J Braz Comput Soc (2013) 19:89–104
DOI 10.1007/s13173-012-0084-4

ORIGINAL PAPER

Reducing the impact of location errors for target tracking in wireless sensor networks

Éfren L. Souza · Eduardo F. Nakamura ·
Horácio A. B. F. de Oliveira · Carlos M. S. Figueiredo

Received: 17 March 2012 / Accepted: 27 July 2012 / Published online: 14 August 2012

Abstract In wireless sensor networks (WSNs), target tracking algorithms usually depend on geographical information provided by localization algorithms. However, errors introduced by such algorithms affect the performance of tasks that rely on that information. A major source or errors in localization algorithms is the distance estimation procedure, which often is based on received signal strength indicator measurements. In this work, we use a Kalman Filter to improve the distance estimation within localization algorithms to reduce distance estimation errors, ultimately improving the target tracking accuracy. As a proof-of-concept, we chose the recursive position estimation and directed position estimation as the localization algorithms, while Kalman and Particle filters are used for tracking a moving target. We provide a deep performance assessment of these combined algorithms (localization and tracking) for WSNs are used. Our results show that by filtering multiple distance estimates in the localization algorithms we can improve the tracking accuracy, but the associate communication cost must not be neglected.

Keywords Target tracking algorithms · Localization systems · Kalman filter · Particle filter

1 Introduction

A wireless sensor network (WSN) [1] is a special type of ad-hoc network composed of resource-constrained devices, called sensor nodes. These sensors are able to perceive the environment, collect, process and disseminate environmental data. Tracking the location of a moving entity (event) represents an important class of applications for WSNs. For instance, animal tracking for long-term assessment of species to improve our knowledge about the biodiversity and support preserving and conserving the wildlife [7,21,28].

Target tracking is particularly dependent on location information, and current localization algorithms [2,24] cannot perfectly estimate every node location [25,30]. Various approaches have been proposed for target tracking in WSN, considering diverse metrics like accuracy, scalability, and density [6,18,31,36,38]. However, there is little research assessing the impact of localization algorithms have on the target tracking performance. Current approaches either assume that every sensor node knows its position perfectly [20], or simulate localization errors by adding a random noise variable in the correct node position [12,19].

In this work, we assess the performance of target tracking algorithms when position information are based on actual localization algorithms. Then, we demonstrate how an information fusion techniques [20] can be used to mitigate errors of localization algorithms, improving the target tracking accuracy. To do that, multiple distance estimates are fused

This work extend the previously evaluation made in Souza et al. [29] by introducing the usage of data fusion to reduce errors in the localization of sensor nodes. The results presented here show the benefits and costs of this new approach.

E. L. Souza (✉) · H. A. B. F. de Oliveira
Federal University of Amazonas, UFAM, Manaus, Brazil
e-mail: efren@dcc.ufam.edu.br

H. A. B. F. de Oliveira
e-mail: horacio@dcc.ufam.edu.br

E. F. Nakamura · C. M. S. Figueiredo
Analysis, Research and Technological
Innovation Center, FUCAPI, Manaus, Brazil
e-mail: eduardo.nakamura@fucapi.br

C. M. S. Figueiredo
e-mail: mauricio.figueiredo@fucapi.br

by a Kalman filter, during the localization process. Such an evaluation is a step towards the understanding of the relationship among localization and target tracking algorithms, and the design of integrated solutions that exploit features and requirements shared by these tasks.

As a proof-of-concept, we evaluate two localization algorithms on two tracking algorithms. The first localization algorithm is the recursive position estimation (RPE) algorithm [2]—a pioneer iterative solution—while the second is the directed position estimation (DPE) algorithm [24]—a solution that evolved from the original RPE. The tracking algorithms we evaluate are the Kalman filter (KF) [15] and particle filter (PF) [3]. These filters can be considered as canonical solutions for the target tracking problem.

The remainder of the work is organized as follows. In Sect. 2, we present the related work and background knowledge required for localization and target tracking problems. In Sect. 3, we present a simple information-fusion approach for reducing localization/tracking errors. Section 4 presents our experimental methodology and quantitative evaluation. Finally, in Sect. 5, we present our conclusions and future work.

2 Background and related work

In this section, we describe the state-of-the-art regarding localization and tracking algorithms, putting emphasis on the algorithms evaluated in this work.

2.1 Localization

A localization system in sensor networks basically consists of determining the physical location of the sensor nodes [25]. These systems are usually divided into three phases: distance estimation, position computation and localization algorithm [5].

In current localization solutions, a limited number of nodes, called beacon or anchor nodes, are aware of their positions. Then, distributed algorithms share beacon information, so that the remainder of the nodes can estimate their position. The ad hoc positioning system (APS) [22] works as an extension of both the distance vector routing and GPS positioning in order to provide a localization system in which a limited fraction of nodes have self-location capability (e.g., GPS-equipped nodes). An approach that uses mobile beacon to provide the node location in sensor networks is proposed by Sichitiu and Ramadurai [27]. In this algorithm, one or more beacon nodes move through the sensor field broadcasting their positions to all nodes within in the beacon range. When a node receives three or more positions it computes its own position. Tatham and Kunz [30] show that the position of the beacon nodes can impact the localization error, furthermore they propose a set of guidelines to improve the positions of the nodes using the smallest number of beacon nodes possible. The recursive position estimation (RPE) [2] iteratively computes the node location information without the need for strategic beacon placement. The directed position estimation (DPE) [24] is a similar algorithm that uses the direction of the recursion to improve the localization accuracy. Both the RPE and DPE propagate position errors throughout the network. However, in the DPE this error is reduced by selecting the best reference neighbors. These two algorithms are evaluated in this work, so they are treated in more detail in the next subsections.

2.1.1 Recursive position estimation

The RPE [2] is a positioning system that requires at least 5 % of the nodes to be beacon nodes, randomly distributed in the sensor field. However, depending on the network density and on the beacons arrangement, we need a larger number of beacons to start the recursion.

In this algorithm, every free node needs the minimum of three references to estimate its position. Estimated positions are broadcasted to help other nodes estimate their positions recursively. The number of estimated positions increases iteratively as new estimated nodes assist others estimating their positions.

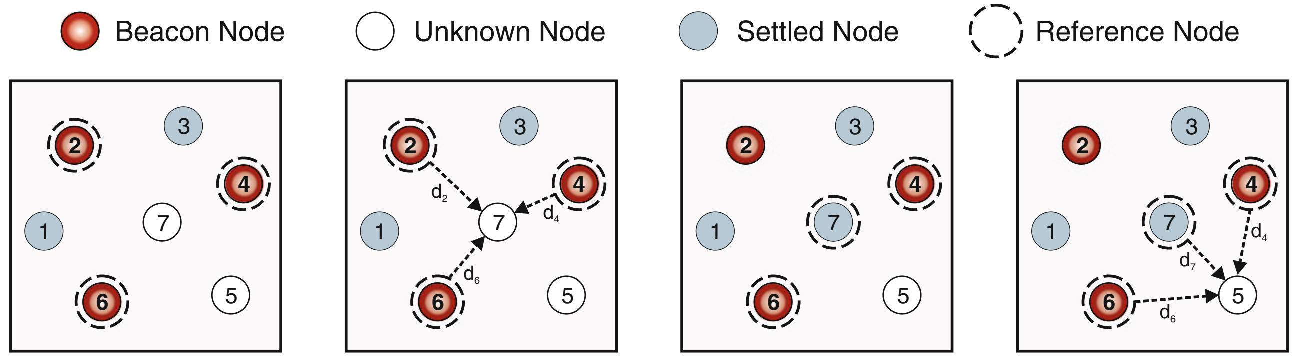
The RPE algorithm can be divided into four phases (see Fig. 1). In the first phase, beacon nodes broadcast their position so they can be used as reference nodes. In the second phase, a node estimates its distance to the reference nodes by using, for example, the received signal strength indicator (RSSI) [5]. In the third phase, the node computes its position by using multilateration [5], and becomes a settled node. In the final phase, the node becomes a reference, and broadcasts its estimated position to assist its neighbors.

By using settled nodes as reference nodes, location errors are propagated. The reason is that the distance estimation process introduce errors in the estimated positions. As a consequence, the most distant nodes of the beacons are likely to have larger errors than the closer ones. In Fig. 1, the location error for node 5 is probably greater than the location error for node 7.

The algorithm attempt to mitigate propagated errors by ignoring the worst references. The references quality is given by the residual value defined as

$$residual(x, y) = \sum_{i=1}^{R} \left(\sqrt{(x_i - x)^2 + (y_i - y)^2} - d_i \right)^2 \tag{1}$$

where R is the number of references, (x, y) is the estimated position, (x_i, y_i) is the ith reference position and d_i is its measured range.



(a) Phase 1 – beacon nodes broadcast their position.

(b) Phase 2 – unknown node estimates its distance to the reference nodes.

(c) Phase 3 – unknown node computes its position and becomes a settled node.

(d) Phase 4 – The new settled node is used as a reference.

Fig. 1 Example and phases of the recursive position estimation (RPE)

The RPE is an algorithm that uses multiple hops to determine the nodes position. Hence, the network topology does not have to follow a special organization, making it suitable for outdoor scenarios.

