Sensors **2009**, *9*, 2760-2779; doi:10.3390/s90402760

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Adapting Mobile Beacon-Assisted Localization in Wireless Sensor Networks

Guodong Teng *, Kougen Zheng and Wei Dong

College of Computer Science, Zhejiang University, Hangzhou 310027, P.R. China; E-Mails: zkg@zju.edu.cn; dongw@zju.edu.cn

* Author to whom correspondence should be addressed. E-Mail: teng@zju.edu.cn

Received: 26 March 2009; in revised form: 14 April 2009 / Accepted: 16 April 2009 / Published: 20 April 2009

Abstract: The ability to automatically locate sensor nodes is essential in many Wireless Sensor Network (WSN) applications. To reduce the number of beacons, many mobile-assisted approaches have been proposed. Current mobile-assisted approaches for localization require special hardware or belong to centralized localization algorithms involving some deterministic approaches due to the fact that they explicitly consider the impreciseness of location estimates. In this paper, we first propose a range-free, distributed and probabilistic Mobile Beacon-assisted Localization (MBL) approach for static WSNs. Then, we propose another approach based on MBL, called Adapting MBL (A-MBL), to increase the efficiency and accuracy of MBL by adapting the size of sample sets and the parameter of the dynamic model during the estimation process. Evaluation results show that the accuracy of MBL and A-MBL outperform both Mobile and Static sensor network Localization (MSL) and Arrival and Departure Overlap (ADO) when both of them use only a single mobile beacon for localization in static WSNs.

Keywords: Wireless Sensor Networks (WSNs); Localization; Mobile Beacon-assisted Localization (MBL); Adapting Mobile Beacon-assisted Localization (A-MBL); Particle filter.

1. Introduction

Wireless Sensor Networks (WSNs) are composed of large numbers of tiny sensor devices with wireless communication capabilities. WSN systems have been developed recently for numerous applications such as military surveillance [1], environmental monitoring [2,3], target tracking [4,5], habitat monitoring [6] and structural monitoring [7], etc. Because many of them require sensor position information, *localization* has been an important problem in WSNs [8] and several localization systems and algorithms have been proposed in the past. The large body of solutions for the sensor node localization problem can be categorized based on whether the localization techniques are *Range-based* or *Range-free*, whether the localization algorithms are *Centralized* or *Distributed*, and whether localization results are *Deterministic* or *Probabilistic*.

In most existing WSNs, sensors are static [9]. The localization of these *static WSNs* relies on several beacons which know their locations scattered throughout the sensor networks and the precision of the localization increases with the number of beacons. The main problem with an increased number of beacons is that they are more expensive than the rest of the sensor nodes, and after these sensor nodes have been localized, the beacons become useless. The leads us to believe that a single mobile beacon which can travel the entire deployment region based on some traverse route can be used to help localize the entire network. Using a single mobile beacon that knows its position is broadly equivalent to using many static beacons each broadcasting once.

In this paper, we propose two mobile beacon-assisted localization approaches, *Mobile Beacon-assisted Localization* (MBL) and Adapting MBL (A-MBL) for static WSNs. Compared to some proposed mobile-assisted approaches [10-12], MBL and A-MBL do not require any specially designed hardware due to the range-free technique employed. Compared to the algorithm requiring the gathering of connectivity data (range or proximity) from the network to a more computationally powerful device [13], MBL and A-MBL focus on distributed algorithms. As the approaches described in [14,12], MBL and A-MBL adopt probabilistic methods which give an area where a sensor might reside, along with the likelihood of such an estimate. Like the Arrival and Departure Overlap (ADO) approach [15], MBL and A-MBL rely on direct Arriver and Leaver information from a single mobile-assisted beacon. Especially, inspired by ideas from [16,17], we adopt an adapting mechanism to improve the efficiency and accuracy of MBL.

This paper offers the following two major contributions:

- We propose a range-free, distributed and probabilistic MBL approach. This approach outperforms both Mobile and Static sensor network Localization (MSL) and ADO when both of them use only a single mobile beacon for localization in static WSNs.
- We propose another approach based on MBL, called A-MBL, to increase the efficiency and accuracy of MBL by adapting the size of sample sets and the parameter of the dynamic model during the estimation process.

The rest of this paper is organized as follows: Section 2 defines mobile beacon-assisted localization problem from Bayesian filter and particle filter perspective. Section 3 presents details of the proposed MBL and A-MBL algorithms. Section 4 shows and discusses our evaluation results. Section 5 gives an overview of related works. Finally, Section 6 concludes our work.

2. Description of the Problem

2.1. Mobile Beacon-assisted Localization Problem

Let us consider a sensor network with *M* static sensor nodes in a 2D plane which do not have *a priori* known locations (called *unknown nodes*) and a single mobile node (called *beacon*), equipped with localization hardware, e.g., GPS, which allows it to know its location at all times. After random deployment of the unknown nodes in a fixed-size area, the beacon traverses the sensor network while broadcasting packets which contain the coordinates of itself and other information. Any unknown node receiving these packets from the beacon (called *observation*) can recognize that it is in the area around the beacon's current location with a certain probability. With each observation in a series of different time, the unknown node's location is bounded in the beacon's transmission area. The accuracy can be improved when the unknown node obtains more observations from the beacon. Location estimates and observations are assumed to be available at discrete times. For dynamic state estimation, the discrete-time approach is widespread and convenient [18].

2.2. Problem Description with Bayesian Filter

If we solve the above-mentioned localization problem with a probabilistic approach, we are interested in estimating the unknown node's *real location R* at the current time-step *t*, given knowledge about the initial location estimate and all observations $o_{1:t} = \{o_1, ..., o_t\}$ up to the current time *t*. This localization problem is an instance of the *Bayesian filtering* problem which requires the estimation of the state of a system that changes over time using a sequence of noisy measurements made on the system [18], where we are interested in constructing the posterior density $p(l_t \mid o_{1:t})$ of the current location estimate l_t conditioned on all observations $o_{1:t}$ from the beacon. To define this localization problem from a Bayesian filter perspective, we assume that we have an *initial distribution*, a *dynamic model* and an *observation model*:

$$p(l_0 \mid o_0), \tag{1}$$

$$l_t = f(l_{t-1}) \quad \text{for} \quad t \geq 1, \tag{2}$$

$$o_t = g(l_t) \quad \text{for} \quad t \geq 1. \tag{3}$$

By Equation (2) and Equation (3) we mean that l_t and o_t are assumed to be generated by functions $f(\cdot)$ and $g(\cdot)$, respectively. The precise form of the functions implies via a change of variables the transition probability density $p(l_t \mid l_{0:t-1}, o_{1:t-1})$ and the observation probability density $p(o_t \mid l_{0:t}, o_{1:t-1})$. We denote by $l_{0:t} = \{l_0, ..., l_t\}$, the location estimate up to time *t*. Note that we could assume Markov transitions and conditional independence to simplify the model due to the constraints in computing and memory power of the sensor node, i.e. current location estimate l_t is only dependent on the previous location estimate l_{t-1} and the current observation o_t is only dependent on the current location estimate l_t, then the dynamic model $p(l_t \mid l_{0:t-1}, o_{1:t-1}) = p(l_t \mid l_{t-1})$ and the observation model $p(o_t \mid l_{0:t}, o_{1:t-1}) = p(o_t \mid l_t)$. The posterior density $p(l_t \mid o_{1:t}) = p(l_t \mid o_t)$ may be obtained after *initialization*, recursively, in two stages: a *prediction* stage and an *update* stage.

