

# OTMCL: Orientation Tracking-based Monte Carlo Localization for Mobile Sensor Networks

Marcelo H. T. Martins

Institute of Industrial Science

The University of Tokyo, Japan

Email: martins@mcl.iis.u-tokyo.ac.jp

Hongyang Chen

Institute of Industrial Science

The University of Tokyo, Japan

Email: hongyang@mcl.iis.u-tokyo.ac.jp

Kaoru Sezaki

Center for Spatial Information Science

The University of Tokyo, Japan

Email: sezaki@iis.u-tokyo.ac.jp

**Abstract**—In this paper we consider a probabilistic approach to the problem of localization in wireless sensor networks (WSNs) consisting of mobile nodes, and propose a distributed algorithm that help unknown nodes determine their positions. Our proposal is a range-free positioning system based on heading data, which can be provided by a variety of orientation-tracking sensors of different precisions, applied to the Monte Carlo sampling technique. Simulation results indicate that the intermix of orientation data with sampling and filtering augments the overall accuracy of the positioning system, while maintaining a low communication cost.

**Index Terms**—sensor networks, localization, mobility, Monte Carlo sampling, orientation tracking.

## I. INTRODUCTION

Accurate localization is a necessary feature for several sensor network applications and protocols. The importance of identifying the physical location of a sensor node arises from the need to associate events and collected data to their occurrence location. For instance, routing protocols can take advantage of location knowledge to optimize relaying paths based on the shortest distance between nodes [1]. Location awareness is also an important technology for other applications, such as geocasting [2], animal tracking [3], and mapping.

Although accurate localization techniques such as global navigation satellite systems (GNSS) exist (e.g. GPS or GLONASS receivers), their cost, large size and energy consumption profile prevent them from being deployed in large-scale, low-power networks. Furthermore, a GNSS device requires a minimum number of satellites along the line of sight to determine its location. This may turn impossible in scenarios such as indoor or underground venues, and urban canyons. An alternative is to employ a small set of nodes who are aware of their own coordinates (hereafter called *anchors*) which will cooperate with unknown nodes and help them estimate their positions.

The earlier localization algorithms primarily concentrated on static networks. In mobile scenarios, these algorithms often fail to obtain a single position estimate with reasonable accuracy. Instead of directly determining location, we can represent a location estimate as a probability distribution over the deployment area. Probabilistic methods provide a better approximation than deterministic algorithms because they explicitly consider the impreciseness of location estimates. The

output of such methods represents an area where a sensor node might be located with a certain likelihood.

In this paper, we focus on the application of localization systems in full-mobile scenarios. Our algorithm, Oriented Tracking-based Monte Carlo Localization (OTMCL), applies the Sequential Monte Carlo method (also known as *particle filtering*) to achieve localization of moving nodes based on the network connectivity information distributed by anchors. We propose the use of orientation data as hints with means to “tame” the particle filter and provide better control over the posterior distribution representing a node’s position. We perform simulation experiments and show that the insertion of orientation data into the particle filter improves the performance of the positioning system.

The rest of the paper is organized as follows. We present an overview of localization algorithms developed for static and mobile sensor networks in Section II. Section III provides details on how the particle filtering technique can be utilized to estimate a node’s position and some of its shortcomings. In Section IV, we derive our proposal using heading data as a guide to obtain accurate location estimates. Section V validates our algorithm by means of experimental results based on simulation. Finally, we present our conclusions and future work in Section VII.

## II. RELATED WORK

The earlier localization algorithms have primarily targeted both static and mobile ad hoc networks (MANETs). Centralized methods have been proposed to solve the localization problem [4]–[6]; however, the distributed aspect of WSNs, added to the limited computational capabilities of individual nodes make them difficult to be effectively used in this environment.

A variety of localization techniques rely on distributed computation over distance measurements. This class of localization methods, named range-based algorithms, requires special hardware to measure input that will be transformed into distance measurements. The majority of these techniques relies on the RSS [7] or Time of Arrival (ToA) of signals due to their low cost, while others depend on more robust (and more expensive) technologies, such as Angle of Arrival (AoA) and Time Difference of Arrival (TDoA) [8]. Methods such as DV-Hop, DV-Distance, and Euclidean [9] operate on

these data to estimate the absolute node locations. A common characteristic shared by these techniques is that they require a relatively dense network in order to achieve high accuracy.

A different class of localization methods rely solely on connectivity information. Range-free algorithms usually assume that all nodes have the same transmission range, and the localization problem becomes a problem of fixing a set of unit disks into a graph given the connectivity constraints. Centroid [10] estimates the location of an unknown node as the average of its neighbors' location, while APIT [11] estimates the node location by isolating the deployment area using various triangles formed by anchors.

One common characteristic of the aforementioned works is that they assume a static network. When mobility is considered, they ultimately fail as location information rapidly becomes stale. An alternative method is to localize sensor nodes using a mobile beacon. In this method, a vehicle carrying a sensor travels through the deployment area while broadcasting its position along the way [12]. It assumes all other nodes are static. The problem with such approach is that the anchor sparseness restrict nodes to localize themselves only when the mobile beacon is inside their ranging area. In case of mobile unknown nodes, this becomes a serious problem when the movement patterns of the mobile beacon and the interested readers are opposite.