2.1.2 Directed position estimation

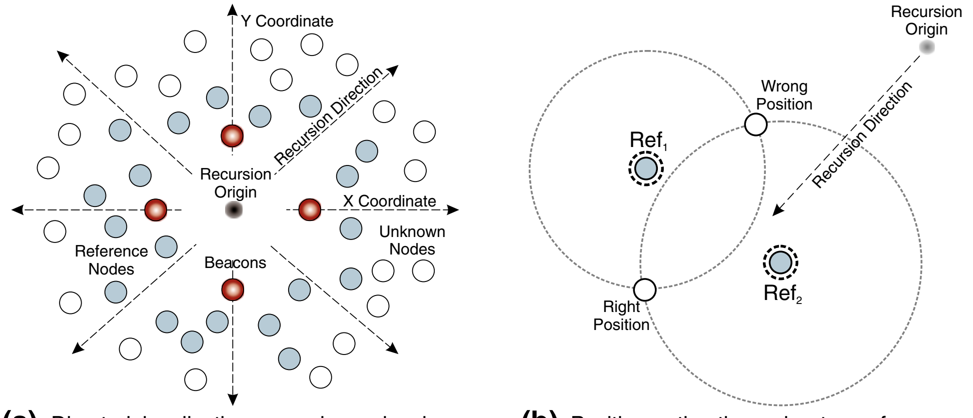
The DPE [24] algorithm is similar to the RPE algorithm. The main idea of the DPE is to start the recursion at a single location, and make it follow a known direction. Then, a node can estimate its position by using only two reference neighbors and the recursion direction. This controlled recursion leads to smaller errors, compared to RPE.

To ensure that the recursion starts at a single point, the algorithm uses a fixed beacon structure. The recursion direction and the beacon structure are depicted in Fig. 2a. This structure has, generally, four beacons that know their distance from the recursion origin and the angle between each pair of beacons. Then, to start the recursion, these beacons inform their positions to their neighbors.

When a node receives the position from two reference neighbors (see Fig. 2b), a pair of possible points results from the system: one is the correct position and the other is the incorrect. Because the direction of the recursion is known, the node can choose between the two possible solutions: the most distant point from the recursion origin is the correct choice.

The algorithm is divided into four phases. In the first phase, beacon nodes start the recursion from a single location. In the second phase, a node chooses two reference points: the pair of nodes with the largest distance between them, and closest to the recursion origin. In the third phase, the node estimates its position. This position is estimated by intersecting the two circles and choosing the most distant point from the recursion origin. In the last phase, the node becomes a reference by sending its information to its neighbors.

Fig. 2 The directed position estimation



(a) Directed localization recursion using beacon structure.

(b) Position estimation using two references. Two possible solution result from the system. The most distant point from the recursion origin is the right position.

The recursion direction can occasionally become wrong. To a correct estimation it is necessary to avoid two possible situations: (a) when the unknown node is closer to the recursion origin than one of the two reference nodes; and (b) when both reference nodes are aligned with the recursion origin. These two scenarios can be detected by comparing the distances from the possible solutions to the recursion origin with the distances from the reference nodes to the recursion origin.

The DPE also propagates localization errors, due to distance estimation errors. However, propagated errors are considerably smaller. Oliveira et al. [24] compare the performance of the DPE with the RPE in several aspects. Their results show that the DPE outperforms RPE in many cases. The DPE works with sparse network, needs fewer beacons, and have smaller errors.

2.2 Target tracking

Target tracking algorithms aim at estimating current and future (next) location of a target. These algorithms are exposed to different sources of noise, introduced by the measurement process and also errors in nodes' location that are used to estimate the target coordinates. Therefore, information fusion [20] is commonly used for filtering such noise sources. Two popular algorithms for this problem are the Kalman and Particle filters.

Several tracking solutions are based on Kalman filters (KF). The reason is that Kalman filters have been used in algorithms for source localization and tracking, especially in robotics [20]. Li et al. [16] propose a source localization algorithm for a system equipped with asynchronous sensors, and evaluate the performance of extended Kalman filter (EKF) [35] and unscented Kalman filter (UKF) [14] for source tracking in non-linear systems. Olfati-Saber [23] proposes distributed Kalman filtering (DKF), in which a centralized KF is decomposed into micro-KFs, so that the distributed approach has a performance equivalent to centralized KF.

Particle filters are popular for modeling non-linear systems subject to non-Gaussian noise. Vercauteren et al. [32] propose a collaborative Particle Filter for jointly tracking several targets and classifying them according to their motion pattern. Arulampalam et al. [3] assess the use Particle filters and the EKF for tracking applications. Considering sensor networks, Rosencrantz et al. [26] developed a Particle Filter for distributed information fusion applied to decentralized tracking. Jiang and Ravindran [13] propose a completely distributed Particle Filter for target tracking in sensor networks, where the communication cost to maintain the particles on different nodes and propagate along the target trajectory is reduced.

Souza et at. [29] assess the performance of target tracking algorithms when position information is provided by localization algorithms. The authors combine the KF and PF with RPE and DPE. In this work, we combine the same algorithms, but we use data fusion of multiple distance estimates during the localization process to improve the target-tracking accuracy.

There are also other distributed approaches for target tracking that are based on cluster [6,33,38] and tree [17,31,37] organizations for in-network data processing.

2.2.1 *Kalman filter*

The Kalman filter is a popular fusion method used to fuse low-level redundant data [20]. If a linear model can describe the system and the error can be modeled as a Gaussian noise, than the Kalman Filter recursively retrieves statistically optimal estimates.

This method, as depicted in Fig. 3a, causes at each discrete-time increment a linear operator application in the current state to generate the new state. The filter considers measurement noise and, optionally, information about the controls on the system. Then, another linear operator, also subject to noise, generates the observed outputs from the true state.

The Kalman filter estimates the state $\mathbf{x}$ of a discrete-time k controlled process that is ruled by the state-space model

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \tag{2}$$

with measurements $\mathbf{y}$ represented by

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, \tag{3}$$

in which $\mathbf{A}$ is the state transition matrix, $\mathbf{B}$ is the input control matrix that is applied to control vector $\mathbf{u}$, $\mathbf{C}$ is the measurement matrix; $\mathbf{w}$ represent the process noise and $\mathbf{v}$ the measurement noise, where these noise sources are represented by random zero-mean Gaussian variables with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ respectively.

Based on the measurement $\mathbf{y}$ and the knowledge of the system parameters, the estimate of $\mathbf{x}$, represented by $\hat{\mathbf{x}}$ is given by

$$\hat{\mathbf{x}}_{k+1} = (\mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k) + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k), \tag{4}$$

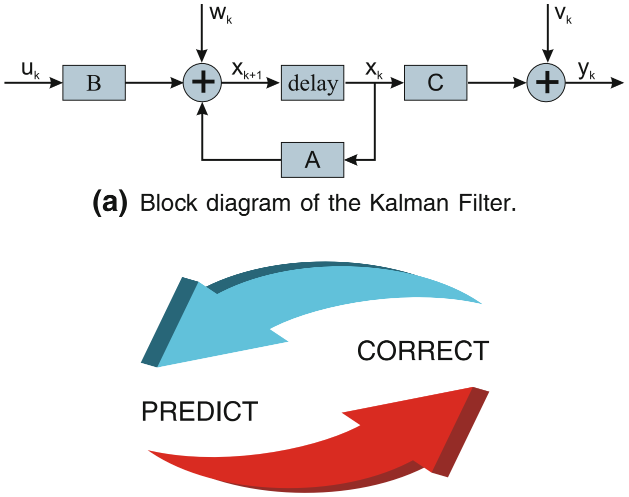
in which $\mathbf{K}$ is the Kalman gain determined by

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{C}^T(\mathbf{C}\mathbf{P}_k\mathbf{C}^T + \mathbf{R})^{-1}, \tag{5}$$

while P is the prediction covariance matrix that can be determined by

$$\mathbf{P}_{k+1} = \mathbf{A}(\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_k\mathbf{A}^T + \mathbf{Q}. \tag{6}$$

The Kalman filter has two phases (see Fig. 3b): time-update (predict) and measurement-update (correct). The time-update is responsible for obtaining the a priori estimates for the next time step and consists of the Eqs. (2) and (3).



(a) Block diagram of the Kalman Filter.

(b) Kalman Filter phases.

Fig. 3 Kalman filter representation and phases

The measurement-update is responsible for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate and consists of the Eqs. (4), (5), and (6) [20]. These phases form a cycle that is maintained while the filter is fed by measurements.

Since many problems cannot be represented by linear models, algorithms have emerged based on the original Kalman Filter formulation to allow these problems to be treated. The major variations of the Kalman filter for non-linear problems are the extended Kalman filter (EKF) [10] and the unscented Kalman filter (UKF) [14]. The EKF is the most popular alternative to non-linear problems. This method uses a linearized model of the process using Taylor series, because this is a sub-optimal estimator. The UKF performs estimations on non-linear systems without the need to linearize them, because it uses the principle that a set of discrete sampling points can be used to parameterize the mean and covariance. The quality of UKF estimates are close to standard KF for linear systems.

Finally, the Kalman filter model allows the elaboration of an algorithm to estimate the optimal state vector values. Thus, it is possible to generate a sequence of state values in each time unit, predicting future states using the current state, and allowing the creation of systems with real-time updates.

2.2.2 Particle filter

Particle Filters are recursive implementations of sequential Monte Carlo (SMC) methods [3]. Although the Kalman filter is a classical solution, Particle Filters represent an alternative for applications with non-Gaussian noise, especially when computational power is rather cheap and sampling rate is slow.

Unlike of the linear/Gaussian problems, the calculations of the posterior distribution of non-linear/non-Gaussian problems are extremely complex. To overcome this difficulty, the Particle Filter adopts an approach called sampling importance. The goal is to estimate the posterior probability density, representing it as a set of particles.

This method attempts to build the posterior probability density function (PDF) based on a large number of random samples, called particles. These particles are propagated over time, sequentially combining sampling and resampling steps. At each time step, the resampling is used to discard some particles, increasing the relevance of regions with high posterior probability. Each particle has an associated weight that indicates the particle quality. Then, the estimate is the result of the weighted sum of all particles.