Initialization: It is assumed that the initial $p(l_0 \mid o_0) \equiv p(l_0)$ of the location estimate, which is also known as the prior density, is available (o_0 being the set of no observations).

Prediction: Suppose that the posterior density $p(l_{t-1} \mid o_{t-1})$ at time *t*-1 is available. The prediction stage involves using a dynamic model Equation (2) to obtain the prior density $p(l_t \mid o_{t-1})$ of the location estimate at time *t* via the Chapman-Kolmogorov equation:

$$p(l_t \mid o_{t-1}) = \int p(l_t \mid l_{t-1}) p(l_{t-1} \mid o_{t-1})\, dl_{t-1} \ . \tag{4}$$

Update: At time step *t*, an observation o_t becomes available, and this may be used to update the prior density via Bayes' rule:

$$p(l_t \mid o_t) = \frac{p(o_t \mid l_t) p(l_t \mid o_{t-1})}{p(o_t \mid o_{t-1})} \tag{5}$$

where the normalizing constant

$$p(o_t \mid o_{t-1}) = \int p(o_t \mid l_t) p(l_t \mid o_{t-1})\, dl_t \tag{6}$$

depends on the likelihood function $p(o_t \mid l_t)$ defined by the observation model Equation (3). In the update stage Equation (5), the observation o_t is used to modify the prior density to obtain the required posterior density of the current location estimate.

2.3. Problem Description with Particle Filter

To address the complexity of the integration step in Bayesian filter, many optimal or suboptimal approaches are proposed. The recurrence relations Equation (4) and Equation (5) form the basic for the optimal or suboptimal Bayesian solution. Solutions do exist in a restrictive set of cases [18], including the *Kalman filter* (optimal Bayesian filter) and *particle filter* (suboptimal Bayesian filter) which approximates the optimal Bayesian solution, etc.

The Kalman filter assumes that the posterior density at every time step is Gaussian and, hence, parameterized by a mean and covariance, provided that certain assumptions hold: $f(\cdot)$ and $g(\cdot)$ in the dynamic model and the observation model are known and also are a linear function. In many situation of interest, such as mobile beacon-assisted localization problem in our scenarios, the assumptions made above do not hold. The Kalman filter cannot, therefore, be used as described – approximations are necessary.

The particle filter solutions offer a number of significant advantages compared with other techniques currently available, including the Kalman filter. These advantages arise principally from the generality of the approach, which allows inference of full posterior distributions in general state-space models, which may be both nonlinear and non-Gaussian.

Thus, in this paper, we use a particle filter (also called *sequential Monte Carlo method*) to perform a Bayesian filter on a sample representation. The key idea is to represent the required posterior density by a set of random samples with associated weights and to compute estimates based on these samples and weights [18]. As the number of samples becomes very large, this characterization of particle filter becomes an equivalent representation to the usual functional description of the posterior density, and the particle filter approaches the suboptimal Bayesian estimate.

In order to develop the details of the algorithm, let $\{<l_t^i, w_t^i>, i=1,...,N\}$ denotes a *random measure* that characterizes the posterior density $p(l_t \mid o_t)$, where $L_t = \{l_t^i, i=1,...,N\}$, is a set of support *samples* (or called *particles*) with associated *weights* $W_t = \{w_t^i, i=1,...,N\}$. The weights are normalized such that $\sum_{i=1}^{N} w_t^i = 1$. Then, the posterior density at t can be approximated as:

$$p(l_t \mid o_t) \approx \sum_{i=1}^{N} w_t^i \delta(l_t - l_t^i) \tag{7}$$

where $\delta(\cdot)$ is Dirac delta function

$$\delta(l_t - l_t^i) = \begin{cases} 1 \;\; if \; l_t = l_t^i \\ 0 \;\; otherwise \end{cases} \tag{8}$$

samples l^i are easily generated from a *proposal density* (or called *importance density*) $q(\cdot)$:

$$l^i \sim q(l) \tag{9}$$

where the symbol ~ denotes sample generated sign, i.e., the samples on the left side are generated from the probability density on the right side. Weights are defined by:

$$w_t^i \propto \frac{p(l_t^i \mid o_t)}{q(l_t^i \mid o_t)} \tag{10}$$

where the symbol $\propto$ is used to denote proportionality up to a normalization constant.

The proposal density is chosen to factorize such that:

$$q(l_t \mid o_t) = q(l_t \mid l_{t-1}, o_t) q(l_{t-1} \mid o_{t-1}). \tag{11}$$

To derive the weight update equation, $p(l_t \mid o_t)$ from Equation (5) is expressed via Bayes' rule:

$$\begin{aligned} p(l_t \mid o_t) &= \frac{p(o_t \mid l_t) p(l_t \mid o_{t-1})}{p(o_t \mid o_{t-1})} \\ &\propto \frac{p(o_t \mid l_t) p(l_t \mid l_{t-1}) p(l_{t-1} \mid o_{t-1})}{p(o_t \mid o_{t-1})} \\ &\propto p(o_t \mid l_t) p(l_t \mid l_{t-1}) p(l_{t-1} \mid o_{t-1}). \end{aligned} \tag{12}$$

The weight update equation can then be shown to be (the proposal density is chosen to be dynamic model Equation (2), i.e. $q(l_t^i \mid l_{t-1}^i, o_t) = p(l_t^i \mid l_{t-1}^i)$):

$$\begin{aligned} w_t^i &\propto \frac{p(l_t^i \mid o_t)}{q(l_t^i \mid o_t)} \\ &\propto \frac{p(o_t \mid l_t^i) p(l_t \mid l_{t-1}^i) p(l_{t-1}^i \mid o_{t-1})}{q(l_t^i \mid l_{t-1}^i, o_t) q(l_{t-1}^i \mid o_{t-1})} \\ &\propto \frac{p(o_t \mid l_t^i) p(l_t^i \mid l_{t-1}^i)}{q(l_t^i \mid l_{t-1}^i, o_t)} w_{t-1}^i \\ &\propto p(o_t \mid l_t^i) w_{t-1}^i. \end{aligned} \tag{13}$$

The generic particle filter proceeds for localization are as follows:

Initialization: N samples and weights are chosen from the initial distribution Equation (14) and the initial observation Equation (15), respectively.

$$l_0^i \sim p(l_0 \mid o_0), \tag{14}$$

$$w_0^i = p(o_0 \mid l_0^i). \tag{15}$$

Prediction: It starts from the set of samples $\{l_{t-1}^i, i = 1,...,N\}$ computed in the previous iteration, and applies the dynamic model to each sample l_{t-1}^i by sampling from the density $p(l_t \mid l_{t-1})$, i.e. for each particle l_{t-1}^i draw one sample l_t^i from $p(l_t \mid l_{t-1})$ by:

$$l_t^i \sim p(l_t \mid l_{t-1}). \tag{16}$$

Update: It takes into account the observation o_t. Each weight w_t^i of the sample in $\{l_t^i, i = 1,...,N\}$ is obtained by the importance weight Equation (13), i.e. the likelihood of $\{l_t^i, i = 1,...,N\}$ given o_t.