Different probabilistic approaches have been proposed for the localization problem in mobile WSNs, while the most successful ones are based on the particle filtering idea. For instance, Hu and Evans [13] proposed MCL, a range-free localization algorithm for mobile sensor networks based on the Sequential Monte Carlo method (SMC). The Monte Carlo method has been extensively used in robotics, where a robot estimates its location based on its motion, perception and possibly a pre-learned map of its environment. Hu and Evans extended the Monte Carlo method to support the localization of moving sensors in a free, unmapped terrain. MCL represents the posterior distribution of possible positions using a set of weighted samples and uses the Monte Carlo method for sampling, filtering and estimating nodes' positions.

Despite being somewhat accurate, MCL's efficiency can be improved. Drawing samples is a long and tedious process that can easily drain a lot of energy, especially from low-power devices such as sensor nodes. While MCL uses anchor information only in the filtering stage, MCB [14] also applies it to constrain the area from which the samples are drawn in the prediction stage. As a result, sampling efficiency is increased, as less samples are rejected by the filtering process, saving node's energy. Zhang et al. [15] go even further. In their approach, which we name ZJL after the authors, they extended MCB by removing the negative effects from two-hop connections to reduce the sampling area.

Depending on the anchor density of a network, there will be occasions when a regular node will not communicate with any anchor, which undermines the localization process. One idea to avoid the lack of coverage is to also allow regular nodes to participate in the cooperative localization process. In ZJL,

one-hop regular neighbors are also taken into account in the prediction and filtering phases of SMC. Another related work, MSL [16], also proposes using a set of probable locations (samples) for each node. These samples are assigned weights that estimate their quality. The main difference between MCL and MSL is that MSL also considers connectivity information from non-anchor neighbors for filtering. It adopts an aggressive formula to estimate the quality of a location estimation and allows only filtering contributions from well-localized regular nodes.

### III. MOTIVATION

The Monte Carlo Localization method is a recursive Bayesian filter that estimates the posterior distribution of a node's location conditioned on observations from neighbors. The key idea is to represent this distribution by a set of samples or particles with non-negative weights, drawn according to the previous distribution over the observations from neighbors.

The Sequential Monte Carlo (SMC) method, also known as particle filtering, provides a convenient and effective solution for non-Gaussian, non-linear and multidimensional systems. Unlike other Bayesian filters, the sequential Monte Carlo has low processing and memory requirements, it is easy to implement and suitable for parallel processing [17]. These characteristics make it easier to accommodate to low-power networks, such as WSNs. Basically, SMC is divided into three stages that are triggered as new movement of a mobile node is detected: *prediction*, *filtering*, and *resampling*.

In the prediction stage, the importance sampling function draws location samples based only on the prior distribution  $p(x_{t-1})$ . For each sample of  $p(x_{t-1})$ , we apply a random mobility model to generate a new set of samples. As nodes are unaware of their speed and direction, the possible predicted positions based on step  $t - 1$  are within a circular region with origin  $x_{t-1}$  and radius  $v_{\max}$  ( $v_{\max}$  is the only known constraint).

$$p(x_t|x_{t-1}) = \begin{cases} \frac{1}{\pi v_{\max}^2} & \text{if } d(x_t, x_{t-1}) < v_{\max}, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Here  $d(x_t, x_{t-1})$  is the Euclidean distance between the current prediction  $x_t$  and  $x_{t-1}$ . As it can be inferred, speeds are distributed uniformly in the interval  $[0, v_{\max}]$ , which reflects the increasing uncertainty about the node's location. Consequently, some of the derived samples do not correspond to the actual location. In the case of MCB and ZJL, the prediction step assumes that the node should move inside a smaller prediction area, called *bounding box*; still, movement data is not available. Fig. 1 illustrates node  $n$  moving from position  $x_{t-1}$  to  $x_t$ . Since the motion model is unknown, samples like  $x_t^{i-1}$  are drawn, which negatively contributes to the localization accuracy. Fig. 2 shows another scenario common of sample degeneracy. Given that  $x_{t-1}'$  is the previous estimated location of node  $n$  and its real location after movement is  $x_t$ , most of the particles generated from  $p(x_t'|x_{t-1}')$

will fail the filtering test. Both cases could be resolved if  $n$  had any notion about its direction.

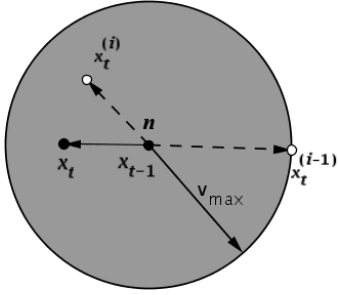


Figure 1. Negative contribution from  $x_t^{(i-1)}$  [18].

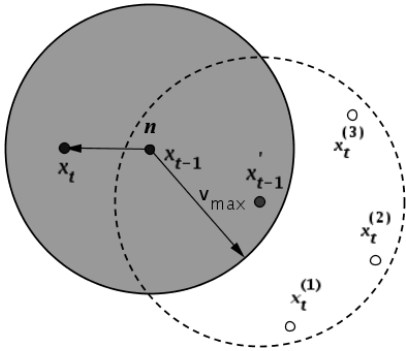


Figure 2. Far-away estimation reduces the chances of obtaining good samples.