The resampling step is the solution adopted to avoid the degeneration problem, where the particles have negligible weights after several iterations. The particles of greater weight are selected and serve as the basis for the creation of the new particles set. Furthermore, the minor particles disappear and do not originate descendants.

As the Kalman filter, the Particle filter algorithm has two phases: prediction and correction. In the prediction phase, each particle is modified according to the existing model, including the addition of random noise in order to simulate the effect of noise. Then, in the correction phase, the weight of each particle is reevaluated based on the latest sensory information available, so that particles with small weights are eliminated (resampling process).

3 Proposed approach

In the evaluation presented later in this work, we show that errors introduced by the localization algorithms are not successfully filtered by the tracking algorithm (Kalman and Particle filters), because the node position errors are not perceived as noise by the filters.

An alternative to reduce the tracking error is reducing localization errors. By reducing the localization error, we make tracking algorithms closer to their ideal operating conditions. Thus, we use a Kalman Filter to improve distance estimation errors and, consequently, localization and tracking accuracy.

In this approach, during the localization process, several distance estimates are performed, that is, each reference node reports its position k times to its neighbors. Nodes receiving these packages create an Kalman Filter instance for each reference. Then, all distance estimates are refined by the corresponding Kalman filters. Thus, the filter obtains a more accurate distance estimate, improving the localization result (Fig. 4). Then, this improved estimate is used by the target tracking algorithm.

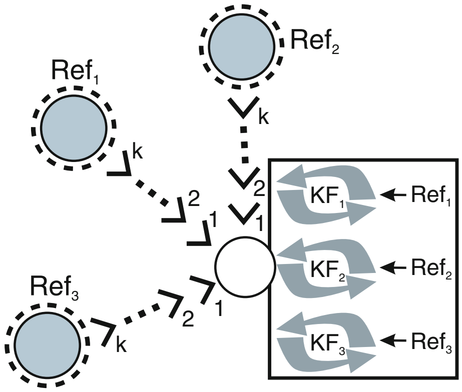


Fig. 4 Fusion of k distance estimations to improve the target tracking performance. For this task, the unknown node create a unique Kalman filter instance for each reference

In this task, the Kalman filter goal is to obtain a constant (distance) estimate. The linear system of the filter is very simple and can be configured as

$$\begin{cases} x_{k+1} = d_{k+1} = d_k + w_k \\ y_k = d_k + v_k \end{cases} \tag{7}$$

in which x and d represent the state, distance in this case, of the discrete-time k; y is a measurement value; w and v represent the process and measurement noise, respectively.

Filtering the distance estimates during localization is a simple process that ensures good results. The distance estimates errors are the largest contributors to overall error of the localization system, and only a small fraction of this error is generated by position computation and localization algorithm [4,24]. Some algorithms try to isolate the distance estimate errors by selecting the best references based on a residual value [2], however this technique is not very efficient, because all references can cause distance estimate errors. Since the node position is calculated and used by any application, such as target tracking, it is difficult to determine the error size and direction, so the best option is to work on the error source.

4 Evaluation

In this section, we evaluate the performance of the KF and PF using the information position provided by the RPE and DPE localization algorithms. We apply the proposed approach in the localization process, where several distance estimates are used to verify its performance.

4.1 Methodology

The evaluation methodology is divided into five phases, as shown in Fig. 5. First, there is a newly deployed sensor network with some beacon nodes, where most of the nodes do not know their position (unknown nodes), so this network must be prepared to track the target. In the second phase, a localization algorithm is applied (RPE or DPE), and during this step, several distance estimates can be used to reduce the localization errors and improve the tracking accuracy, following the proposed approach (see Sect. 3). In the third phase, nodes know their position, so when three or more nodes detect the target, they compute its position with multilateration. In the fourth phase, the nodes send the target position to the sink node. In the final phase, the sink node predicts the next target position and reduces the measurement noise, performing the tracking algorithm (Kalman filter or Particle filter). While there are measurements, the target tracking continues (back to phase three).

The experiments were performed by simulation (implemented in Java), where the sensor field is composed of n sensor nodes, with a communication range of r_c, that are distributed in a two-dimensional squared sensor field $Q = [0, s] \times [0, s]$. As a proof-of-concept, we consider symmetric communication links, i.e., for any two nodes u and v, u reaches v if and only if v reaches u. Thus, we represent the network by the Euclidean graph $G = (V, E)$ with the following properties:

- $V = \{v_1, v_2, \ldots, v_n\}$ is the set of sensor nodes;
- $\langle i, j \rangle \in E$ iff v_i reaches v_j, i.e. the distance between v_i and v_j is less than r_c.

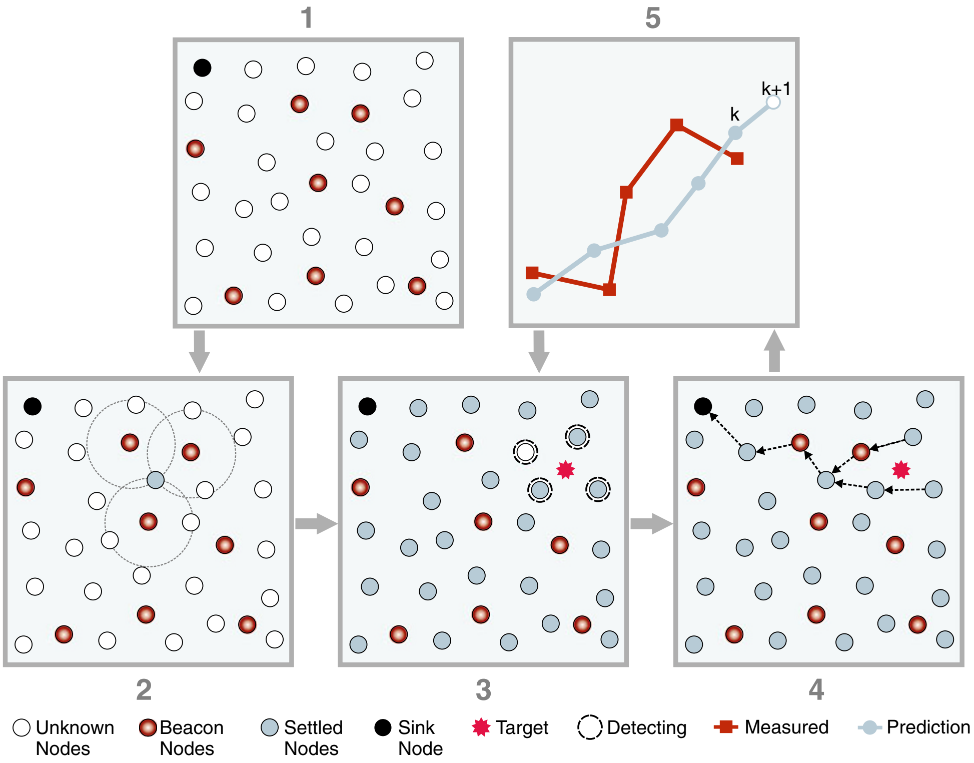
To detect the target, we use the binary detection model [11,34]. In this model, for a given event e (target presence), every sensor v, whose distance d between it and the target is smaller than a detection radius r_d, assuredly detects the event. Then, the probability of a sensor node to detect an event is defined as

$$P(v, e) = \begin{cases} 1, & \text{if } d \leq r_d \\ 0, & \text{otherwise} \end{cases}. \tag{8}$$

The default network configuration is composed of $n = 150$ sensors nodes randomly distributed on a $Q = [0, 70] \times [0, 70]$ m^2 sensor field. The communication and detection ranges are $r_c = r_d = 15$m for every node. This configuration defines a network density of 0.03 nodes/m^2, which is sufficient for the majority of nodes to have their locations estimated by both RPE and DPE algorithms. Oliveira et al. [24] use this same configuration in their experiments to evaluate the localization systems. Therefore, we adopted this configurations to estimate the nodes position and tracking the target.

Node locations are estimated by RPE or DPE. In the RPE algorithm, 5 % of the nodes are beacons, while DPE always uses four beacons. To simulate the inaccuracies of the distance estimations, usually obtained by RSSI, time of

Fig. 5 Methodology phases: (*1*) a newly deployed sensor network with beacon and unknown nodes; (*2*) a localization algorithm is applied, moreover several distance estimates are used to improve the tracking accuracy; (*3*) three or more nodes detect the target and compute its position; (*4*) the target position is sent to the sink node; (*5*) the sink node performs the tracking algorithm for predict the future target position and reduce the measurement noise, then back to phase three



arrival (TOA) and time difference of arrival (TDoA) [8,9], each range sample is disturbed by a zero-mean Gaussian variable with standard deviation equals to 5 % of the distance. This assumption is reasonable and leads to non-Gaussian errors in the localization algorithms [25]. During the localization algorithm, we vary the number of distance estimates used by the Kalman Filter as 1, 10, 20, 50, 100, and 200 measurements for each reference.

The target tracking is performed with Kalman or Particle filters. The Kalman filter has its linear system equations represented by

$$\begin{cases} x_{k+1} = \begin{bmatrix} px_{k+1}\\ py_{k+1}\\ vx_{k+1}\\ vy_{k+1}\end{bmatrix} = \begin{bmatrix}1&0&T&0\\0&1&0&T\\0&0&1&0\\0&0&0&1\end{bmatrix} \times \begin{bmatrix}px_k\\py_k\\vx_k\\vy_k\end{bmatrix} + w_k \\ y_k = \begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix}\times\begin{bmatrix}px_k\\py_k\\vx_k\\vy_k\end{bmatrix} + z_k\end{cases} \tag{9}$$

in which x represents the state of a discrete-time k, composed by the position (px, py) and velocity (vx, vy); y is a measurement value; w and v represent the process and measurement noise, respectively.