3. Mobile Beacon-Assisted Localization

3.1. MBL

Assumption: We assume that all unknown nodes are randomly deployed in an area of size *S* and each sensor (unknown node or beacon) has the same ideal radio range *r*. The algorithm does not assume very tightly synchronized clocks. The beacon is capable of moving a distance v_b in a time step in any direction where $0 \le v_b \le v_{max}$. The beacon knows v_{max}, but it does not know the value of v_b or the direction of movement in any time step. At time *t*, every unknown node within the radio range of the beacon will hear a location announcement from that beacon. In a realistic deployment, it would be necessary to deal with network collisions and account for missed messages [19].

Initialization: In this stage, all unknown nodes have no information about their locations. The initial set of samples Equation (17) for each unknown node is chosen randomly from the whole deployed area and represented by a set of uniformly distributed samples with equal weights Equation (18). The weight equal to one represents the importance of corresponding sample, which infers one of the location estimates of the unknown node. The beacon's current initial position Equation (19) is also chosen randomly from the whole deployment area. The beacon's previous initial position (for further use) is chosen randomly from out of the deployment area and out of the radio range of any unknown node Equation (20):

$$L_0 = \{l_0^i \mid l_0^i \sim p(l_0) = \frac{1}{S}, 1 \le i \le N\}, \tag{17}$$

$$w_0^i = 1, 1 \le i \le N, \tag{18}$$

$$beacon_c \sim \frac{1}{S}, \tag{19}$$

$$beacon_p = (beacon_p \sim \frac{1}{\overline{S}}) \wedge (d(beacon_p, R) > r). \tag{20}$$

where L_0 denotes the initial set of each sample, w_0^i denotes the initial weight of each sample, $beacon_c$ denotes the beacon's current location, $d(beacon_p,R)$ denotes the distance between locations $beacon_p$ and *R*, *R* denotes the unknown node's real location, $\overline{S}$ denotes the area out of the deployment area, and $beacon_p$ represents the beacon's previous location.

Prediction: In this stage, we adopt a dynamic model that the unknown node is capable of moving a distance v_{node} in a time step in any direction where $0 \le v_{node} \le \alpha$. The unknown node knows v_{node}, but it does not know the value of v_{node} or the direction of movement in any time step. Then, the unknown node generates new samples as follows:

$$P_t=\{l_t^i \mid l_t^i \text{ is selected from } p(l_t^i \mid l_{t-1}^i), \text{ where } l_{t-1}^i \in L_{t-1} \text{ for all } 1 \leqslant i \leqslant N\}, \tag{21}$$

where P_t represents approximation of prior density at time t after prediction stage, L_{t-1} represents approximation of posterior density at time t-1 , and the transition equation for each sample described as follows:

$$p(l_t \mid l_{t-1}) = \begin{cases} \dfrac{1}{\pi\alpha^2} & if\ filter(R) = TRUE \\ 0 & if\ filter(R) = FALSE. \end{cases} \tag{22}$$

The parameter α is needed for unknown nodes to provide enough variability in choosing new samples, i.e., the parameter α is used to limit a sample *impoverishment* phenomenon. Every time step, the beacon randomly moves a distance v_b in any direction from the previous location. A new sample is generated from each current sample by randomly choosing a point within a circle centered at the current location of the sample and the radius α when the *filter(R)* equal to $TRUE$, where *filter(R)* represents the filter condition of real location R of the unknown node. The details of condition *filter(R)* will be described in next stage (update stage). In general, the smaller the parameter a, the higher the accuracy of localization is obtained, but the longer time the stable phase of localization process is achieved. The appropriate value α is a tradeoff between the high precision of localization and the short period of localization time. Here, the appropriate value α could be determined empirically, such as α=0.1r.

Update: In this stage, the unknown node filters the impossible samples based on new observations. The unknown node updates samples as follows:

$$U_t=\{l_t^i \mid l_t^i \text{ where } l_t^i \in P_t \text{ and } w(l_t^i)=1\} \tag{23}$$

where U_t represents approximation of posterior density at time t after the update stage, and the weight $w(l_t^i)$ will be obtained by $p(o_t \mid l_t^i)$.

In order to state the description of observation $p(o_t \mid l_t^i)$, we define four states for every unknown node during the localization process.

Outsider: The unknown node is out of the radio range of the beacon, i.e.,

$$d(beacon_c, R) > r. \tag{24}$$

Insider: The unknown node is within the radio range of the beacon, i.e.,

$$d(beacon_c, R) \le r. \tag{25}$$

Arriver: The unknown node receives the current location announcement of the beacon, but did not receive the location announcement from the beacon's previous location, i.e.,

$$d(beacon_c, R) \le r \wedge d(beacon_p, R) > r. \tag{26}$$

Leaver: The unknown node received the preceding location announcement from the beacon, but does not receive the location announcement from the beacon's current location, i.e.,

$$d(beacon_c, R) > r \wedge d(beacon_p, R) \le r. \tag{27}$$

In this paper, we only rely on observations from the beacon. This has two advantages. First, the number of unknown nodes will not affect the accuracy of localization. Second, the computation and communication costs drop drastically, since nodes are no longer involved in the localization of other nodes [9]. There are two possible ways to gather observations from the beacon:

1. Once the unknown node is in Insider state, it gathers this observation, i.e. the filter condition of the real location R for any unknown node is:

$$filter(R) = \begin{cases} TRUE & if\ d(beacon_c, R) \le r \\ FALSE & otherwise. \end{cases} \tag{28}$$

2. When the unknown node is in Arriver or Leaver state, it gathers this observation, i.e. the filter condition of the real location R for any unknown node is:

$$filter(R) = \begin{cases} TRUE & if\ AL = TRUE \\ FALSE & otherwise, \end{cases} \tag{29}$$

where AL represents Arriver or Leaver state:

$$\begin{aligned} AL = & (d(beacon_c, R) \le r \wedge d(beacon_p, R) > r) \\ & \vee (d(beacon_c, R) > r \wedge d(beacon_p, R) \le r). \end{aligned} \tag{30}$$

From an implementation perspective, in the first way, the beacon just transmits information about its own current location. Once the unknown node hears a location announcement from that beacon, an observation (also called *constraint*) is built to update.

In the second way, the beacon transmits both its current location and its location at the previous time step in each announcement. The unknown node needs to save state (Insider or Outsider) in previous time step with a tag:

$$tag = \begin{cases} TRUE & if\ state = Insider \\ FALSE & if\ state = Outsider. \end{cases} \tag{31}$$

This procedure at an unknown node is then as described by Algorithm 1. As can be seen, the tag which represented by variable *StateTag* is initialized to FALSE when the localization started (step 1-3). Then, we get the filter condition of location R for an unknown node (step 4-10). Finally, we get *StateTag* with step12-15.

Algorithm 1. State at an unknown node.

```
1:  If t=0 then
2:    StateTag=FALSE
3:  end if
4:  filter(R)=FALSE
5:  if Insider ∧ (StateTag=FALSE) then
6:    filter(R)=TRUE
7:  end if
8:  if Outsider ∧ (StateTag=TRUE) then
9:    filter(R)=TRUE
10: end if
11: if Insider then
12:   StateTag=TRUE
13: else
14:   StateTag=FALSE
15: end if
```

The first way was adopted by Monte Carlo Localization (MCL) [19] and MSL [9]. As the approach ADO, we adopt second way in the update stage, however. We will evaluate the accuracy of them in Section 4.