In order to remove the invalid samples, the *filtering* stage is performed. Predictions that do not lie within the connectivity constraints imposed by neighbors are discarded. These constraints come from the unit disk model (or unit ball model, in the case of 3D localization) with transmission range  $r$ . Two-hop-away neighbors also collaborate on the filtering process by imposing further restrictions. In the case of MCL and MCB, only anchors nodes determine the filtering boundaries, while ZJL and MSL also rely on one-hop and two-hop regular neighbors whose location estimates are relatively accurate.

After filtering, there may be fewer than  $N$  possible location estimates. A common problem with the sequential importance sampling is sample degradation. After a few iterations of the algorithm, only a few particles will have a normalized observation likelihood above zero. This implies that a large computational effort is devoted to updating particles which are invalid, i.e., they do not obey to the connectivity observations described by their neighbors. *Prediction* and *filtering* repeat until a minimum number  $N$ , called the effective sample size, is found or the timer used to extract them expires.

#### IV. ALGORITHM OVERVIEW

Based on the liabilities of the Monte Carlo-based localization methods and the promises of navigation systems, we propose a new localization algorithm, named Orientation Tracking-based Monte Carlo Localization (OTMCL), that

merges both ideas to decrease the error estimate of a node's position.

Before predicting the position of an unknown node, OTMCL defines the initial prediction area as a bounding box in a similar fashion to MCB. The bounding box is the computationally simplest form of all linear bounding containers and is used in many geometric applications, such as ray tracing, hidden object detection and collision avoidance. The prediction area is described by the intersection of the square-approximated transmission ranges of a node's one-hop and two-hop anchor neighbors. The coordinates of the rectangle are given as follows:

$$x_{\min} = \max_{i=1}^n x_i - r, \quad x_{\max} = \min_{i=1}^n x_i + r \quad (2)$$

$$y_{\min} = \max_{i=1}^n y_i - r, \quad y_{\max} = \min_{i=1}^n y_i + r \quad (3)$$

where  $(x_i, y_i)$  are the coordinates of the  $i^{\text{th}}$  anchor neighbor, and  $r$  is its communication range. In case of two-hop anchor neighbors, we substitute  $r$  with  $2r$ . Without loss of generality, the minimal area described by the rectangle can be substituted by the minimal volume when performing 3D localization. When the localization method is triggered for the first time,  $p(x_0|x_{t-1})$  is not defined. In the initialization phase, samples are extracted from the bounding box and the initial location is set as the average of the collected samples. After the initialization phase, the following position updates will consist of repetitions of the *prediction* and *filtering* stages.

The main difference between our proposal and previous methods is the way samples are generated in the prediction stage. Up to this point, no assumption about the direction a moving node takes at each step has been made, leading to sampling of random values using a uniform distribution over the whole prediction area.

Orientation data can be extracted from a variety of specialized hardware: compass needles, digital compasses, rate gyros, and complete dead reckoning navigation sensors. Keir et. al [19] showed it is possible to accurately maintain track of orientation using a Analogue Devices ADXL213 accelerometer, similar to the model used in today's commercial sensor nodes. Even so, heading inaccuracy happens and can affect the performance of OTMCL. Heading inaccuracy arises from several factors. For instance, rate gyros begin to drift out within seconds after the last contact to its reference landmark. Magnetic compasses interference sources include the declination angle – the difference between the Earth's geographic north and the magnetic north –, which can be as large as  $25^\circ$  depending on the location; time-varying distortions of the magnetic field due to moving, ferrous materials or unpredictable electrical currents; and temperature changes.

Another source of heading variance arises from twists with each step of the body carrying the navigation sensor, and the fact that the sensor is often moving on an uneven terrain. Accelerometers or gimbals, if available, can help compasses detect pitch and roll angles, and allow for tilt compensation. An alternative is to use a gyroscope to maintain a known

inertial reference orientation at all times. Given the pitch ( $\theta$ ) and roll ( $\phi$ ) rotational angles, we can transform the magnetic components to the local level plane coordinate system, and then, determine the azimuth, or compass direction ( $\gamma$ ), as follows [20].

$$\begin{aligned} X_H &= X \cos(\phi) + Y \sin(\theta) \sin(\phi) - Z \cos(\theta) \sin(\phi) \\ Y_H &= Y \cos(\theta) + Z \sin(\phi) \\ \gamma &= \arctan(Y_H/X_H), \end{aligned}$$

where  $(X_H, Y_H)$  are the horizontal components of the Earth's magnetic field. Still, MEMS accelerometers usually do not provide the required precision to distinguish such source of errors when the body presents a minimalistic movement. In addition, the amount of hardware necessary to design such a system, added to a large power consumption, might prevent its implementation for financial or physical reasons. A magnetic heading sensor may also desensitize periodically, and a recalibration becomes necessary. Finally, even in the case of using high-accuracy heading sensors, their precision still depends on the A/D conversion resolution of the interface between the orientation sensor and the node per se.