The Particle filter uses 1,000 particles. This value was set based on a previous empirical tests that showed that more than 1,000 particles do not improve tracking significantly. The Particle filter used in the experiments is represented by Algorithm 1.

For illustration purposes, the Particle Filter algorithm presented considers only one dimension, in which x is the position, v is the velocity and w is the weight of each N particles in a discrete-time k; y is the input measurement value. First, the algorithm randomly distributes the particles (line 2). The particle propagation and the calculus of their importance consider the distance from each particle to the measurement position (lines 4–10). The normalization process (line 12) prepares the particles weights for the resampling process (lines 14–21). Finally, the prediction of the position is calculated (line 23).

For the sake of simplicity, we consider an uniform movement, so that the movement is modeled by a linear system, suitable for both Kalman and Particle filters. The target trajectory is composed of 1,000 points to consider a significative sample. The distance between the points of the trajectory is 0.1 m (uniform motion) to keep the target within the area monitored. The interval between each measurement is $T = 1$s. The maneuvers of the target are determined by an angle randomly generated within $-25°$ and $25°$ every 25 steps.

```
input  : The measured $y_k$
output: The prediction $x_{k+1}$
// Initialize the particles
for $i = 1 : N$ do $x_0^i \leftarrow random()$;
// Sample particles from importance density and compute
weights
$totalWeight \leftarrow 0$;
for $i = 1 : N$ do
    $x_k^i \leftarrow x_{k-1}^i + v_{k-1}^i + gaussian()$;
    $v_k^i \leftarrow v_{k-1}^i + gaussian() * 0.05$;
    $w_k^i \leftarrow 1/distance(y_k, x_k^i)$;
    $totalWeight \leftarrow totalWeight + w_k^i$;
end
// Normalize weights
for $i = 1 : N$ do $w_k^i \leftarrow w_k^i / totalWeight$;
// Resampling
$slice_k^0 \leftarrow w_k^0$;
for $i = 2 : N$ do $slice_k^i \leftarrow slice_k^{i-1} + w_k^i$
for $i = 1 : N$ do
    $c \leftarrow random()$;
    $j \leftarrow 0$;
    while $j < N - 1 andslice_k^j < c$ do $j \leftarrow j + 1$;
    $resampling_k^i \leftarrow particle_k^j$;
end
// Result
for $i = 1 : N$ do $x_{k+1} \leftarrow x_{k+1} + x_k^i * w_k^i$;
$returnx_{k+1}$;
```

Algorithm 1: *Particle Filter.*

In Figs. 6–12, each point is plotted as an average of 100 random topologies to ensure a lower variance in the results. The error bars represent the confidence interval of 99 %.

4.2 Simulation results

4.2.1 Target tracking behavior

To illustrate the behavior of the tracking algorithms, we show some snapshots in this section. In these snapshots, the RPE algorithm could not find the location of two nodes (from 150 nodes), and the average error of the node locations is 3.49 m. Adopting the same instance, the DPE algorithm managed to estimate the location of every node, and the average location error is 2.56 m.

These two scenarios are compared with the ideal setting, in which the localization system is perfect. For all cases, the performance of Kalman and Particle filters are presented. The results are summarized in Table 1.

The Fig. 6a–c shows a target moving through the sensor field (red line). Orange points represent measurements and blue points are the results of Kalman Filter tracking algorithm. Figure 6d–f shows the error calculated from real target and Kalman Filter estimation for each measured point. Figure 6g–l represents the same case using the Particle Filter tracking algorithm instead. These figures illustrates the influence of the localization errors caused by each algorithm. In general, the greater the localization error, the greater the tracking error, independent of the tracking algorithm. The influence of localization errors is clearly visible in the region around point (20, 15) in Fig. 6c, in which localization errors lead to a wrong track estimation.

Tracking with Kalman Filter has better results when node's location information are ideal, or when they are estimated by the DPE. However, when the node locations are estimated by the RPE, the Particle Filter presents the best results. The reason is that the Particle Filter is less affected by measurement errors, this fact becomes clear in the following sections.

As a general conclusion, Fig. 6 shows that both filters successfully reduce the errors resulting from the estimation of the target location, but errors resulting from the localization algorithms are not significantly filtered.

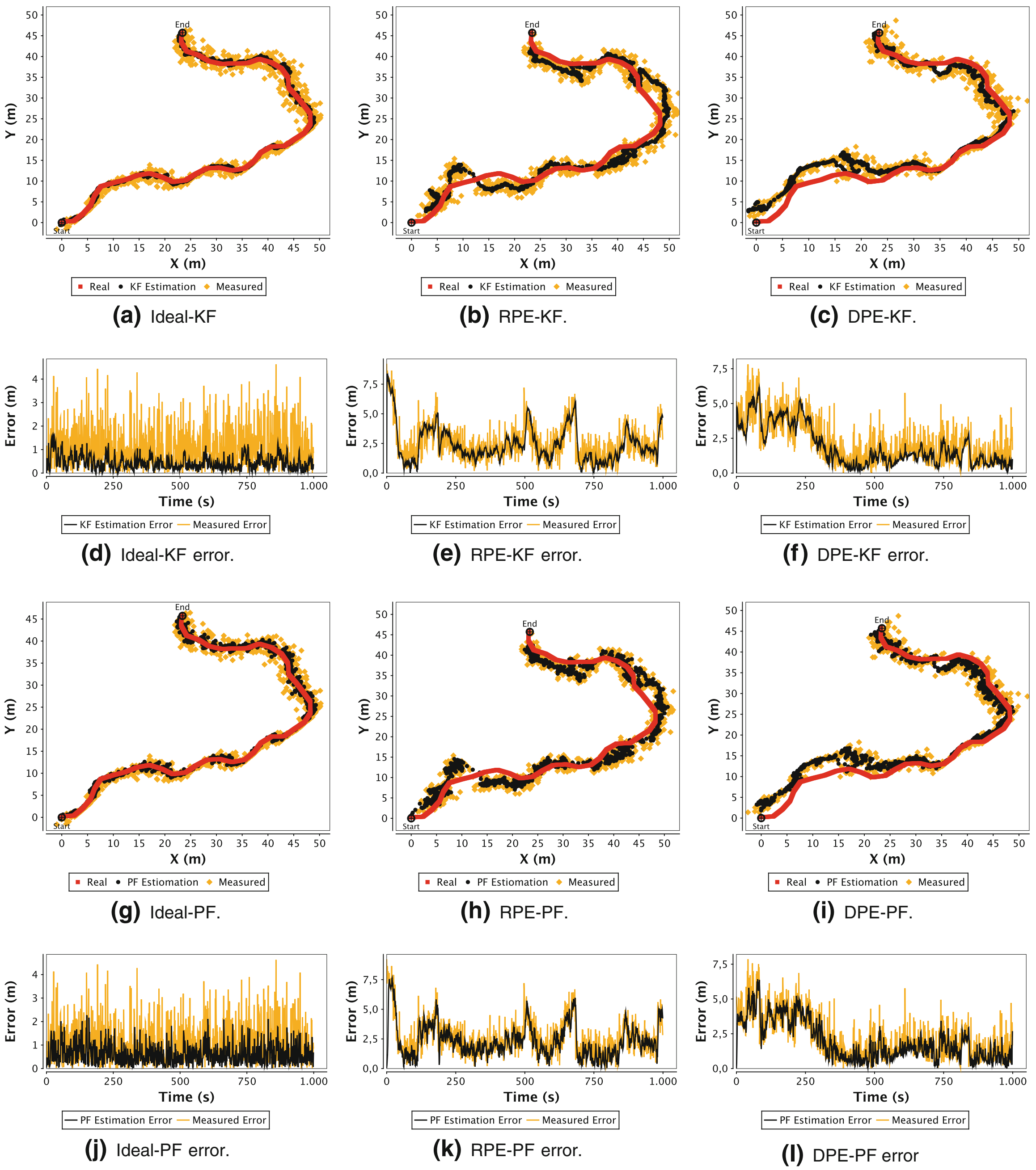
4.2.2 Costs and benefits of multiple distance estimates.

More distance estimates during the localization process can reduce the localization and tracking errors. However, it is necessary to send additional packets, i.e., more resources will be consumed to get this benefit. Therefore, in this section we evaluate the costs and benefits of using several distance estimates. This analysis is important, since it helps define how much you should spend for a given performance in the target tracking.

Both the RPE and DPE have the communication complexity of $O(n)$, where n is the number of nodes. The Fig. 7a shows the number of packets sent when the number of distance estimates increases. Using k distance estimates causes each beacon and settled node to broadcast its position k times, increasing the communication complexity to $O(kn)$. This figure also shows that the RPE sends fewer packets than the DPE. This occurs because of the network density used in the experiment that lead RPE, in some topologies, to estimate fewer node positions than DPE, so these nodes do not broadcast their location information, reducing the number of packets sent.

The Fig. 7b shows the improvement in accuracy of tracking when the number of distance estimates increase. Using 10 distance estimates already provides a significant improvement when compared to results obtained with only 1 estimation. In the target tracking using the RPE, the Particle Filter has better results, because this algorithm reduces a small fraction of non-Gaussian noise introduced by localization algorithm. In target tracking with DPE, the Kalman Filter becomes more accurate with more than 10 distance estimates, because the localization error is pretty low, so that the Kalman Filter starts to operate under ideal conditions.

