Tri-MCL: Synergistic Localization for Mobile Ad-hoc and Wireless Sensor Networks

Arne Bochem, Yali Yuan and Dieter Hogrefe
Institute of Computer Science, Telematics Group
University of Goettingen, Germany
{bochem,yali.yuan,hogrefe}@informatik.uni-goettingen.de

***Abstract*—Localization is a highly important topic in wireless sensor networks as well as in many Internet of Things applications. Many current localization algorithms are based on the Sequential Monte Carlo Localization method (MCL), the accuracy of which is bounded by the radio range. High computational complexity in the sampling step is another issue of these approaches. We present Tri-MCL which significantly improves on the accuracy of the Monte Carlo Localization algorithm. To do this, we leverage three different distance measurement algorithms based on range-free approaches. Using these, we estimate the distances between unknown nodes and anchor nodes to perform more fine-grained filtering of the particles as well as for weighting the particles in the final estimation step of the algorithm. Simulation results illustrate that the proposed algorithm achieves better accuracy than the MCL and SA-MCL algorithms. Furthermore, it also exhibits high efficiency in the sampling step.**

***Index Terms*—Localization, Wireless Sensor Network, Internet of Things, Monte Carlo Localization, Range Free**

I. Introduction

Node localization plays an important role in Wireless Sensor Networks (WSNs) and Internet of Things (IoT) applications since it is not only useful in many basic network applications but also necessary in network operation. Examples are applications such as habitat monitoring [23], animal tracking [12], vehicle tracking [9], and environment monitoring [22], as well as network operation methods such as location-based routing protocols saving significant energy by eliminating the need for route discovery [13], [14], [17].

Global Positioning System (GPS) is the straightforward solution for sensor node localization; however, it has disadvantages such as high cost, high power use and no indoor operation. One reasonable solution is that only a small proportion of sensor nodes is equipped with a GPS module and the rest get their positions through another localization scheme. The sensor nodes equipped with a GPS are called seeds or anchors. Many localization algorithms have been proposed not only for static sensor networks [4], [19], [5], [15], but also for mobile sensor networks [6], [10], [1], [21] in the past several years.

A popular representative of localization algorithms for mobile sensor networks is Monte Carlo Localization (MCL) [10]. The key idea of MCL is that the positions of unknown nodes are determined by a set of weighted samples and each sample, usually called particle, represents a possible location of the node. The most important contribution of MCL is that it is especially designed for mobile WSNs, i.e. all nodes including anchors are allowed to move arbitrarily during network operation time. However, the sampling phase and filtering phase need to be repeated in order to obtain each particle, so it always suffers from high computational cost which will shorten the network life time significantly.

In this paper, we present the design and evaluation of a new algorithm for mobile sensor networks and IoT applications, called Tri-MCL. Tri-MCL follows the general MCL approach [10]. In order to improve the localization accuracy and sampling efficiency, our algorithm employs three different, synergistic distance measurements based on range-free methods and historical information to measure the distances between unknown nodes and anchor nodes. These distances are then used for filtering and weighting the particles in a more precise manner in the final estimation step of the algorithm. Tri-MCL is an interactive process operating over multiple distance estimation values to form a consolidated fusion by interactively exploiting the synergies in these range-free distance measurement approaches, which is the key difference from the traditional MCL approach. Our approach takes inspiration from Zhou et al. [31], who have presented a Tri-Training strategy for semi-supervised learning, but their work concentrated mainly on selecting unlabeled instances in an interactive voting manner for machine learning. However it should be noted that our approach itself does not employ (semi-supervised) machine learning, making it suitable for use on resource constrained devices such as wireless sensor nodes.

The structure of this paper is organized as follows: Section 2 reviews related works of existing MCL-based algorithms. In section 3, we describe our proposed scheme. Simulation results are shown in Section 4. Finally, we draw conclusions in section 5.

II. Related Works

Many localization algorithms have been designed for mobile sensor networks [28], [10], [7], [1], [11], [26], [20], [24], [16].