Being this way, we adapt our localization system to handle heading data of different accuracies and do not assume the usage of any specific orientation sensor. As a node moves, external factors may divert the measured orientation ( $\alpha$ ) from the actual one ( $\gamma$ ), even though this difference cannot be measured online. For this reason, in order to cover all possible directions a node may traverse, we must consider all the cases where  $\alpha$  is less or equal to the extremes of the sensor's variance  $\beta$ . The value of  $\beta$  is determined by the largest angular offset for a certain class of orientation sensor, and we can extract such value from the sensor's manual or define it empirically. Fig 3 illustrates an example of a node moving to a certain direction and how the likelihood presence area is estimated. Eq. 4 describes mathematically the sector responsible for the sampling area  $A$ .

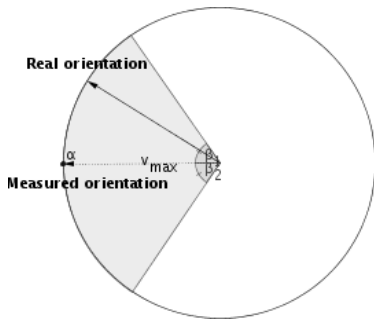


Figure 3. Estimating OTMCL's sector area.

$$A = \frac{1}{2} \int_{\alpha-\beta}^{\alpha+\beta} v_{\max}^2 d\theta \quad (4)$$

From (4), we derive the transition equation of OTMCL:

$$p(x_t|x_{t-1}) = \begin{cases} \frac{1}{A} & \text{if } d(x_t, x_{t-1}) < v_{\max}, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

As the prediction area is inversely proportional to the probability of drawing a valid sample, OTMCL promises a higher location accuracy than MCL and others. A valid sample is a sample that passes the filtering test based on the node's one-hop and two-hop neighboring anchors' positions. The filtering condition is given as follows:

$$filter(x^i) = \{\forall s \in S, d(x^i, s) \leq r \wedge \forall t \in T, r \leq d(x^i, t) \leq 2r\},$$

where  $S$  is the set of one-hop anchors of node  $n$  at step  $t$  and  $T$  is its set of two-hop anchors.

If our proposal cannot validate enough samples, we replace the sampling area from the heading-derived sector to the bounding box described by Eq. 2 and repeat the *prediction* and *filtering* stages.

The algorithm implementation must set a bound on the number of times a node tries to draw samples if the sample set does not contain the required minimum number of particles to estimate the final location. For example, the authors of MCL, MCB, MSL, and ZJL chose a set size  $N$  of 50. MCL tries to fill the set with at most 20000 attempts per sample. In order to save battery power, the authors of MCB tried to ensure that the sample set is full in 50 to 100 tries per sample. Nonetheless, there may be occasions in which a node is unable to draw any valid sample from the prediction-filtering loop. In the case of the aforementioned algorithms, the node is repositioned at the middle of the deployment area and the algorithm restarts at the initialization phase. This heavily penalizes the localization performance. In order to avoid this, OTMCL takes a different approach. Based on the current orientation data, we perform a crude dead reckoning, choosing a random velocity from the interval  $[0, v_{\max}]$  to estimate the node's next step. This means that OTMCL guarantees localization coverage even when filtering fails or anchor nodes are temporarily unavailable. List 1 recapitulates our ideas in algorithmic form.

## V. EVALUATION

In this section, we present a complete performance evaluation of OTMCL. First, we describe the scenario used in our analyses, and then, by using simulation, we study the performance of OTMCL as a function of the nominal communication range and deployment parameters. We also present a comparative study with four other state-of-the-art localization techniques for mobile WSNs.

### A. Simulation Scenario

WSNs nodes are generally deployed in an ad-hoc manner. In our simulation, this is no exception; we randomly distribute 320 nodes over an obstacle-free area of  $500 \times 500$  square units. With a view to reduce costs, nodes with special capabilities are kept to a minimum. Being this way, we select 32 nodes to act as anchors. In a real deployment, anchor nodes will probably have a larger battery capacity, as they are responsible

**Algorithm 1** The OTMCL core

---

```

1: procedure OTMCL(Neighbors,  $\alpha$ ,  $\beta$ )
2:   build bounding box
3:   if node  $\neq$  localized  $\vee p(x_{t-1}) == \emptyset$  then
4:     if Neighbors  $\neq \emptyset$  then
5:       find  $N$  samples from bounding box
6:     end if
7:   else
8:     find  $N$  samples from sampling sector( $\alpha$ ,  $\beta$ )
9:     if  $|p(x_t)| < N$  then
10:      find remaining samples from bounding box
11:    end if
12:    if  $(|p(x_t)| < N)$  then
13:       $x_t = x_{t-1} + \mathcal{U}(0, v_{\max}) * \mathcal{N}(\alpha, \beta)$ 
14:    end if
15:  end if
16:   $x_t = \frac{\sum w_i \times x^i}{N}$ 
17: end procedure