It is not feasible to use a very high number of distance estimates, because the benefit achieved becomes lower in



(a) Ideal-KF **(b)** RPE-KF. **(c)** DPE-KF.

(d) Ideal-KF error. **(e)** RPE-KF error. **(f)** DPE-KF error.

(g) Ideal-PF. **(h)** RPE-PF. **(i)** DPE-PF.

(j) Ideal-PF error. **(k)** RPE-PF error. **(l)** DPE-PF error

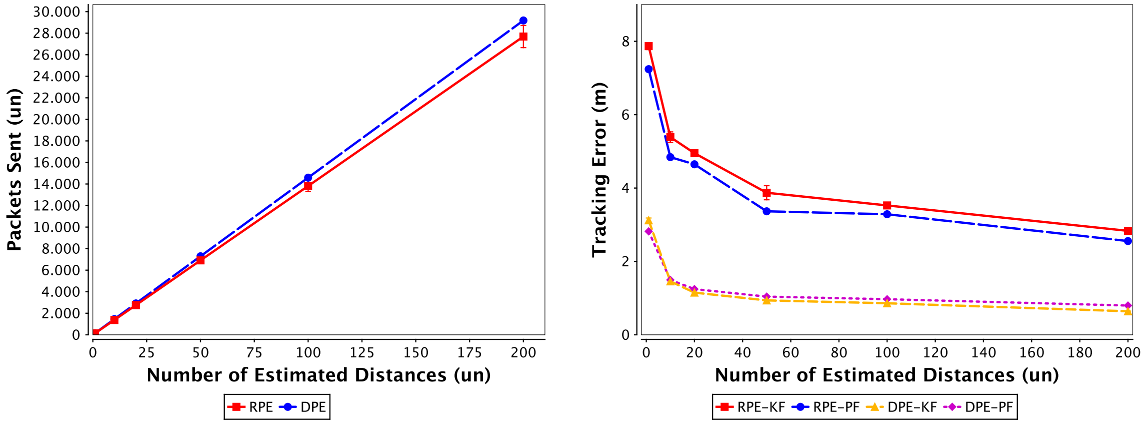
Fig. 6 Performance of Kalman and Particle filters using the RPE and DPE localization algorithms

comparison with the required cost. Using 10 distance estimates is enough to achieve improvements of 50 % with a reasonable cost. Until 50 distance estimates can significantly reduce the tracking error, but most estimates (100 and 200) lead to a little improvement of results with high cost.

4.2.3 Impact of distance estimation inaccuracy

Distances estimated by sensor nodes are not perfect. Depending on the monitored environment, the associated errors can be greater, which affects the tracking performance. In general, these errors can be modeled by a zero-mean Gaussian variable, in which the standard deviation is a percentage of the actual distance [4].

Thus, to evaluate different situations, we vary the standard deviation from 0 to 15 % of the distance (for RPE and DPE estimation processes). A standard deviation of 0 % corresponds to a perfect distance estimate. Deviations between 0 and 8 % can represent the estimates obtained by techniques that use time of arrival of the signal as TOA and TDoA, which



(a) Packets Sent.

(b) Improved tracking.

Fig. 7 Costs and benefits of multiple distance estimates

have errors smaller than 1 m. While larger deviations represent errors obtained by more imprecise methods, like RSSI.

Figure 8a–d presents the error performance by varying the distance estimation inaccuracy and the number of estimates, showing 3D graphs with the combination of RPE and DPE with KF and PF. Figure 8e, f shows the cases of 1 and 50 distances estimates, respectively.

When the distance estimation inaccuracy is low (between 5 and 10 % of the distance), the accuracy improvement of the target tracking using DPE is negligible, regardless of the number of distance estimates used. When this imprecision is high (between 15 and 30 %), with 50 distances estimates or more, note that the average error converges to 1 m (Fig. 8b, d). However, with RPE, the improvement is noticeable even when the distance estimates inaccuracy is low (Fig. 8a, c).

It is also interesting to note in Fig. 8e, f that the Particle filter outperforms the Kalman Filter, especially when RPE is chosen as the localization algorithm. The reason is that the non-linearity and non-Gaussian nature of the Particle Filter results in reducing a small fraction of the non-Gaussian noise introduced by the localization algorithm.

4.2.4 Impact of the network density

The impact of the density of the network is evaluated by increasing the number of nodes in the same sensor field, so that the network density varies from 0.03 to 0.07 nodes/m^2. The smallest density used in this experiment allows both the RPE and DPE algorithms to estimate the location of the most of the sensor nodes.

In this case, Fig. 9 shows that, for the DPE algorithm, the target tracking error remains constant independent of the network density. The reason is that the same beacon structure is used regardless the network density. However, for the RPE algorithm, the number of beacons increase with the network density, because we ensure that 5 % of the nodes are beacons. As a result of the increasing number of beacons, the tracking error reduces accordingly.

Multiple distance estimates are important for improving the target tracking accuracy when the RPE is used, especially in sparse networks, as show in Fig. 9a and c. With the DPE instead, the network density does not interfere in the target tracking error. Therefore, for 50 distance estimates or more the average error converges to 0.7m (Fig. 9b, d).

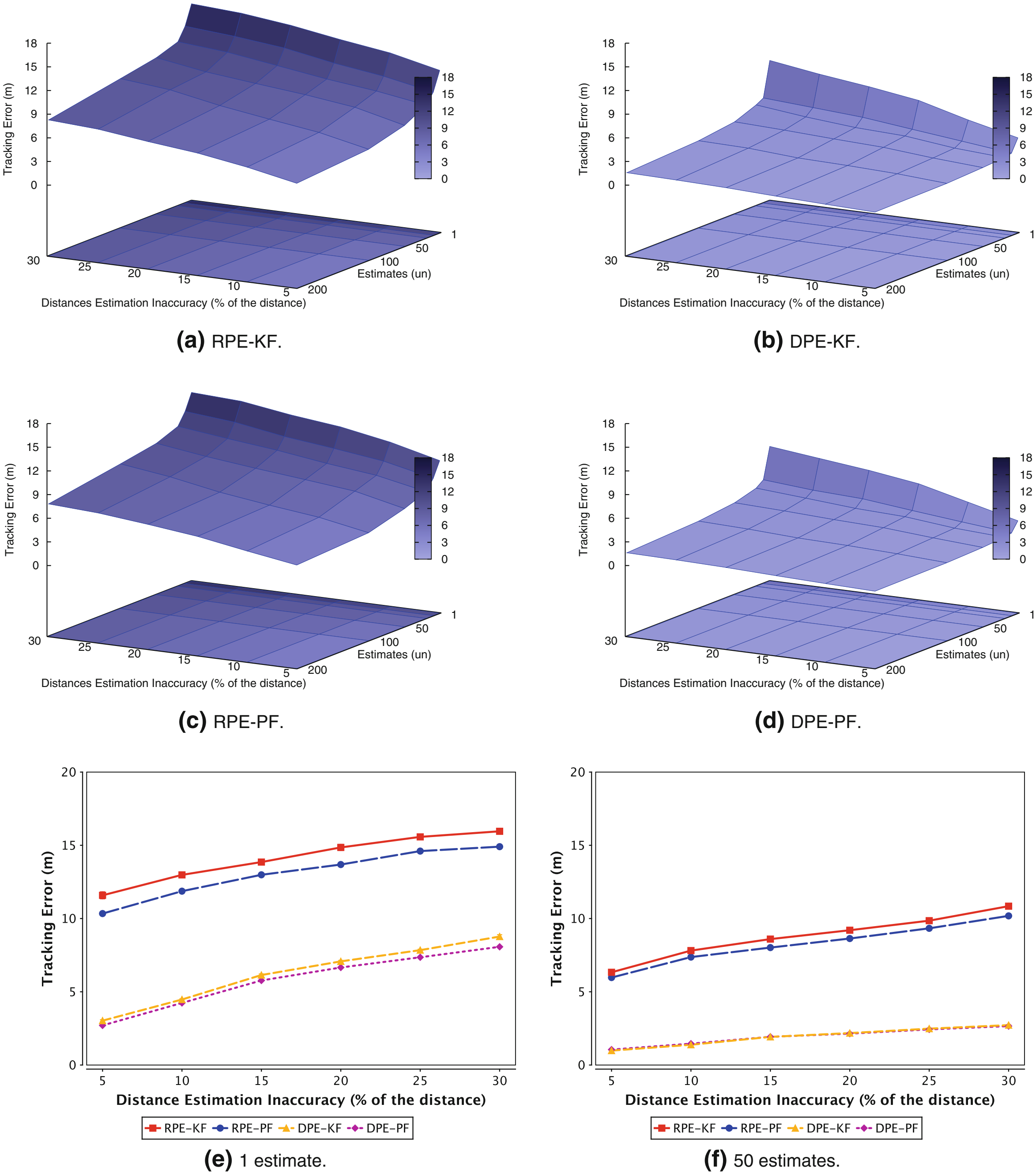
Figures 9(e and f) show that with a single distance estimate during the location process, the Particle Filter is slightly better with both RPE and DPE. The reason is that it filters a small fraction of the non-Gaussian localization errors. When we use 50 distance estimates, the performance of the Kalman Filter becomes equivalent to the Particle Filter in the case of RPE and better in the case of DPE, since the Kalman Filter operates under ideal conditions with low localization errors.

4.2.5 The impact of the network scale

In this section, we evaluate how the network scale affects the combinations of localization and tracking algorithms.

In this context, we vary the number nodes from 100 to 350, while keeping a constant density of 0.03 nodes/m^2. Therefore, the monitored area is resized according to the number of sensor nodes. As the percentage of beacons used by RPE is 5 %, the number of beacons also increases according to the number of nodes in this case. The DPE keeps using only a single structure of four beacons.

