations. For instance, marginalizing out the information about node 4 in that example only affects node 1 (see Fig. 8):

$$\begin{aligned} \bar{\Omega}_{11} &= \Omega_{11} - \Omega_{14}\Omega_{44}^{-1}\Omega_{41} \\ \bar{\xi}_1 &= \xi_1 - \Omega_{14}\Omega_{44}^{-1}\xi_4 \end{aligned} \tag{16}$$

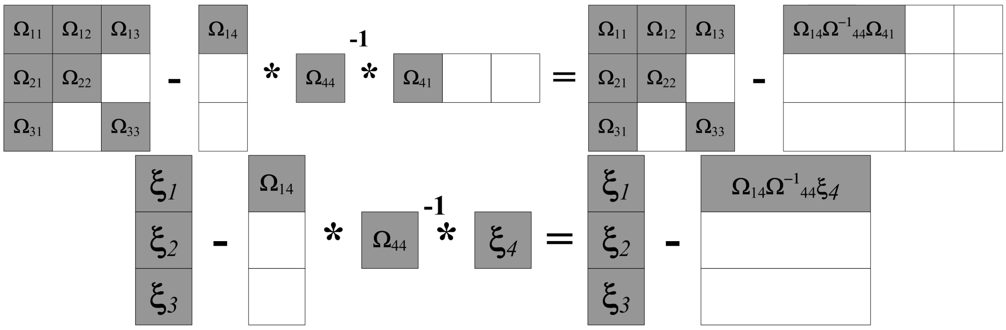


Fig. 8. Operations involved in the marginalization. Marginalization of the information about one node only requires small block operations and maintain the structure of the information matrix. In this example, operations involved to remove node 4 in the example shown in Fig. 6.

4.3.1 Conservative fusion rules

The above presented filter is employed locally at each node, and thus the full computation is decentralized. In the fusion operations (14) and (15), special care should be take to avoid considering several times the same information. In the most general decentralized case, there will be unknown correlations that should be taken into account. If they are not, the filter will diverge due to double counting of information [6]. This effect is commonly known as *rumor propagation*.

For the Information Filter, the Covariance Intersection method [7] gives consistent results even in presence of unknown correlations. Thus, the equations (14) and (15) are reformulated as follows:

$$\begin{aligned} \Omega_t^1 &= \alpha \Omega_{t-1}^1 + (1-\alpha)\Omega_t^i \\ \xi_t^1 &= \alpha \xi_{t-1}^1 + (1-\alpha)\xi_t^i \end{aligned} \tag{17}$$

where $\alpha \in [0,1]$. α is usually selected as the one that maximizes the determinant of the final information matrix. However, given the limited processing power of the nodes, a more heuristic approach is employed, and the local information is always more weighted than the received one.

Although not considered in the current implementation, as a similar approach could be used to limit the effect of not accounted cross-information due to the linearization of the measurement equations [8].

(a) (b) (c)

Fig. 9. Experiment setup. (a): 25 Mica2 nodes spread in the parking area of the University of Seville. (b): Romeo, the mobile robot used to interact with the network. (c): Node mounted on Romeo

5 EXPERIMENTAL RESULTS

This section details the results of an experimental setup conceived to test the above algorithms. A wireless sensor network composed of 25 Mica2 nodes was deployed in a parking area, see Fig. 9(a). The position of all the nodes were computed using a DGPS device in order to validate the algorithm estimations.

A ground robot (Romeo) is used to localize the nodes, see Fig. 9(b). Romeo is equipped with DGPS, gyroscope, compass and other navigation devices. Three computers onboard Romeo (one Pentium IV and two Pentium Mobile) allow complex data processing. In addition, one WSN node was mounted and connected to Romeo through a serial link cable, Fig. 9(c). The node runs a software that relays all received messages to Romeo.

The Particle Filter based localization has been implemented in C++ and runs onboard Romeo. The onboard software received all the messages from the WSN and the positioning information from the DGPS device. A filter with 4000 particles and non-prior information was implemented per node.

Figure 10 shows the evolution of the error in the estimated position for X, Y and Z axes for one node of the network. The error is computed as the distance between the estimation using the Particle Filter approach and the actual DGPS position of the node. The figure shows the estimated standard deviation per axis as well. It can be seen that the std. dev. is consistent with the error committed.

It is not possible to show the evolution of the error for all the nodes due to space restrictions. Instead, it is presented in Fig. 11 the mean error committed in the estimated position of the 25 nodes by using the Particle Filter approach. The figure shows the mean error of all the network after receiving x messages.