```

---

for transmitting location information to other nodes, and also have access to accurate and fresh location information. Since it is difficult to obtain an ideal placement to cover most of the unknown nodes, anchor nodes are also made mobile. For a deployment over a large area, unmanned vehicles or airplanes can act as mobile beacons. We assume that unknown nodes are not aware of their initial position. This is a reasonable premise if we consider the case of nodes being dropped from the air or the burden of initializing a network consisting of hundreds or thousands of nodes. Node mobility is modeled using a modified version of the Random Waypoint Model (RWP), where sensors do never stop at waypoints in order to prevent the velocity decay problem [21]. The performance of OTMCL is measured in terms of location error for a wide range of displacement parameters and node distributions. The location error is defined as the average difference between the estimated location and the actual location of the unknown nodes. We normalize the estimation to the radio range  $r$ . Hence, the location estimation error of a sensor network composed of  $N$  unknown nodes is given by:

$$\delta = \frac{\sum_{n=1}^N \sqrt{(x'_n - \hat{x}_n)^2 + (y'_n - \hat{y}_n)^2}}{N \times r}. \quad (6)$$

Where  $(x'_n, y'_n)$  are the estimated coordinates of node  $n$ , and  $(\hat{x}_n, \hat{y}_n)$  are its real coordinates. Table I summarizes our simulation parameters.

### B. Simulation Results: Performance Study

1) *Impact of Orientation Variance*: Fig. 4 plots the overall performance of OTMCL as a function of the orientation variance of the inertial sensors. Axis  $X$  describes the angle variance (in degrees) of compasses with different accuracies. Axis  $Y$  presents the estimation error relative to the communication range. From the results, we can see that OTMCL is effective on limiting the estimation error increase as the inertial system becomes less accurate. We infer that OTMCL

Table I  
SIMULATION PARAMETERS

Simulation area	$500 \times 500$
Total number of nodes	320
Number of anchors	32
Node placement	Uniform
Number of samples	50
Anchor density	1
Node density	10
Routing protocol	AODV
Communication range	50
Maximum velocity	10
Mobility model	Random Waypoint ( $pause_t = 0$ )
Communication model	Unit disk graph

is capable of retrieving a good set of valid samples without reaching the limits of sampling trials thanks to the heading-derived sampling area. For the following analyses, we select three categories of nodes based on their orientation variance: nodes with a (probably) expensive, high-accuracy heading sensor ( $\beta = 10^\circ$ , e.g., gyroscope, anisotropic magnetoresistive (AMR) sensors); nodes with low-cost, poorly-calibrated sensors ( $\beta = 30^\circ$  and  $\beta = 45^\circ$ ), and nodes with non-calibrated, prone-to-bias, magnetic sensors (e.g., a simple magnetic needle,  $\beta = 90^\circ$ ).

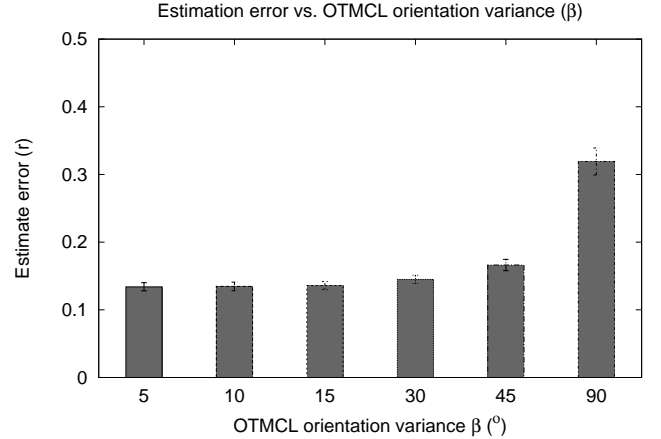


Figure 4. Accuracy of OTMCL over different  $\beta$ .

### C. Simulation Results: Comparative Study

We compare OTMCL with four other state-of-the-art particle filtering-based localization techniques designed for full mobility: MCL, MCB, MSL, and ZJL.

1) *Communication Cost*: Previous research has shown that a major portion of power consumption of sensor nodes comes from its radio transceiver. In order to increase network lifetime, several proposals have been considered, such as optimizing the time when radio interfaces should be kept awake, turning off auxiliary components when they are not being used, among

Table II  
COMMUNICATION COST ANALYSIS

Algorithm	Avg. # of TX-messages from neighbors
MCL	~ 4.5
MCB	~ 4.5
ZJL	~ 10.5
MSL	~ 21
OTMCL	~ 4.5

others. Nevertheless, when nodes need to transmit or receive packets, there is nothing that can be done to prevent battery discharging. In this section, we analyze the average power consumption of each particle filtering-based algorithm considering the number of messages forwarded per node. Table II shows the average number of messages relayed to node  $n$ , where  $n$  is the final destination of a message, by its one-hop and two-hop anchors, to guarantee its localization. We can infer that ZJL and MSL are power-hungrier than the other algorithms. While in the cases of MCL, MCB and OTMCL only anchor nodes controllably flood the network with beacon messages, ZJL and MSL assume that regular nodes should also cooperate. In the following sections, we will show that OTMCL can achieve the best localization performance without the need of a heavy transmission payload.

2) *Convergence Time*: The second simulation analysis concerns the location accuracy convergence. As it can be seen from Fig. 5, the initial sharp decline on the estimation error curves of the SMC-based algorithms is due to the anchor beacons being incorporated into the sequential Monte Carlo process. This is followed by a stable phase declaring the balance between updates to the posterior location distribution and uncertainty introduced by mobility (the *nominal performance point*).