Figure 10 shows that increasing the network scale, the tracking errors with DPE increase accordingly. The reason is that a higher number of nodes generates a higher propagation of position error, since the same number of beacons is maintainedregardless the number of nodes. However,



(a) RPE-KF.

(b) DPE-KF.

(c) RPE-PF.

(d) DPE-PF.

(e) 1 estimate.

(f) 50 estimates.

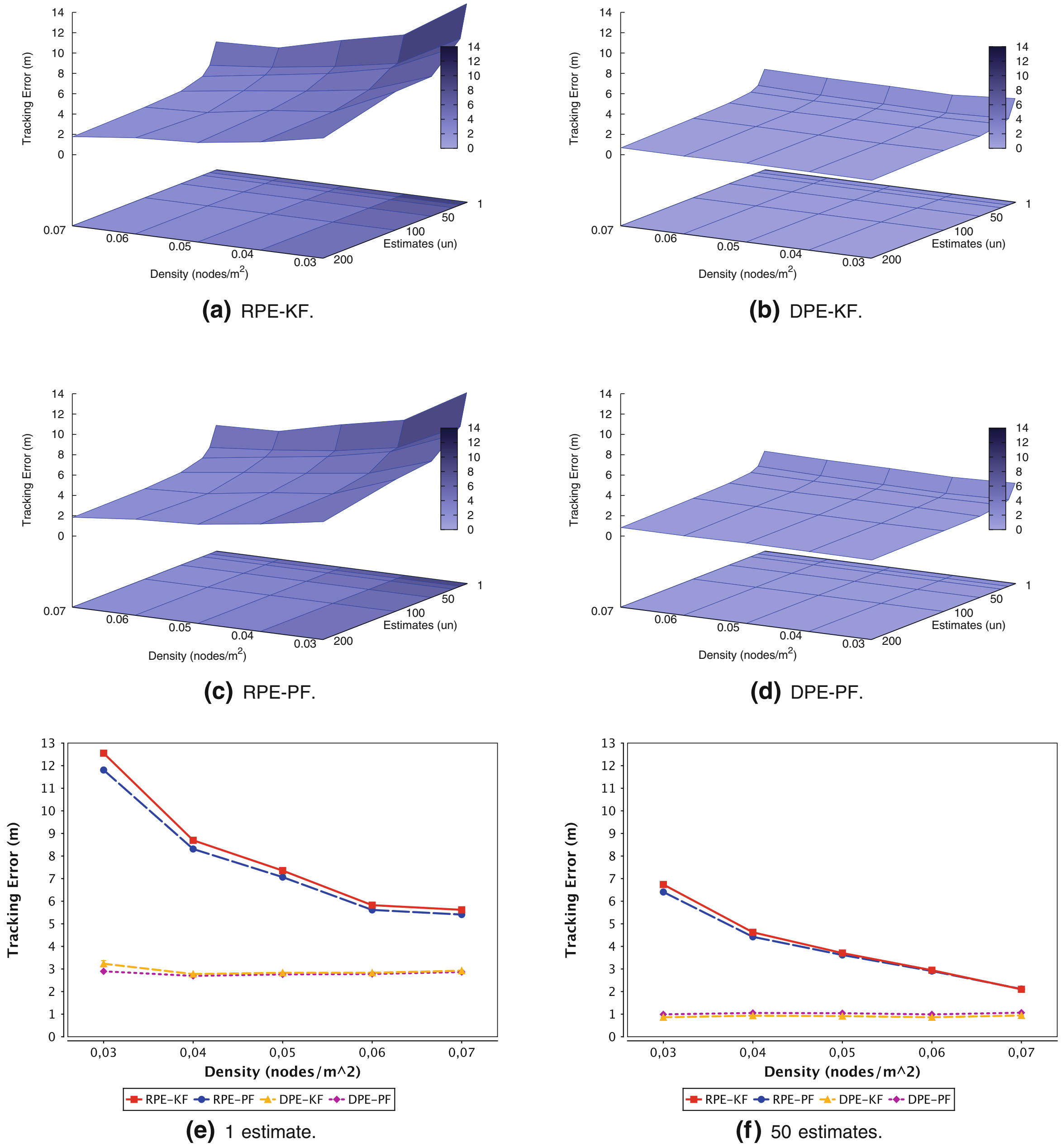
Fig. 8 Impact of distance estimation inaccuracy

with the RPE, the tracking errors remains almost constant, because the number of beacons increase with the network scale (cf. [24]).

When there are many nodes in network (between 250 and 350), using multiple distance estimates in DPE improves significantly the target tracking accuracy, however they have little influence when there are few nodes (Fig. 10b, d). With RPE instead, the usage of multiple distance estimates is important for any number of nodes (Fig. 10a, c).

4.2.6 The impact of the number of beacons

The number beacons used by the DPE and RPE lead to different localization errors. Therefore, affecting the tracking solutions. For the DPE, Oliveira et al. [24] show that increasing the number of beacon nodes of the structure does not improve significantly the result of the localization. Therefore, this evaluation considers only the RPE algorithm.



(a) RPE-KF.

(b) DPE-KF.

(c) RPE-PF.

(d) DPE-PF.

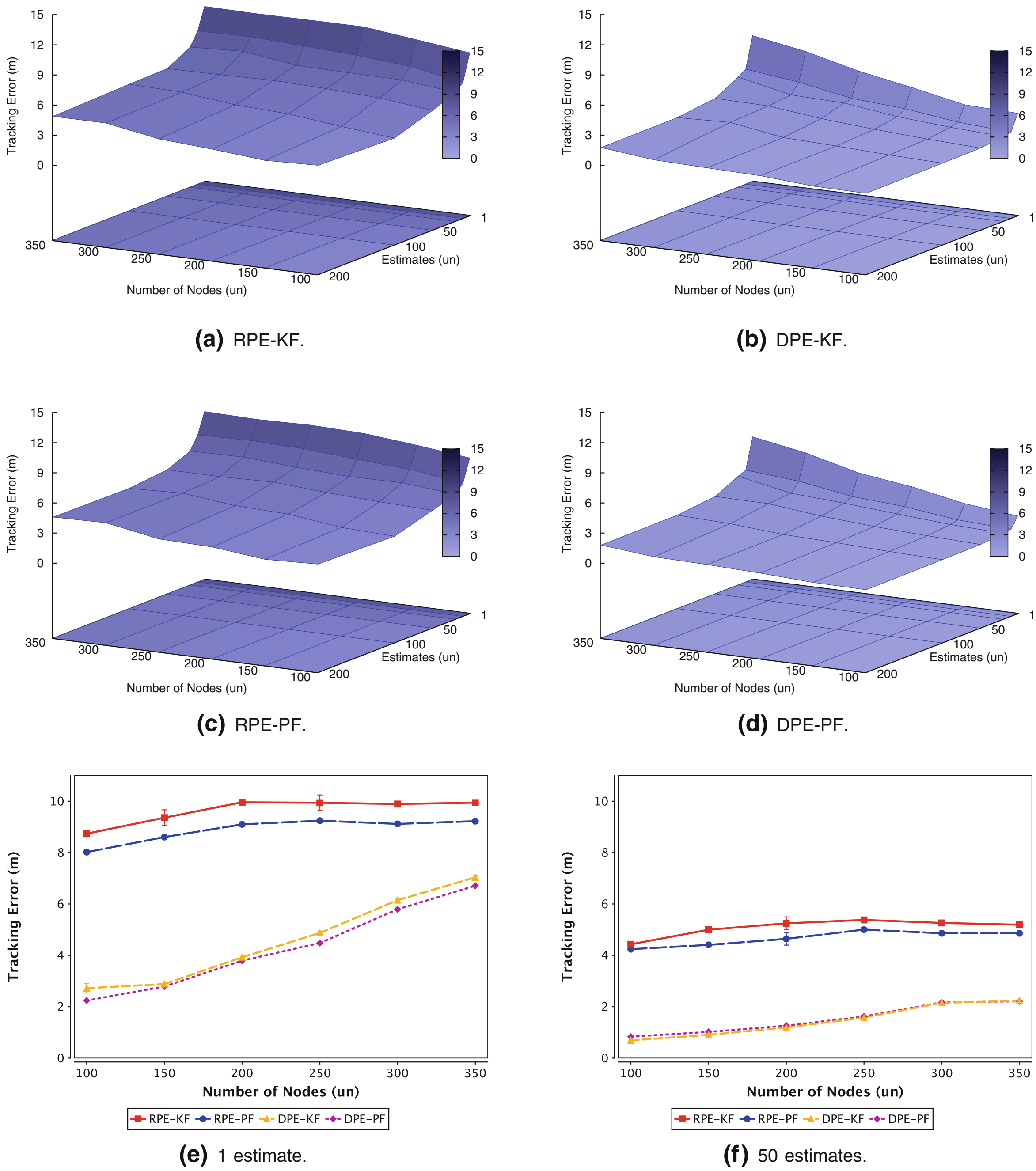
(e) 1 estimate.

(f) 50 estimates.

Fig. 9 Impact of the network density

In this experiment, the number of beacons is increased from 5 to 35 % of the total number nodes. A greater number of beacons means that the localization algorithm has more references for estimating the location of remaining nodes, which leads to smaller errors. Hence, the tracking error is also inversely proportional to the number of beacons. This behavior is depicted in Fig. 11.