The Information Filter approach for position refinement is still not implemented inside the nodes, so that all the required data for testing the algorithm (*RSSI*, emitter and receiver) were extracted from the network through a gateway and stored in a file. It means that the experiments were carried out off-line but with actual data

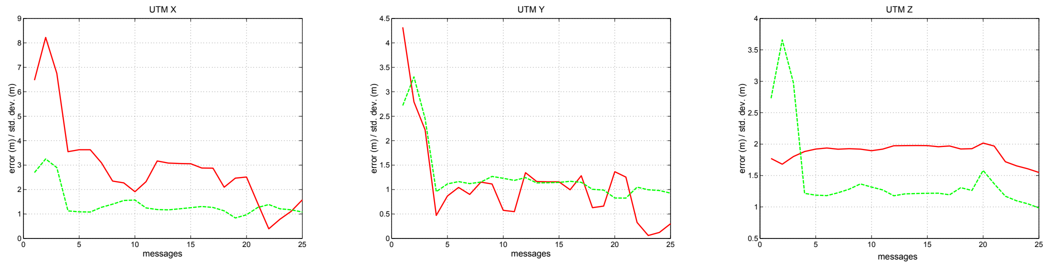


Fig. 10. Estimation error in meters computed as the distance between the DGPS position and the Particle Filter based estimation (red-solid). Standard deviation in meters obtained from the particles (green-dashed). The estimated distribution is consistent with the errors.

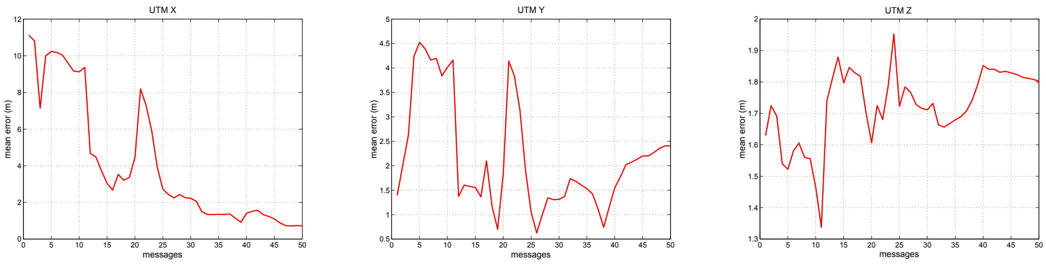


Fig. 11. The figures shown the mean error in meters committed after receiving *x* messages.

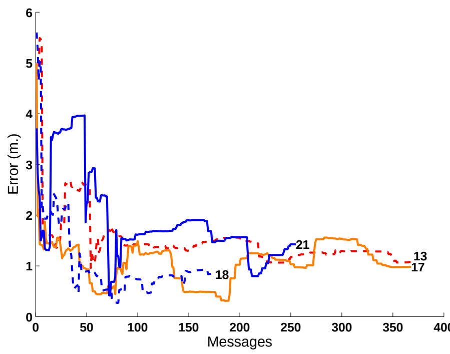


Fig. 12. Evolution of the error between the estimated position and actual position of several nodes employing the decentralized IF scheme.

from the network.

Figure 12 shows the correction on the error of the estimated position for several nodes. Figure 13 shows the estimated position for the two nodes compared to the actual ones.

6 CONCLUSIONS AND FUTURE WORK

The paper has shown a technique for estimating the position of the nodes of a WSN by using the received signal strength in a mobile robot and a distributed method

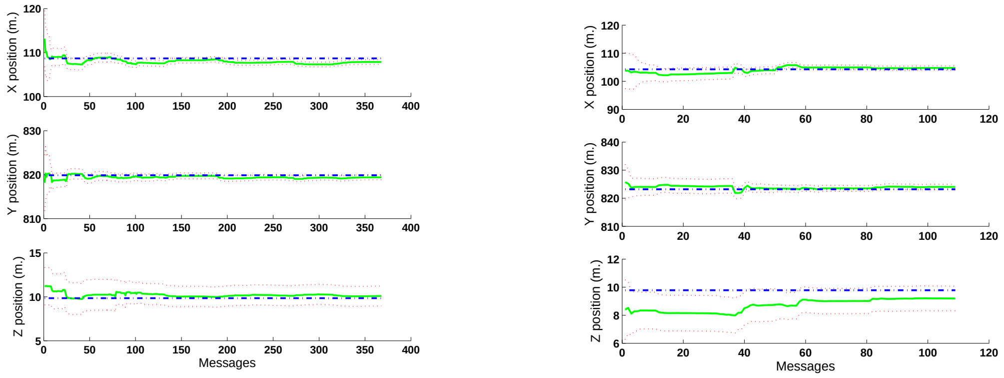


Fig. 13. Evolution of the estimated position for two different nodes by using the IF approach. Solid, estimated position. Dash-dot, position measured by DGPS. Depending on the distribution of the neighbor nodes, some coordinates are better resolved than others.

for position refinement using the normal data flow of the network. Both techniques have been tested in real conditions and the results showed their good feasibility.

The Particle Filter based localization has been tested online in a real robot. The needed computation to run the filters is low and the errors committed to the estimation are reasonable. The authors explored the implementation of this algorithm inside the nodes but the memory requirements for the particle storage, around 4k bytes, make it difficult given the usual memory restrictions in the nodes.

The decentralized scheme allows that improvements in part of the network propagates to others. Although this decentralized position refinement using the Information Filter has been tested offline, the results are promising. The low computation requirements needed to update the filter joint to a fixed state vector with three or four neighbors make possible the implementation inside the nodes. Next steps will consider this issue.