Then, the weight of sample is determined by the filter condition:

$$w_t = p(o_t \mid l_t) = \begin{cases} 1 \ \ if \ \ filter(R) = TRUE \\ 0 \ if \ \ filter(R) = FALSE. \end{cases} \tag{32}$$

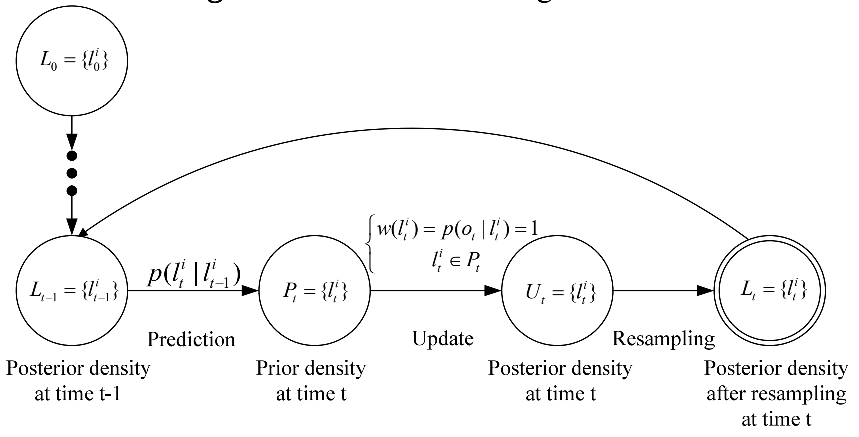
Resampling: A common problem with particle filter is the *degeneracy* phenomenon. The degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the approximation to $p(l_t \mid o_{1:t})$ is almost zero. A suitable measure of degeneracy of the algorithm is the effective sample size N_{eff} defined as:

$$N_{eff} = 1 / \sum_{i=1}^{N} (w_t^i)^2, \tag{33}$$

notice that $N_{eff} \le N$, and small N_{eff} indicates severe degeneracy. The method by which the effects of degeneracy can be reduced is to use *resampling* whenever a significant degeneracy is observed (i.e., when N_{eff} falls below some threshold N_T). The basic idea of resampling is to eliminate particles that have small weights and to concentrate on particle with large weights. We adopt a Systematic resampling algorithm [20] in this paper since it is simple to implement, takes $O(N_s)$ time, and minimizes the Monte Carlo variation.

Finally, in order to give more clear description of MBL, we describe the main stages as a state machine diagram with labeled transitions, see Figure 1.

Figure 1. State machine diagram of MBL



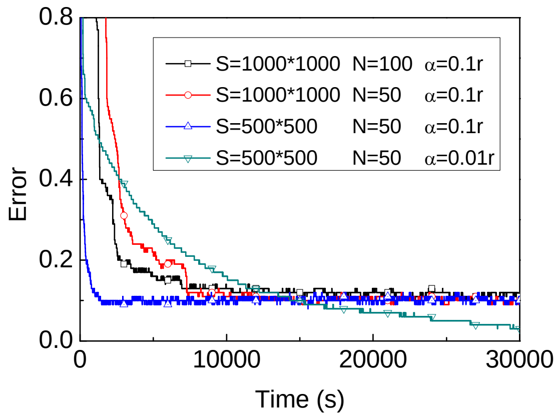
$L_0 = \{l_0^i\}$
$L_{t-1} = \{l_{t-1}^i\}$
$p(l_t^i \mid l_{t-1}^i)$
$P_t = \{l_t^i\}$
$w(l_t^i) = p(o_t \mid l_t^i) = 1$
$l_t^i \in P_t$
$U_t = \{l_t^i\}$
$L_t = \{l_t^i\}$
Prediction
Update
Resampling
Posterior density at time t-1
Prior density at time t
Posterior density at time t
Posterior density after resampling at time t

3.2. A-MBL

3.2.1. Number of Samples

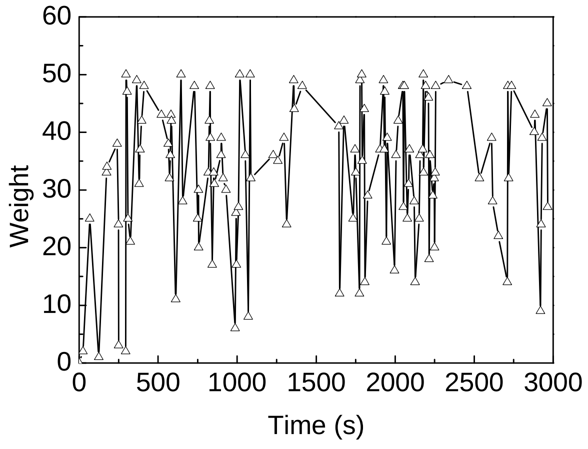
For reference, we first evaluate the trend of location error defined by Equation (35) of four different MBL exemplars (evaluation results shown in Figure 2). Figure 2 shows that, under the same conditions (S=1,000×1,000, α=0.1r), keeping more samples improves efficiency in the beginning of localization. The time complexity of the update stage and the memory requirement to keep samples both are linear in the number of samples needed for the estimation, however. Therefore, the attempt will be made to make more effective use of the available samples, thereby allowing sample sets of reasonable size. As a result, with a fixed number of samples one has to choose a tradeoff so as to allow address efficiency, time and space complexity problem. Before we introduce our method for adaptive particle filter, let us first discuss two existing technique to changing the number of samples during the filter process.

Figure 2. Different exemplars of MBL.



Likelihood-based adaptation [21]: The intuition behind this approach is as follows. If the sample set is well in tune with the sensor reading, each individual importance weight is large and the sample set remains small. If we adopt this approach, we will choose smaller sample set than previous when $\sum_{i=1}^{N} w_t^i \geq \textit{threshold}$, i.e. the sum of weight $\sum_{i=1}^{N} w_t^i$ has always been greater than a certain value (threshold) since the beginning of a certain time during the localization, the number of sample will be reduced. As shown in Figure 3, the sum of weights for a single unknown node changes over time in MBL. It is clear that more severe fluctuations of weight changing over time in MBL, however. Thus, the conditions of likelihood-based adaptation for MBL shall be deemed invalid.

Figure 3. Sum of weight for MBL.



KLD-Sampling adaptation [17]: The key idea of the KLD-sampling approach is to bound the error introduced by the sample-based belief representation. At each iteration, this approach generates samples until their number is large enough to guarantee that the KL-distance between the maximum likelihood estimate and the underlying posterior does not exceed a pre-specified bound. However, the additional cost of KLD-sampling is higher, especially for sensor node, since each lookup takes time logarithmic in the size of the state space. Thus, it is inefficient for us to adopt this approach.

3.2.2. Parameter α

As shown in Figure 2, under the same conditions (S=500×500, N=50), a greater value of α improves the efficiency in the beginning of localization and improves accuracy at the other extreme. The question is how to determine the value of parameter α to achieve both high precision of localization and a short localization time period. We find that the localization error ultimately will remain stable regardless of the value of α. Thus, the key idea is to reduce the value of α when the stable phase of localization process is achieved, i.e., the dynamic model maintains greater value of α in the beginning of localization to achieve shorter localization time and updates to smaller value of α to obtain higher precision.