When the amount of beacon nodes is significantly large, the localization error becomes so small that the Kalman Filter tends to have better results than the Particle Filter, it is up 20 % better with 1 estimate and up 10 % to 50 estimates (Fig. 11c, d). The reason is that the non-Gaussian errors resulting from localization systems are reduced in such a way that the Kalman Filter starts to operate under ideal conditions, which means it converges to the optimal solution for target tracking. When there are few beacons (10 %), multiple distance estimates improves significantly the target tracking accuracy. However, when more than 10 % of the nodes are beacons note that the average error converges to 0.6 m, both for target tracking with Kalman and Particle filters (Fig. 11a, b).



(a) RPE-KF. **(b)** DPE-KF.

(c) RPE-PF. **(d)** DPE-PF.

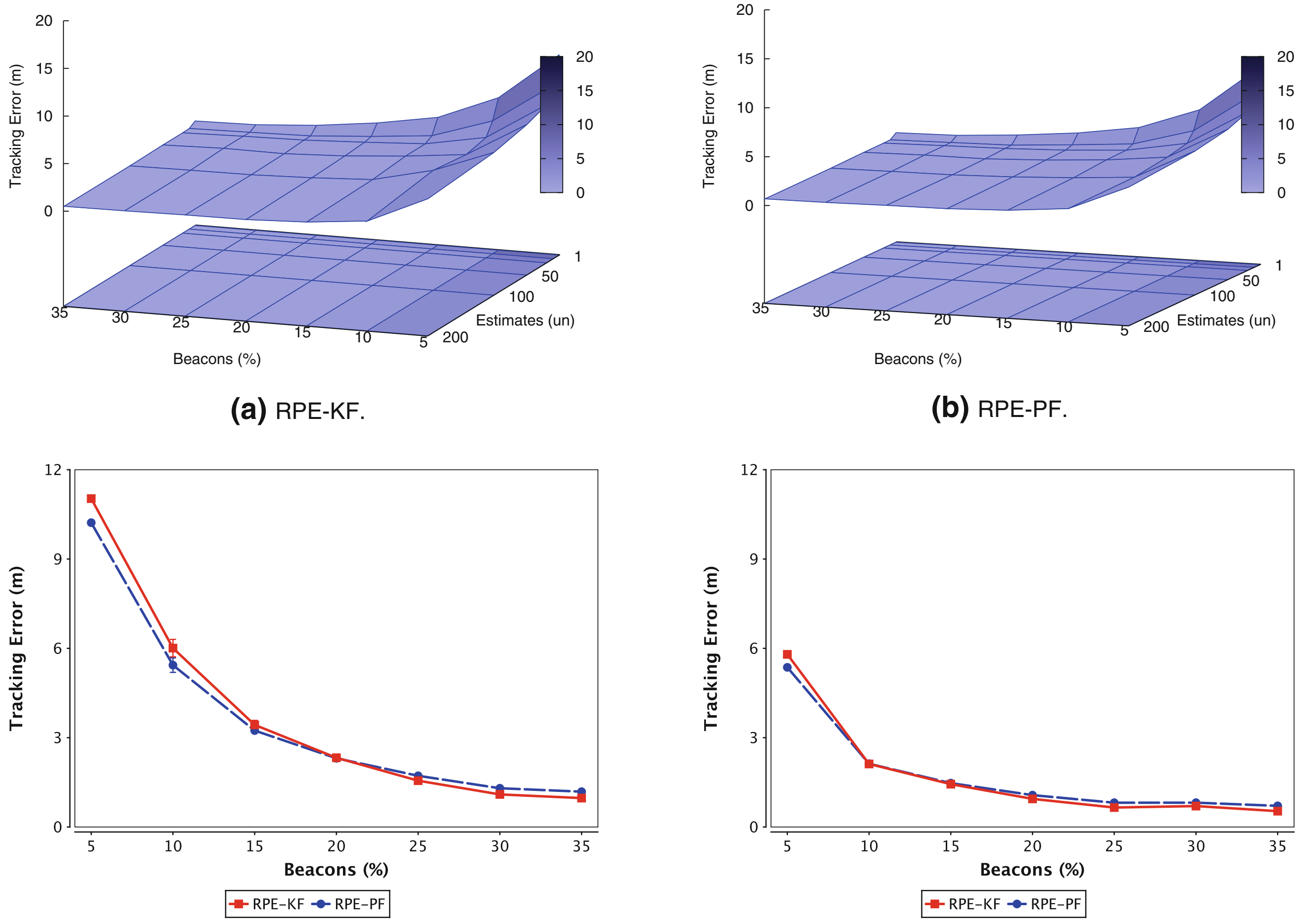
(e) 1 estimate. **(f)** 50 estimates.

Fig. 10 Tracking scale

4.2.7 The impact of the beacon structure

As stated earlier, increasing numbers beacons by structure in the DPE does not improve the localization results. However, the DPE may benefit from multiple beacon structures [24]. Thus, to evaluate the performance of the target tracking algorithms with multiple beacon structures, we vary the number of such structures from 1 to 5. This experiment represents a situation for the DPE algorithm that is analogous to the previous experiment for the RPE algorithm.

By performing a single distance estimate in the localization process (Fig. 12(c)) and using a three-beacon structure, the tracking results with Kalman and Particle filters are very close. However, when more structures are used, the Kalman Filter tracking is favored. The opposite occurs when we use less than three-beacon structures. With 50 distance estimates, the Kalman Filter outperforms the Particle Filter with any number of beacon structures, because the non-Gaussian noise introduced by the location is very low (Fig. 12(d)). Besides, when we use of 50 distance estimates, the average tracking



(a) RPE-KF.

(b) RPE-PF.

(c) 1 estimate.

(d) 50 estimates.

Fig. 11 Impact of the number of beacons

error converges to 0.6 m regardless of the number of beacon structures available (Fig. 12a, b).

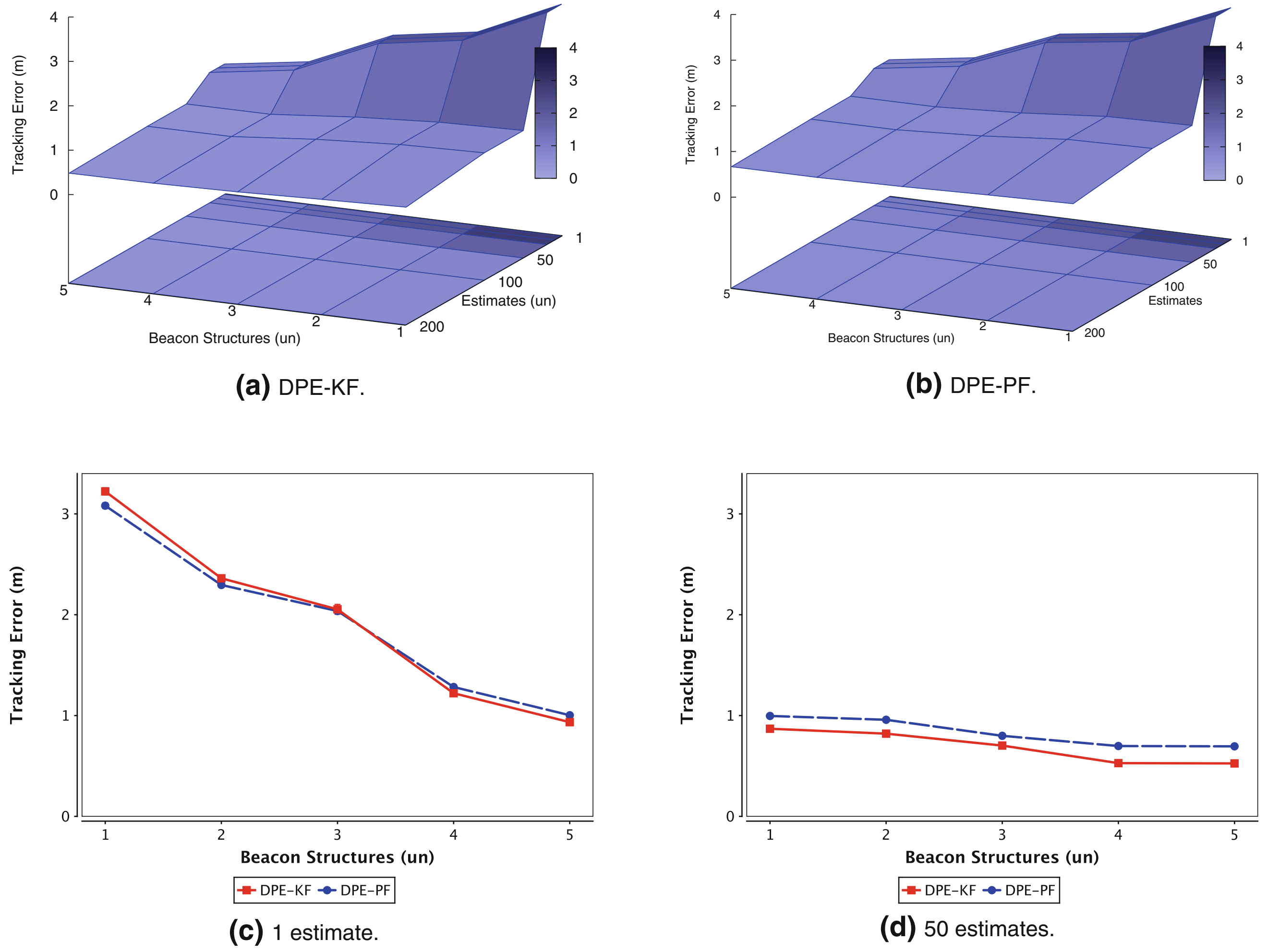
5 Conclusions and future work

In this work, we demonstrated how information fusion can reduce errors during the localization process, while assessing the impact of actual localization algorithms on target tracking algorithms. For these evaluations, we chose the RPE and DPE algorithms to compute node positions, since RPE is a pioneer solution and DPE is more accurate and cheaper than the RPE. The target tracking techniques we have chosen were the Kalman and Particle filters. These filters are very popular and can be considered as canonical solutions for the target tracking problem.