We can see from the results that OTMCL outperforms MSL when  $\beta = 10^\circ$  or  $30^\circ$ , presents a similar performance when  $\beta = 45^\circ$  and is still the best option among the anchor-only-based algorithms when  $\beta = 90^\circ$ . This can be explained by the higher guarantee from OTMCL that the location estimate is on the right track in terms of orientation. On the other hand, MSL shows a rapid convergence to stability, as it uses much more connectivity information from anchor and non-anchor nodes. The price paid by MSL is the reliance on a larger communication overhead, which can be too expensive when we consider the network lifetime.

3) *Node density*: Fig. 6 shows the impact of node density variation on the accuracy of each algorithm while the anchor density is kept constant. While MCL, MCB and OTMCL are little affected by the node density, ZJL and MSL take advantage of the abundance of connectivity information from regular nodes (in the case of ZJL, only one-hop regular neighbors are considered). The average number of simultaneously cooperating neighbors for MSL varies between 5 and 96, and for ZJL from 5 to 25, while in the case of MCL, MCB and OTMCL the number of useful neighbors (anchors and two-hop

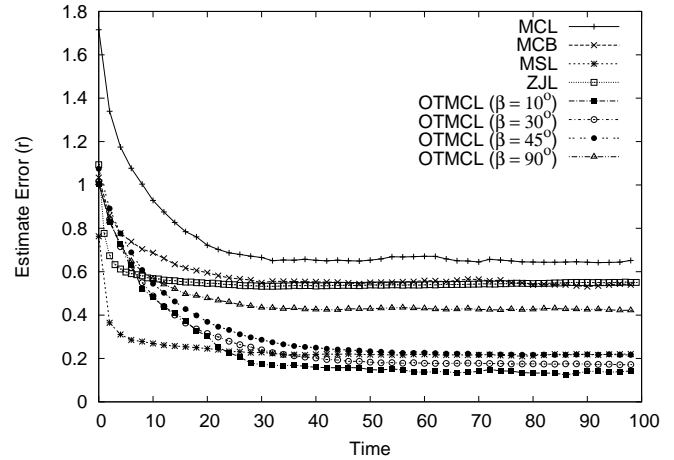


Figure 5. Location accuracy over time.

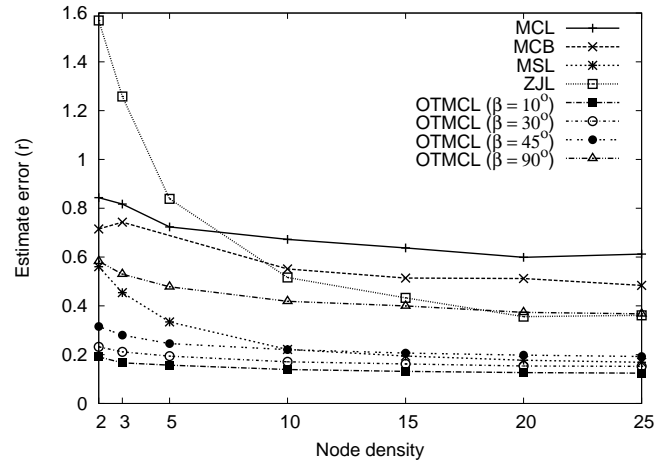


Figure 6. Impact of node density over estimation error.

anchors) does not change much (between 2 and 3 for each time step).

Just like in the previous analysis, OTMCL running on nodes with high- and medium-grade orientation sensors ( $\beta = 10^\circ, 30^\circ$  or  $45^\circ$ ) present a comparable or even better accuracy than MSL with a fraction of message transmissions. Moreover, OTMCL ( $\beta = 90^\circ$ ) performs better than the other anchor-only Monte Carlo methods (MCL and MCB).

4) *Anchor density*: Increasing anchor density facilitates the retrieval of location information by regular nodes. The anchor density is defined as follows:

$$a_d = \frac{\pi r^2 a_m}{TotalArea}, \quad (7)$$

where  $a_m$  is the total number of anchors, and  $TotalArea$  is the total deployment area. As an example, a anchor density  $a_d$  of 0.5 corresponds to 16 anchors in a network of 320 nodes. Fig. 7 shows the average estimate error of different localization algorithms when the anchor density changes. We can see that all SMC-based algorithms take advantage of a higher anchor density, as more position announcements are

available for filtering. From the results, we can conclude that OTMCL is capable of achieving a decent performance even when anchor nodes are scarce and requires much less message exchanges than MSL, whose accuracy is similar.

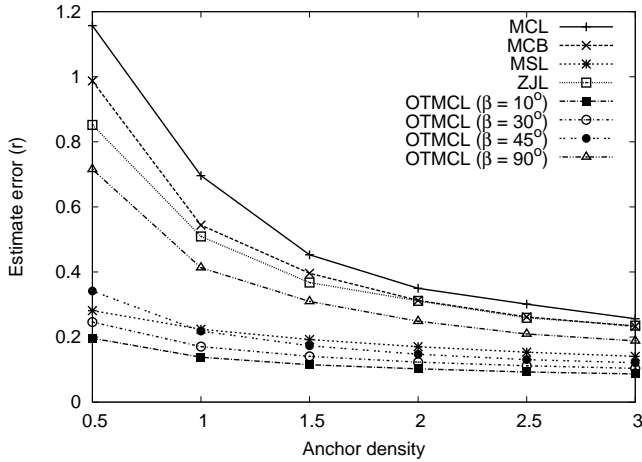


Figure 7. Impact of anchor density over estimation error.