Certainly, some robot motions are much more convenient for the localization of the nodes than others. The inclusion of the network localization in the NRS path planning will be researched. Equation (12) allows to obtain an estimation of the information gain (for instance as the expected difference of entropy of the probability distributions) for a given position of the robot and an expected value of the *RSSI*. Figure 14 shows the estimated information gain for different positions of a robot and a network of three nodes. This information gain can be used within greedy exploration strategies to generate robot actions that will reduce the uncertainty in the estimated position of the nodes of the WSN.

Finally, in this paper the likelihood function that relates *RSSI* and distance was estimated off-line and then maintained fixed. Improvements to re-estimate online the functions by using methods like Expectation-Maximization [10] will be researched.

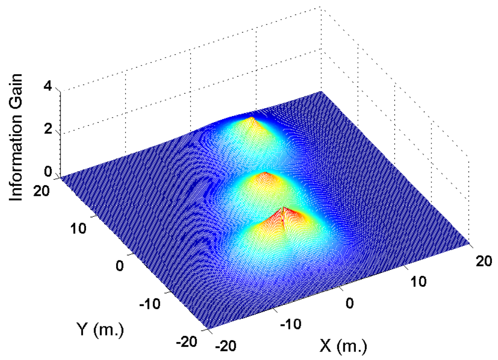


Fig. 14. Expected information gain for a network of three nodes at positions $(0,0)$, $(5,12)$ and $(-3,-10)$ and a mobile robot. The information gain is computed as the expected variation of the entropy of the distribution on the node position.

7 ACKNOWLEDGMENTS

The authors thank the help of Antidio Viguria for supporting part of the WSN experiments needed to validate the Particle Filter approach presented in this paper. The authors also thank the collaboration of the AWARE project team.

References

[1] A. Brooks, S. Williams, A. Makarenko, Automatic online localization of nodes in an active sensor network, in: Proc. IEEE Int. Conf. Robotics and Automation, New Orleans, 2004.

[2] N. Bulusu, J. Heidemann, D. Estrin, GPS-less low-cost outdoor localization for very small devices, IEEE Personal Communications 7 (5) (2000) 28–34.

[3] P. Dang, F. L. Lewis, D. O. Popa, Dynamic localization of air-ground wireless sensor networks, in: Proceeding of the 14th Mediterranean Conference on Control and Automation, 2006.

[4] A. Doucet, N. de Freitas, N. Gordon (eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, 2001.

[5] R. Eustice, H. Singh, J. Leonard, Exactly Sparse Delayed-State Filters for View-Based SLAM, IEEE Transactions on Robotics 22 (6) (2006) 1100–1114.

[6] S. Grime, H. F. Durrant-Whyte, Data fusion in decentralized sensor networks, Control Engineering Practice 2 (5) (1994) 849–863.

[7] S. Julier, J. Uhlmann, A non-divergent estimation algorithm in the presence of unknown correlations, in: Proceedings of the American Control Conference, vol. 4, 1997.

[8] S. Julier, J. Uhlmann, Using covariance intersection for SLAM, Robotics and Autonomous Systems 55 (1) (2007) 3–20.

[9] R. Moses, D. Krishnamurthy, R. Patterson, A self-localization method for wireless sensor networks, EURASIP Journal on Applied Signal Processing (4) (2003) 348–358.

[10] R. Neal, G. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, MIT Press, Cambridge, MA, USA, 1999, pp. 355–368.

[11] N. Patwari, A. O. Hero, Using proximity and quantized RSS for sensor localization in wireless networks, in: Proc. of the 2nd ACM international conference on Wireless sensor networks and applications, WSNA 2003, ACM Press, New York, NY, USA, 2003.

[12] V. Ramadurai, M. L. Sichitiu, Localization in wireless sensor networks: A probabilistic approach, in: Proceedings of the 2003 International Conference on Wireless Networks (ICWN 2003), Las Vegas, NV, 2003.

[13] I. Rekleitis, D. Meger, G. Dudek, Simultaneous planning, localization, and mapping in a camera sensor network, Robotics and Autonomous Systems 54 (11) (2006) 921–932.

[14] M. Sichitiu, V. Ramadurai, Localization of wireless sensor networks with a mobile beacon, in: Proc. of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), 2004.

[15] M. Terwilliger, A. G. V. Bhuse, Z. H. Kamal, M. A. Salahuddin, A localization system using wireless sensor networks: A comparison of two techniques, in: Proc. of the First Workshop on Positioning, Navigation and Communication, Hannover, Germany, 2004.

[16] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, The MIT Press, 2005.

[17] S. Thrun, M. Montemerlo, The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures, The International Journal of Robotics Research 25 (5-6) (2006) 403–429.

[18] K. Whitehouse, D. Culler, Macro-calibration in sensor/actuator networks, Mobile Networks and Applications 8 (4) (2003) 463–472.

[19] S. Yang, H. Cha, An empirical study of antenna characteristics toward rf-based localization for ieee 802.15.4 sensor nodes, in: 4th European conference on Wireless Sensor Networks, EWSN 2007, Delf, The Netherlands, 2007.