To judge the localization to reach the stable phase, a simple and intuitive approach is to adopt *coefficient of variation* c_v of recent location estimating results, which is defined as:

$$c_v = \frac{\sigma}{\mu}, \tag{34}$$

where σ and μ is the *standard deviation* and the *mean* of recent location estimating results, respectively. When $c_v<\varepsilon$, we update α by $\alpha=\eta\alpha$, where ε is a pre-specified threshold and η is adjustment factor for α. However, when N is relatively small, the coefficient of variation is not good to judge the stable phase for a single unknown node, and maintaining recent location estimating results requires additional memory for sensor node. Thus, it is also inefficient for us to adopt this approach.

3.2.3. Our proposed approach and implementation

We adopt two predefined adjustment tables in our approach, one for the number of samples N, the other for the parameter α (examples shown in Table 1 and 2). The two tables which include the following fields: TIME, N and TIME, ALPHA. Once some record in the table is matched, the number of samples and the value of α in the unknown node will be adjusted according to the corresponding time. The table of N and α used in this paper was determined empirically. In general, the size of these two tables is very small. Both the implementation and computation overhead of this approach are also small. Thus, we adopt this approach to adapting previous proposed algorithm MBL.

The implementation details are described in Algorithm 2, where L_N is the adjustment list of the number of sample and L_N*.length* is the length of the list, so does the L_α to the parameter α. The beacon transmits the information about these tables when contacting the unknown node for the first time (step 2-6), and the unknown node keeps the value in the list L_N and L_α. Then, the unknown node adjusts the number of samples N (step 7-13). In step 8, the N which is on the left of assignment denotes new number of samples. The x_t^i denotes the samples with new N in step 11 and the w_t^i denotes the weights with new N in step 12. The parameter α (step 14-17) based on above information from the beacon. Finally, the complete A-MBL for every unknown node is shown in Algorithm 3.

Algorithm 2. adaptive step in unknown node.

```
1: procedure ADAPTING
2:  if $(d(beacon,l) \le r) \wedge (firstContracted = FALSE)$ then
3:      $L_N[k_N] \leftarrow InitValueFromBeacon$
4:      $L_\alpha[k_\alpha] \leftarrow InitValueFromBeacon$
5:      $firstContract \leftarrow TRUE$
6:  end if
7:  if $(t = L_N[k_N].t) \wedge (k_N < L_N.length)$ then
8:      $N_p \leftarrow N$
9:      $N \leftarrow L_N[k_N].N$
10:     $k_N \leftarrow k_N + 1$
11:     $x_t^i \leftarrow x_t^j (i \leftarrow 1, N; j \leftarrow 1, N_p)$
12:     $w_t^i \leftarrow w_t^j (i \leftarrow 1, N; j \leftarrow 1, N_p)$
13:  end if
14:  if $(t = L_\alpha[k_\alpha].t) \wedge (k_\alpha < L_\alpha.length)$ then
15:      $\alpha \leftarrow L_\alpha[k_\alpha].\alpha$
16:      $k_\alpha \leftarrow k_\alpha + 1$
17:  end if
18: end procedure
```

Algorithm 3. A-MBL.

```
1: $k_\alpha \leftarrow 0$
2: $k_N \leftarrow 0$
3: $firstContracted \leftarrow FALSE$
4: for $i \leftarrow 1, N$ do
5:   INITIALIZATION
6: end for
7: for $t \leftarrow 1, T$ do
8:   for $i \leftarrow 1, N$ do
9:      $l_t^i \sim p(l_t^i \mid l_{t-1}^i)$
10:     $w_t^i \leftarrow p(o_t \mid l_t^i)$
11:  end for
12:  $w_t^i \leftarrow NORMALIZE(w_t^i)$
13:  $N_{eff} \leftarrow NEFF(w_t^i)$
14:  if $(N_{eff} < N_T \wedge filter(l) = TRUE)$ then
15:     $\{< l_t^j, w_t^j >\} \leftarrow RESAMPLING(\{< l_t^i, w_t^i >\})$
16:  end if
17:  ADAPTING
18: end for
```

4. Evaluation

4.1. Assumption

The key metric [9] for evaluating a localization algorithm is the accuracy of the location estimates or *localization error*. This is computed as follows:

$$Error = \frac{1}{M}\sum_{i=1}^{M} \| e_i - R_i \|, \tag{35}$$

where M is the number of unknown nodes, R_i denotes the real location of the i-th unknown node, e_i denotes the location estimate of the i-th unknown sensor where i=1,…,M and $\| e_i - R_i \|$ denotes the distance between locations e_i and R_i. The errors shown in the simulation results are in terms of the radio range, i.e., the errors shown are computed by dividing the error in Equation (35) by the radio range of sensors. Most parameter settings for our simulations are those used in [19, 9]. Our results were obtained using sensors randomly distributed in a 500 units × 500 units square field, i.e. S=500×500. In our experiments, we set ideal radio range r=100, the number of unknown nodes M=100, the number of samples for an unknown node N=50, the parameter α=0.1r in the prediction stage, the maximum speed of beacon v_{max}=1.0r and two predefined adjustment tables (shown in Table 1 and 2) for adapting unless otherwise specified. Other simulation parameters of the unknown node are based on the MicaZ sensor node. The beacon's movement is implemented using random waypoint mobility model.

Table1 Predefined adjustment tables for N.

ID	Time	N
1	0	50
2	2,000	20

Table 2 Predefined adjustment tables for α.

ID	Time	Alpha
1	0	0.1
2	1,500	0.01

In this section, we first evaluate MBL algorithm under various parameters configuration, such as the maximum speed of the beacon, the number of samples for an unknown node, and the impact of parameters α in the prediction stage. Then, under the same conditions (e.g. only use a single mobile beacon for localization), we compare the efficiency and accuracy of MBL, A-MBL, MSL, and ADO in different parameters configuration. In addition, we will consider a noisy environment with random noise added to measurements.

4.2. Parameters of MBL

Maximum speed of the beacon: Figure 4 shows the convergence of MBL algorithm under four different v_{max} scenarios. In general, the faster the speed of the beacon, the quicker the stable phase is reached. Because the faster the speed of the beacon, the more the number of unknown nodes which the beacon could contact with. When v_{max} is greater than or equal to $0.6r$, the convergence of MBL is particularly fast, and the localization process can be divided into the initialization phase and the stable phase. In the initialization phase, the estimate error decreases dramatically as new observations (the localization is improved by both the current observation and previous observations) are incorporated. In the stable phase, the impact of filter and the beacon's mobility reach some balance, and the estimate error fluctuates around a minimum value. When v_{max} is greater than or equal to $0.8r$, the curves of convergence (e.g. v_{max}=$1.0r$ and v_{max}=$1.2r$ shown in Figure 4) are very close to each other. Based on above comparison, we set v_{max}=$1.0r$ as the default parameters configuration. When the time is greater than 1000 under this default parameters configuration, the error fluctuates slightly about a constant value (nearly 0.1).

Number of samples for MBL: Maintaining more samples for the MCL algorithm can improve accuracy, but requires additional memory [19]. Based on this comment, we should select an appropriate number of samples which does not affect the accuracy of localization and also does not waste memory either. Figure 5 shows the impact of sample size on location accuracy. The estimate error drops rapidly at the beginning, since a small number of samples cannot adequately reflect the probability distribution. The estimate error is fairly stable after sample size 50 and the accuracy improves only minimally by increasing the number of samples to 100. Hence, MBL is efficient in both memory and computation when the number of sample is close to 50. So, we choose 50 as default sample quantity to save memory and achieve good accuracy.