5) *Node speed*: As we can see from Fig. 8, for all the SMC-based algorithms the maximum speed variation does not affect much the location accuracy. At higher speeds, nodes hear new anchors more often which adds new information to the filtering phase, keeping the sample set updated. On the other hand, as larger distances are traversed at each time unit, the sampling area becomes larger, decreasing the algorithm’s accuracy. MCL is the only method that takes advantage of increasing speeds when they are still low. The high- and medium-grade versions of OTMCL also perform better than MSL in this analysis.

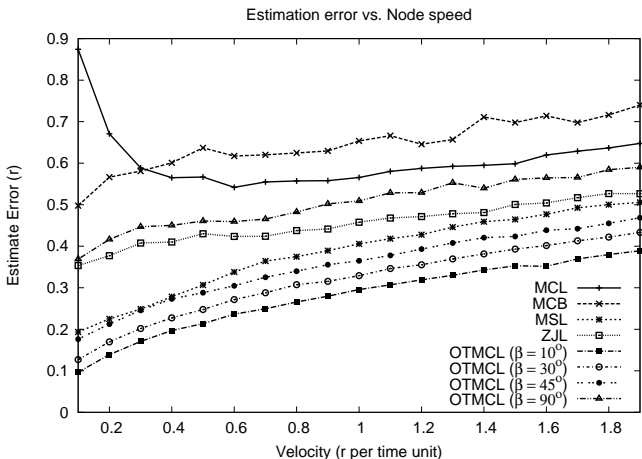


Figure 8. Impact of node speed over estimation error.

6) *Communication Irregularities*: Variability in actual radio transmission patterns can have a substantial impact on localization accuracy depending on the localization technique. Unlike the perfect circles of radius  $r$  assumed in our previous

experiments, the measured reception distance of radios can vary substantially with environmental conditions and antenna irregularities.

In this section, we verify the impact of radio irregularity on estimate error. We consider an idealized radio model proposed by He et al. [11] in which all nodes within half the maximum transmission range of anchors are guaranteed to hear from the anchor, whereas nodes between the maximum radio range and half of that range may or may not hear from the anchor depending on the radio pattern in that direction. The degree of irregularity (DOI) parameter is defined as the maximum radio range variation per unit degree change in direction. For instance, if  $DOI = 0.1$ , then the actual communication range in each direction is randomly chosen from  $[0.9r, 1.1r]$ . We acknowledge that this is a crude model of real radio ranges, which are affected by noise, fading, antenna directionality, the presence of obstacles and many other factors. Accurate modeling of radio ranges [22], [23] is beyond the scope of this work and is left as future work. We simulate the DOI variation parameter using a Gaussian distribution.

From Fig. 9 we can see that MCL, MCB, ZJL and MSL algorithms can only tolerate small values of DOI ( $\leq 0.2$ ). When it gets higher, their accuracy severely degrades. The higher DOI we have, the higher is the number of links with underestimated and overestimated transmission ranges. In the case of underestimated ranges, nodes which are distant to each other by less than  $r$  and should be exchanging messages when  $DOI = 0$ , are prone to not communicate with their one-hop neighbors. In this case, unknown nodes have a lesser chance to receive observations from anchor nodes which might lead to a coarser localization or, in extreme situations, to the re-initialization of the localization process. When we have overestimated ranges, nodes which are farther than the nominal values  $r$  or  $2r$  become neighbors. As a consequence, the localization process fails because the sampled locations do not obey the filtering boundaries.

As for OTMCL, despite the message exchange hardships, it is still able to filter a larger number of valid samples than the other SMC-based algorithms, thanks to the reduced sampling area guaranteed by the orientation-tracking sensors. In extreme situations ( $DOI = 0.5$ ), when no samples can be validated, the node infers its next location based on dead reckoning, instead of repositioning itself at the middle of the deployment area.

## VI. OTHER CONSIDERATIONS

*Deployment Cost*. One of the main concerns on sensor network design is guaranteeing that the final cost of an integrated system is not overly expensive, so that it can be deployed in a large scale without becoming a huge financial burden. At first, our proposal might resemble some of the range-based localization algorithms, which increment the cost of the system by requiring specialized hardware for location estimates. Nonetheless, contrary to these techniques, orientation sensors are not used exclusively for localization and the cost of adding them can be amortized if they are applied to

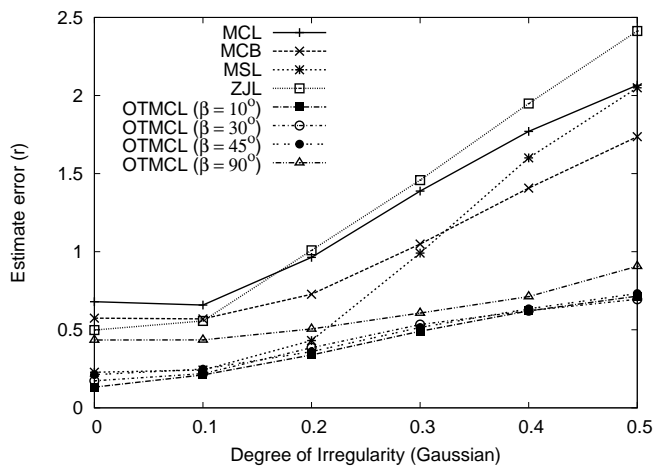


Figure 9. Impact of communication irregularities.

other types of sensing tasks. Examples of such applications include body posture detection and activity monitoring, suitable for health-care applications, and augmented or virtual reality applications.