As a general conclusion, by using up to 50 distance estimates ensures better results in the target tracking with a moderate cost. Above this value, the errors converges, so the reduction of errors is small compared to the associated cost. Furthermore, the reduction in localization error using information fusion enhances the performance of Kalman Filter over the Particle Filter, especially when the DPE is used.

Kalman and Particle filters successfully filter the errors associated with the target location estimation. However, the errors introduced by the localization algorithms are not successfully filtered by the tracking algorithm. The reason is that the Kalman Filter is not designed to filter non-Gaussian noise. On the other hand, the Particle Filter is designed to filter non-Gaussian noise. Consequently, the Particle Filter tends to outperform the Kalman as the localization errors increase. However, even the Particle Filter cannot significantly filter the non-Gaussian localization errors. Results show that for tracking applications with severe accuracy constraints, the localization algorithms need to improve their estimations to guarantee the performance of target tracking algorithms.

This work leads to some particularly interesting directions. The first is to properly characterize the localization errors, so that we can understand the expected magnitude, direction, and orientation of the error resulting from localization algorithms. Such knowledge allows us to design new



(a) DPE-KF.

(b) DPE-PF.

(c) 1 estimate.

(d) 50 estimates.

Fig. 12 Impact of the beacon structure

Table 1 Target tracking errors

Localization algorithm	Tracking error (m)	
	KF	PF
RPE	2.39	2.32
DPE	1.87	1.94
Ideal	0.43	0.61

tracking algorithms that use such information to compensate and reduce the impact of localization errors, depending of the localization algorithm used.

Another future direction includes reducing the location algorithms inaccuracy by using all the location information reported to nodes. These algorithms usually provide a minimum number of references (three for the RPE and two for the DPE) required for calculating the node position, ignoring the additional information received after the calculation. This approach can lead to accuracy improvement, and it does not require extra communication. Finally, the cross-layer design of localization and tracking algorithms, not explored yet, may lead to improved solutions for both problems.

Acknowledgments This work is supported by the Brazilian National Council for Scientific and Technological Development (CNPq), under the grant numbers 474194/2007-8 (RastroAM), 55.4087/2006-5 (SAUIM) and 575808/2008-0 (Revelar), and also the the Amazon State Research Foundation (FAPEAM), trough the grant 2210.UNI175.3532.03022011 (Projeto Anura—PRONEX 023/2009).

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor Networks: a survey. Comput Netw 38:393–422
2. Albowicz J, Chen A, Zhang L (2001) Recursive position estimation in sensor networks. In: Proceedings of the 9th international conference on network protocols (ICNP'01), pp 35–41
3. Arulampalam MS, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Trans Signal Process 50:174–188

4. Bachrach J, Eames AM, Eames AM (2005) Localization in sensor networks, chapter 9, pp 277–310. Wiley, New York
5. Boukerche A, de Oliveira HABF, Nakamura EF, Loureiro AAF (2007) Localization systems for wireless sensor networks. IEEE Wireless Commun 14:6–12
6. Chong CY, Zhao F, Mori S, Kumar S (2003) Distributed tracking in wireless ad hoc sensor networks. In: Proceedings of the 6th international conference of information fusion (Fusion'03), pp 431–438
7. Ehsan S, Bradford K, Brugger M, Hamdaoui B, Kovchegov Y, Johnson D, Louhaichi M (2012) Design and analysis of delay-tolerant sensor networks for monitoring and tracking free-roaming animals. Trans Wireless Commun 11(3):1220–1227
8. Fukuda K, Okamoto E (2012) Performance improvement of TOA localization using IMR-based NLOS detection in sensor networks. In: Proceedings of the 26th international conference on information networking (ICOIN'12), pp 13–18
9. Gibson JD (1999) The mobile communication handbook. IEEE Press, New York
10. Grewal MS, Andrews AP (2001) Kalman filtering: theory and practice using MATLAB. Wiley, New York
11. He T, Bisdikian C, Kaplan L, Wei W, Towsley D (2010) Multi-target tracking using proximity sensors. In: Proceedings of the military communications conference (MILCOM'10), San Jose, California, USA, pp 1777–1782
12. He T, Huang C, Blum BM, Stankovic JA, Abdelzaher T (2003) Range-free localization schemes for large scale sensor networks. In: Proceedings of the 9th ACM international conference on mobile computing and networking (MobiCom'03), pp 81–95
13. Jiang B, Ravindran B (2011) Completely distributed particle filters for target tracking in sensor networks. In: Proceedings of the 25th parallel distributed processing, Symposium (IPDPS'11)
14. Julier SJ, Uhlmann JK (1997) A new extension of the kalman filter to nonlinear systems. In: Proceedings of the international aerosense, Symposium (SPIE'97), pp 182–193
15. Kalman RE (1960) A new approach to linear filtering and prediction problems. J Basic Eng 82:35–45
16. Li T, Ekpenyong A, Huang YF (2006) Source localization and tracking using distributed asynchronous sensor. IEEE Trans Signal Process 54:3991–4003
17. Lin CY, Peng WC, Tseng YC (2006) Efficient in-network moving object tracking in wireless sensor networks. IEEE Trans Mobile Comput 5:1044–1056
18. Lin KW, Hsieh MH, Tseng VS (2010) A novel prediction-based strategy for object tracking in sensor networks by mining seamless temporal movement patterns. Expert Syst Appl 37:2799–2807
19. Mazomenos EB, Reeve JS, White NM (2009) A range-only tracking algorithm for wireless sensor networks. In: International conference on advanced information networking and applications workshops (AINAW'07), pp 775–780
20. Nakamura EF, Loureiro AAF, Orgambide ACF (2007) Information fusion for wireless sensor networks: methods, models, and classifications. ACM Comput Surv 39:1–55
21. Neto JMRS, Silva JJC, Cavalcanti TCM, Rodrigues DP, da Rocha Neto JS, Glover IA (2010) Propagation measurements and modeling for monitoring and tracking in animal husbandry applications. In: Proceedings of the instrumentation and measurement technology conference (I2MTC'10). Austin, Texas, USA, pp 1181–1185
22. Niculescu D, Nath B (2001) Ad hoc positioning system (aps). In: Proceedings of the global telecommunications conference GLOBECOM'01), pp 2926–2931
23. Olfati-Saber, R.: Distributed kalman filter with embedded consensus filters. In: Proceedings of the 44th Conference on Decision and Control—European Control Conference (CDC-ECC'05), pp. 8179–8184 (2005)
24. Oliveira HABF, Boukerche A, Nakamura EF, Loureiro AAF (2009) An efficient directed localization recursion protocol for wireless sensor networks. IEEE Transactions Computing 58:677–691
25. Oliveira, H.A.B.F., Nakamura, E.F., Loureiro, A.A.F., Boukerche, A.: Error analysis of localization systems in sensor networks. In: Proceedings of the 13th International Symposium on Geographic, Information Systems (GIS'05), pp. 71–78 (2005)
26. Rosencrantz M, Gordon G, Thrun S (2003) Decentralized sensor fusion with distributed particle filters. In: Proceedings of the Conference on Uncertainty in AI (UAI)
27. Sichitiu, M.L., Ramadurai, V.: Localization of wireless sensor networks with a mobile beacon. In: Proceedings of the International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04), pp. 174–183 (2004)
28. Souza, E.L., Campos, A., Nakamura, E.F.: Tracking targets in quantized areas with wireless sensor networks. In: Proceedings of the 36th Local Computer Networks (LCN'11), pp. 235–238. Bonn, Germany (2011)
29. Souza, E.L., Nakamura, E.F., de Oliveira, H.A.: On the performance of target tracking algorithms using actual localization systems for wireless sensor networks. In: Proceedings of the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '09), pp. 418–423. Tenerife, Canary Islands, Spain (2009)
30. Tatham, B., Kunz, T.: Anchor node placement for localization in wireless sensor networks. In: Proceedings of the 7th Conference on Wireless and Mobile Computing, Networking and, Communications (WiMob'11), pp. 180–187 (2011)
31. Tsai HW, Chu CP, Chen TS (2007) Mobile object tracking in wireless sensor networks. Computer Communications 30:1811–1825
32. Vercauteren T, Guo D, Wang X (2005) Joint multiple target tracking and classification in collaborative sensor networks. IEEE Journal on Selected Areas in Communications 23:714–723
33. Walchli, M., Skoczylas, P., Meer, M., Braun, T.: Distributed event localization and tracking with wireless sensors. In: Proceedings of the 5th International Conference on Wired/Wireless Internet, Communications (WWIC'07), pp. 247–258 (2007)
34. Wang Z, Bulut E, Szymanski BK (2010) Distributed energy-efficient target tracking with binary sensor networks. ACM Transactions on Sensor Networks 6(4):1–32
35. Welch, G., Bishop, G.: An introduction to the kalman filter. In: The 28th International Conference on Computer Graphics and Interactive, Techniques (SIGGRAPH'01) (2006)
36. Yang, H., Sikdar, B.: A protocol for tracking mobile targets using sensor networks. In: Proceedings of the 1st International Workshop on Sensor Network Protocols and Applications (SNPA'03), pp. 71–81 (2003)
37. Zhang W, Cao G (2004) Dctc: Dynamic convoy tree-based collaboration for target tracking in sensor networks. IEEE Transactions on Wireless Communications 3:1689–1701
38. Zhao F, Shin J, Reich J (2002) Information-driven dynamic sensor collaboration for tracking applications. IEEE Signal Processing 19:61–72