Parameter α for MBL: Figure 6 shows the impact of parameter α on location accuracy. If $\alpha = 0$, i.e., all samples of the unknown node always keep static in the prediction stage. As a result, it cannot provide enough variability in choosing new samples. Hence, to improve the accuracy, the algorithms should increase the number of samples for each unknown node. In Figure 6, when α=0, the number of samples is needed to reach about 5,000 in order to achieve the similar precision as α=0.1r (but this number of samples is just 50). The parameter α significantly improves the accuracy and reduces the number of samples. Based on a number of experimental results, we adopt α=0.1r and the number of samples N=50 for an unknown node as default values.

4.3. Comparison of Different Algorithms

In order to compare different algorithms under the same conditions, MSL in our evaluation will only use a single mobile beacon for localization.

Efficiency: The graph in Figure 7 shows the efficiency comparison for MBL (α=0.01r), A-MBL under same conditions except the parameter α. In order to obtain close precision, MBL spends more time with a fixed α than A-MBL with adapting α obtained from predefined adjustment tables.

Accuracy: Figure 8 shows the comparison of accuracy for MBL, A-MBL, MSL and ADO under same maximum speed of the beacon (v_{max}). Figure 8 illustrates that the curve of ADO drops to a certain value, then to the horizon, because the Arriver and Leaver information only be used once by ADO when the beacon passes by traverse route. Even though the beacon in ADO pass by the radio ranges of the unknown node on many occasions, the accuracy of unknown node will not be improved. Thus, as the time goes on, the average accuracy of all unknown nodes does not decrease. As shown in Figure 8, MBL and MSL, these two curvature of the curves are very similar to each other. As the time goes on, these two curves both reach to the stable phase. MBL shows nearly 50% better performance and nearly equal time of localization when compared to the MSL. Before the parameter α in A-MBL adjusted, the accuracy of MBL and A-MBL are very close. Once the parameter α in A-MBL has been adjusted, the accuracy of A-MBL is further improved. The accuracy of MBL still keeps stable, however. Thus, it can be seen from the Figure 8 that A-MBL outperforms MBL, MSL and ADO.

Figure 4. Location convergence.

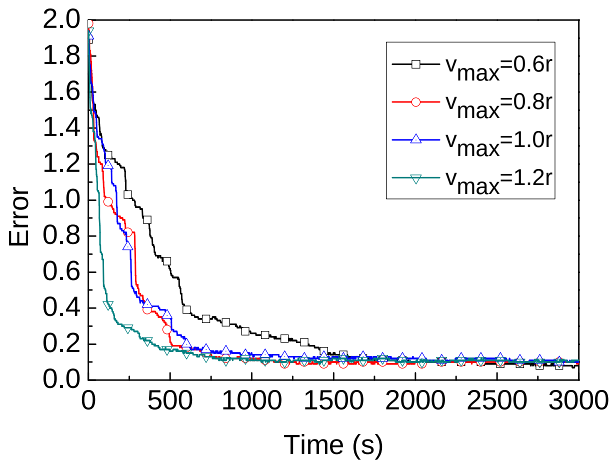


Figure 5. Impact of sample size.

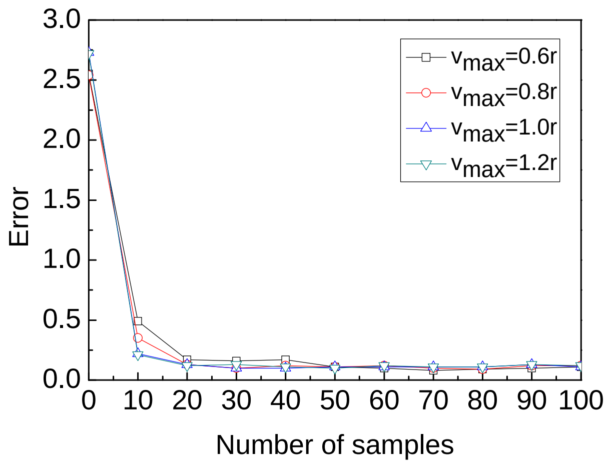


Figure 6. Impact of parameter α.

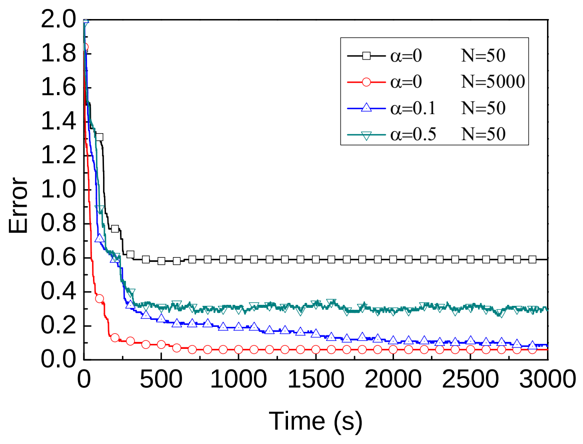


Figure 7. Comparison of efficiency.

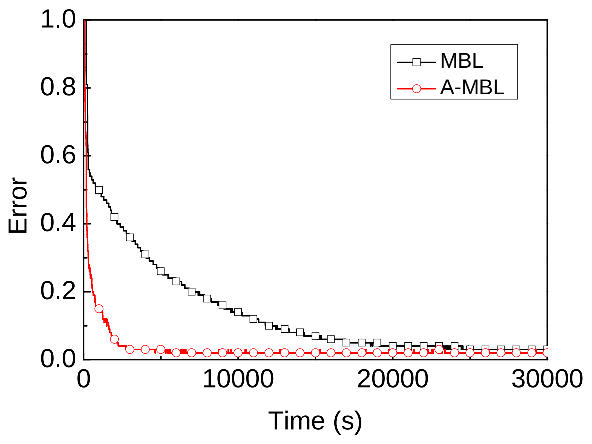


Figure 8. Comparison of accuracy.

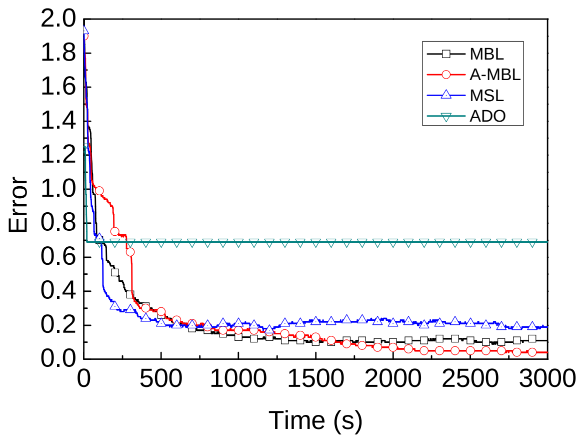


Figure 9. Speed of beacon.

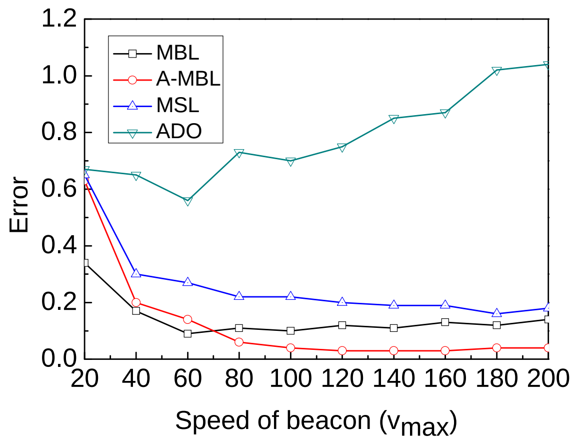


Figure 10. Number of unknown nodes.