## VII. CONCLUSION

We present a distributed, probabilistic approach to the localization problem in mobile WSNs. Our proposal is based on the Monte Carlo sampling idea to probabilistically represent and update a moving node's position. We take advantage of orientation data provided by sensors of different accuracies. Simulation results show that the proposed algorithm can achieve similar or better performance than other state-of-the-art algorithms based on the particle filtering concept, while keeping a low-communication profile. As future work, we intend to explore the variability of the required number of samples per estimation according to the accuracy of the previous location distribution. We also intend to implement and test our algorithm in a field deployment to verify the influence of physical obstacles and other sources of communication anisotropy over the estimation error.

## ACKNOWLEDGMENTS

This research is partially supported by CREST's "Advanced Integrated Sensing Technology" project of Japan's Science and Technology Agency. The authors would like to thank David Evans, Aline Baggio, Masoomeh Rudafshani, and Shigeng Zhang for providing assistance with MCL, MCB, MSL, and ZJL, respectively.

## REFERENCES

- [1] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *Wirel. Netw.*, vol. 6, no. 4, pp. 307–321, 2000.
- [2] —, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," in *WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*. Washington, DC, USA: IEEE Computer Society, 1999, pp. 101–110.

- [3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," *SIGARCH Comput. Archit. News*, vol. 30, no. 5, pp. 96–107, 2002.
- [4] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *MobiHoc '03: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*. New York, NY, USA: ACM, 2003, pp. 201–212.
- [5] L. Doherty, K. S. J. Pister, and El, "Convex position estimation in wireless sensor networks," in *INFOCOM'01: Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2001, pp. 1655–1663.
- [6] T.-C. Chang, K. Wang, and Y.-L. Hsieh, "Enhanced color-theory-based dynamic localization in mobile wireless sensor networks," in *WCNC'07: Proceedings of the IEEE 2007 Wireless Communications and Networking Conference*, 2007, pp. 3064–3069.
- [7] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *INFOCOM'00: Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, 2000, pp. 775–784.
- [8] N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller, "The Cricket compass for context-aware mobile applications," in *MobiCom'01: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.
- [9] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Comput. Netw.*, vol. 43, no. 4, pp. 499–518, 2003.
- [10] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, October 2000.
- [11] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "Range-free localization and its impact on large scale sensor networks," *Trans. on Embedded Computing Sys.*, vol. 4, no. 4, pp. 877–906, 2005.
- [12] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed online localization in sensor networks using a moving target," in *IPSN '04: Proc. of the 3rd Int'l Symposium on Information Processing in Sensor Networks*. New York, NY, USA: ACM, 2004, pp. 61–70.
- [13] L. Hu and D. Evans, "Localization for mobile sensor networks," in *MobiCom '04: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM Press, 2004, pp. 45–57.
- [14] A. Baggio and K. Langendoen, "Monte Carlo localization for mobile wireless sensor network," *Ad Hoc Netw.*, vol. 6, no. 5, pp. 718–733, 2008.
- [15] S. Zhang, J. Cao, L. Chen, and D. Chen, "Locating nodes in mobile sensor networks more accurately and faster," in *SECON '08: Proc. of the 5th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2008, pp. 37–45.
- [16] M. Rudafshani and S. Datta, "Localization in wireless sensor networks," in *IPSN '07: Proc. of the 6th Int'l Conference on Information Processing in Sensor Networks*. New York, NY, USA: ACM, 2007, pp. 51–60.
- [17] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, June 2001.
- [18] W. Wang and Q. Zhu, "Sequential Monte Carlo localization in mobile sensor networks," *Wireless Networks*, vol. 15, pp. 481–495, May 2009.
- [19] M. Keir, C. Hann, J. Chase, and X. Chen, "A new approach to accelerometer-based head tracking for augmented reality & other applications," in *CASE'07: Proceedings of the IEEE International Conference on Automation Science and Engineering*, Sept. 2007, pp. 603–608.
- [20] M. Caruso, "Applications of magnetic sensors for low cost compass systems," *Position Location and Navigation Symposium, IEEE 2000*, pp. 177–184, 2000.
- [21] J. Yoon, M. Liu, and B. Noble, "Sound mobility models," in *MobiCom '03: Proceedings of the 9th Annual Int'l Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2003, pp. 205–216.
- [22] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Models and solutions for radio irregularity in wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 2, pp. 221–262, 2006.
- [23] M. Z. Zamalloa and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Trans. Sen. Netw.*, vol. 3, no. 2, p. 7, 2007.