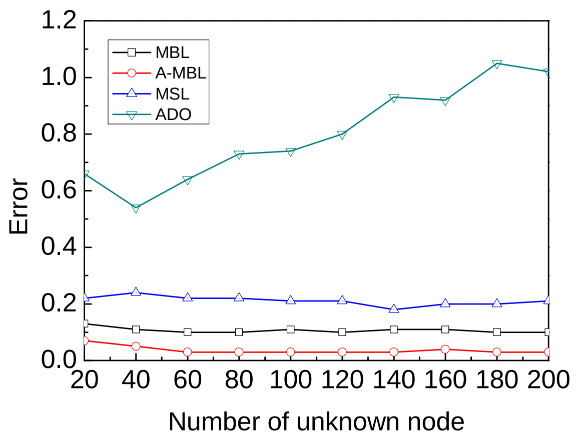
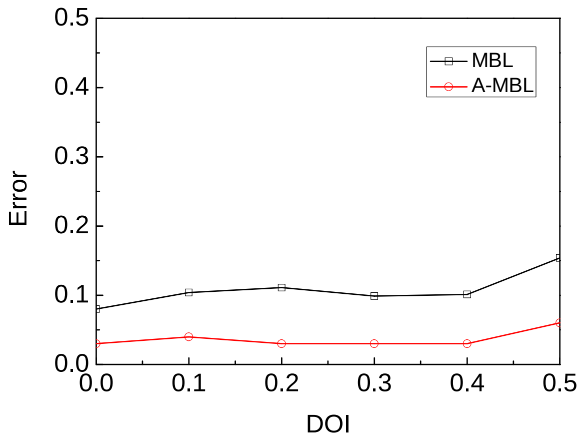


Figure 11. Impact of Irregularity.



Speed of the beacon: Figure 9 shows how the localization error varies with the changing speed of the beacon. As a result, the faster speed of the beacon, the worse accuracy of ADO is achieved. Because the faster speed of the beacon which just travels the whole deployment area of a sensor network only once, the more the numbers of unknown nodes do not be localized. With further increase in the speed of beacon, the accuracy of MBL is close to MSL, for the area of Arriver and Leaver will become smaller. Finally, when $v_b \geq 2r$, the area of Arriver and Leaver is equal to zero, and then MBL will degrade to MSL. The emergence of the significant error when the beacon (v_{max}=20) at low speed, just because the time (*t*=3,000) is too short for convergence of MBL, A-MBL, and MSL. As a result of Figure 9, A-MBL always performs better than both MBL and MSL in some range of speed.

Number of the unknown nodes: In this experiment we vary the number of unknown nodes from 20 to 200. We set the time of beacon movement as 3,000 to achieve the stable phase. As shown in Figure 10, the number of unknown nodes will not affect the accuracy of A-MBL, MBL and MSL, just because each of them only use the beacon for localization in our evaluations (not from neighborhood information).

4.4. Irregularity

All of our previous experiments assumed an ideal scenario where location sensory data are not influenced by irregular radio range and any receiver within the radio range of sender will hear the packets from that sender. However, on the one hand, variability in actual radio transmission patterns can have a substantial impact on localization accuracy depending on the localization technique [19]. On the other hand, the packet reception depends not only on the sender, but also on the receiver.

Figure 11 shows the impact of degree of irregularity on estimate error. The MBL and A-MBL are not substantially affected. We use degree of irregularity (DOI) to denote the maximum radio range variation in the direction of radio propagation. For example, if DOI = 0.1, then the actual radio range in each direction is randomly chosen from [0.9r, 1.1r].

If the unknown node (receiver) within radio range of the beacon (sender) will not hear a location announcement (such as network collisions or missed packages) from that beacon at some time, the beacon must keep moving for longer time to send location information which the unknown node could receive them, i.e. the time to achieve final stable phase of accuracy as ideal state will be extended. The accuracy of MBL and A-MBL will not be affected in such scenarios.

5. Related Works

In this section, we provide a brief survey focusing on mobile-assisted localization approaches suitable for WSNs. MAL [11] proposed by Priyantha *et al.* involves a mobile-assisted localization method which employs a mobile user to assist in measuring distances between node pairs until these distance constraints form a “globally rigid” structure that guarantees a unique localization. In [10], Kim *et al.* propose a novel range-based localization scheme which involves a movement strategy with a low computational complexity of mobile beacon, called mobile beacon-assisted localization (MBAL).

Different from above range-based deterministic mobile beacon-assisted approaches, many range-base probabilistic approaches have been proposed. Sichitiu *et al.* [12] propose a radio frequency (RF) based (i.e. the received signal strength indicator (RSSI) is used for ranging) localization method using Bayesian inference for processing information from one mobile beacon. Caballero *et al.* [22] present range-based (which process the RSSI value in each node in order to localize the nodes of a static wireless network) methods for the 3D localization of an outdoor WSN by using a single flying beacon-node on-board an autonomous helicopter. The technique is based on particle filtering and allows a distributed computation of the position of the nodes. Marinakis *et al.* [23] present, based on Markov

Chain Monte Carlo (MCMC) methodology, a hybrid Inference for Sensor Network Localization using a Mobile Robot, However, this method requires a special hardware platform for the above inference technique: a mobile robot is observed by one of the component stationary cameras in a sensor network. Ihler *et al.* [24] present and demonstrate range-based (scenarios in which each sensor is equipped with a wireless and/or acoustic transceiver and distance is estimated by received signal strength or time delay of arrival between sensor locations) the utility of nonparametric belief propagation (NBP), a generalization of particle filtering, for both estimating sensor locations and representing location uncertainties. Peng *et al.* [14] propose a range-based (which measure the RSS at different distances between a transmitter and a receiver pair) probabilistic, constraint-based approach robust to range measurement inaccuracies.

All above range-based approaches are constrained by the expensive cost and high energy consumptions of the ranging hardware devices. Furthermore, in many practical situations, the measurements are far from accurate (and even sometimes unobtainable) due to highly dynamic environments [25].

Due to the hardware limitations and energy constraints of sensor nodes, range-free localization approaches are cost-effective alternatives to range-based approaches. Walking GPS [26] is a range-free localization, in which the deployer (either person or vehicle) carries a GPS device that periodically broadcasts its location. Each node computes its location estimate according to either the broadcasting positions of the moving beacon or the positions of its neighbors. ADO provides a distributed method to localization of sensor nodes using a single moving beacon where sensor nodes compute their position estimate based on the range-free technique. The method uses the arrival and departure information of a walking beacon.

Our proposed method differs significantly from previous range-base or range-free mobile-assisted localization works because we adopt range-free techniques and solve the problem from particle filter perspective.

Our work is similar to that of Hang *et al.* [13], which discusses the Monte Carlo sampling steps in the context of the localization using a single beacon for various types of observations such as ranging, Angle of Arrival (AoA), connectivity and combinations of those. This method works more like an online algorithm, in which all computation is done at the beacon.

Coates *et al.* [27] present two distributed particle filtering algorithms for Sensor Networks. Different from our work which locate static unknown nodes, this approach is used to track posterior distributions in Markovian state-space models using sensor networks.

The range-free algorithm MCL proposed by Hu *et al.* only works in mobile sensor networks. MCL works well in mobile sensor networks as long as the speed of movement is not very low. The paper proposes two possible filter approaches. The first approach adopts Arriver and Leaver information (i.e. beacon transmits both its current location and its location at the previous time step in each announcement). The second adopts the current neighborhood (beacons and unknowns nodes) information. The other range-free algorithm called MSL proposed by Rudafshani *et al.*, works well when some or all nodes are static or mobile. Unlike MCL in the prediction stage, the parameter α is needed for MSL to work well when no sensors move. Each node maintains a set of weighted samples denoting its possible locations and its weight is determined using the current neighborhood information. Though these approaches are not especially for a single mobile-assisted sensor networks, our work is inspired by above approaches. In the prediction stage, we adopt the parameter α proposed by MSL. In the update stage, we adopt Arriver and Leaver information proposed by MCL.

6. Conclusions

In this paper, we propose two range-free, distributed and probabilistic mobile beacon-assisted localization approaches for static WSNs, MBL and A-MBL. Evaluation results show that the accuracy of A-MBL outperforms MBL, MSL and ADO in static WSNs when all of them use only a single mobile beacon for localization. As future work, two new issues will be considered. First, whether the use of information from unknown nodes (especially the neighbor nodes) which may have greater communication cost to increase efficiency and accuracy in mobile beacon-assisted localization needs further research. Second, though adopting predefined adjustment tables in A-MBL is convenient and effective, obtaining these tables is difficult. We will also consider some of self-adaptive mechanism in our approaches to achieve more flexibility.

Acknowledgements

This work is supported by the National Basic Research Program of China (973 Program) under grant No. 2006CB303000.

References

1. Volgyesi, P.; Balogh, G.; Nadas, A.; Nash, C.B.; Ledeczi, A. Shooter localization and weapon classification with soldier-wearable networked sensors. In *The 5th International Conference on Mobile Systems, Applications and Services*, New York, NY, USA, 2007; pp. 113–126.
2. Manes, G.; Fantacci, R.; Chiti, F.; Ciabatti, M.; Collodi, G.; Palma, D.D.; Manes, A. Enhanced system design solutions for wireless sensor networks applied to distributed environmental monitoring. In *The 32nd IEEE Conference on Local Computer Networks*, Washington, DC, USA, IEEE, 2007; pp.807–814.
3. Song, G.M.; Zhou, Y.X.; Ding, F., Song, A.G. A Mobile Sensor Network System for Monitoring of Unfriendly Environments. *Sensors* **2008**, *8*, 7259-7274.
4. Mechitov, K.; Sundresh, S.; Kwon, Y.; Agha, G. Poster abstract: cooperative tracking with binarydetection sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, New York, NY, USA, 2003; pp. 332–333.
5. Wang, X.; Wang, S.; Ma, J.J. An improved particle filter for target tracking in sensor systems. *Sensors* **2007**, *7*, 144-156.
6. Szewczyk, R.; Mainwaring, A.; Polastre, J.; Anderson, J.; Culler, D. An analysis of a large scale habitat monitoring application. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2004; pp. 214–226.
7. Mechitov, K.; Kim, W.; Agha, G.; Nagayama, T. High-frequency distributed sensing for structure monitoring. In *The First International Workshop on Networked Sensing Systems*. Tokyo, Japan, 2004.

8. Kwon, Y.; Agha, G. Passive localization: Large size sensor network localization based on environmental events. In *IEEE International Conference on Information Processing in Sensor Networks*, St. Louis, Missouri, USA, , 2008; pp.3–14.
9. Rudafshani, M.; Datta, S. Localization in wireless sensor networks. In *The 6th international conference on Information processing in sensor networks*, Cambridge, Massachusetts, USA, ACM, 2007; pp.51–60.
10. Kim, K.; Lee, W. MBAL: A mobile beacon-assisted localization scheme for wireless sensor networks. In *The 16th International Conference on Computer Communications and Netw*orks, Honolulu, Hawaii USA, 2007; pp.57–62.
11. Priyantha, N.B.; Balakrishnan, H.; Demaine, E.D.; Teller, S. Mobile-assisted localization in wireless sensor networks. In *The 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, USA, 2005; pp.172–183.
12. Sichitiu, M. L.; Ramadurai, V. Localization of wireless sensor networks with a mobile beacon. In *The First IEEE Conference on Mobile Ad-hoc and Sensor Systems*, Philadelphia, USA, 2004; pp.174–183.
13. Huang, R.; Zruba, G.V. Monte carlo localization of wireless sensor networks with a single mobile beacon. *Wirel. Netw***. 2008**, doi: 10.1007/s11276-008-0096-3.
14. Peng, R.; Sichitiu, M.L. Robust, probabilistic, constraint-based localization for wireless sensor networks. In *The Second Annual IEEE Sensor and Ad Hoc Communications and Networks*, Santa Clara, California, USA, 2005, pp.541–550.
15. Xiao, B.; Chen, H.K.; Zhou, S.G. A walking beacon-assisted localization in wireless sensor networks. In *International Conference on Communications*, Glasgow, Scotland, 2007; pp.3070–3075.
16. Beeson, P.; Murarka, A.; Kuipers, B. Adapting proposal distributions for accurate, efficient mobile robot localization. In *IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA, 2006; pp.49–55.
17. Fox, D. Adapting the sample size in particle filters through KLD-sampling. *Int. J. Rob. Res.* **2003**, *22*, 985–1003.
18. Arulampalam, S.; Maskell, S.; Gordon, N. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188.
19. Hu, L.; Evans, D. Localization for mobile sensor networks. In *The 10th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2004; pp.45–57.
20. Kitagawa, G. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *J. Comput. Graph. Stat.* **1996**, *5*, 1–25.
21. Fox, D.; Burgard, W.; Dellaert, F. Monte carlo localization: Efficient position estimation for mobile robots. In *The Sixteenth National Conference on Artificial Intelligence*, Orlando, Florida, USA, 1999; pp.343–349.
22. Caballero, F.; Merino, L.; Maza, I.; Ollero, A. A Particle Filtering method for Wireless Sensor Network Localization with an Aerial Robot Beacon. In *IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, 2008; pp.596–601.

23. Marinakis, D.; Meger, D.; Rekleitis, I.; Dudek G. Hybrid Inference for Sensor Network Localization using a Mobile Robot. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, Vancouver, British Columbia, Canada, 2007; pp.1089–1094.
24. Ihler, A.T.; Fisher, J.W.; Moses, R.L.; Willsky, A.S. Nonparametric Belief Propagation for Self-Localization of Sensor Networks. *IEEE J. Sel. Area Comm.* **2005**, *23*, 809–819.
25. Li, M.; Liu, Y. Rendered path: range-free localization in anisotropic sensor networks with holes. In *The 13th Annual ACM International Conference on Mobile Computing and Networking*, Montreal, Quebec, Canada, ACM, 2007; pp.51–62.
26. Stoleru, R.; He, T.; Stankovic, J. A. Walking gps: A practical solution for localization in manually deployed wireless sensor networks. In *The 29th Annual IEEE International Conference on Local Computer Networks*, Tampa, FL, USA, 2004; pp.480–489.
27 Coates, M. Distributed Particle Filters for Sensor Networks. In *Third International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, USA, 2004; pp. 99–107.