In 2004, the Monte Carlo method is firstly introduced by L. Hu and D. Evans for localizing nodes in wireless sensor networks, called MCL [10]. MCL-based localization represents the posterior distribution of a node's location by a set of weighted particles, and in each time unit, the particles are updated based on new observations about beacons from anchor nodes. The authors proposed the localization method for a general network environment where nodes and anchors can

DOI 10.1109/LCN.2016.61

move arbitrarily. It does not require any additional hardware. This makes the approach suitable for both mobile ad hoc sensor networks and IoT applications. The main drawback of the MCL algorithm is that it has to redo the sampling and filtering phases for each particle. Usually, it will iterate many times in order to obtain enough valid particles, which is really time consuming. This makes it less suitable for sensor networks with limited computational abilities. The values of the particle weights are only 1 or 0, making this part of the algorithm coarse-grained.

In [1], A. Baggio and K. Langendoen proposed the Monte Carlo localization Boxed (MCB) algorithm. The sampling area was generated by building boxes in the intersection of the anchor nodes' communication scope, which improves the time efficiency significantly in the prediction phase of MCL. However, when the particle number of MCB equals to that of MCL, the localization error will not be improved. Even worse is that the number of valid particles will increase with the increasing number of the anchor nodes. As a result, the set of valid particles will be much larger than necessary for estimating a node's location.

M. Rudafshani and S. Datta [20] proposed the MSL*, MSL to improve the filtering phase of MCL using the location information of unknown nodes within two hops, but the additional communication was needed to pass samples or accuracy information.

S. Hartung et al. [8] proposed the Sensor-Assisted Monte Carlo Localization (SA-MCL) method to solve the problem of temporary connection loss to anchor nodes due to changing network topologies. They used three different additional sensors to estimate the localization of unknown nodes. In [11], SMC was proposed to improve the localization accuracy by using the angle-of-arrival (AoA) measurements. Another range-based scheme [7] assumed that the distance or angle between anchor nodes and unknown node can be measured based on signal measurements such as received signal strength indication (RSSI), time of arrival (TOA), or angle of arrival (AOA). However, the authors in [8], [11], [7] all need additional hardware support to improve the accuracy or solve problems of MCL.

In [29], weighted MCL (WMCL) was proposed. WMCL can improve the localization accuracy and sampling efficiency with low anchor densities, but the communication cost is much higher than for the original MCL algorithm. The RDMCL method was proposed in [30], which is based on the Received Signal Strength (RSS), distance and direction of the moving anchor nodes and MCL. RDMCL used three methods based on the number of nodes' one-hop neighbor anchors to build a more effective sampling area. The authors in [25] proposed a Weighted Monte Carlo Localization based on the Smallest Enclosing Circle algorithm to solve the localization problem of node mobility in IoT scenarios. This algorithm generates the smallest enclosing circle of anchor nodes by using the hop counts from anchor nodes.

Symbol	Introduced	Meaning
V_{Max}	III-D	Maximum possible node speed, also defines radius around nodes
e_t	III-C3	HistDR position estimate at time t
L_t	III-D	Set of particles at time t
N	III-F	Number of particles in L_t
p_k	III-F	k-th particle
A	III-F	Set heard anchor nodes
a_i	III-F	i-th heard anchor node
ϕ	III-G1	Number of distance estimation methods
$r_{\text{RingWidth}}$	III-F	Filtering ring width parameter
$r_{\ldots,i}$	III-F	Distance estimate to anchor i according to method ...
$d(a,b)$	III-F	Distance between points or particles a and b
$p_{i,k}$	III-G	Copy of p_k for anchor a_i
σ	III-G1	$\sigma > 0$, used to avoid dividing by zero
α	III-G3	Weighting factor $\alpha = 0.75$
$\omega_{\text{Fail},i,k}$	III-G1	Penalty factor for $p_{i,k}$, relating to number of failed methods
$\omega_{\text{Range},i,k}$	III-G2	Penalty factor for $p_{i,k}$, relating how well the distance estimate matches
$\omega_{i,k}$	III-G3	Weighting factor for $p_{i,k}$
(x_t, y_t)	III-G4	Final position estimate at time t

TABLE I: Table of symbols.

III. Localization Scheme

In this section, our proposed localization scheme, Tri-MCL, is described in detail. Tri-MCL consists of three phases: initialization, sampling and filtering. But in the sampling phase, there is only a simplified re-sampling phase in Tri-MCL, which effectively reduces the computational cost. The Tri-MCL filtering phase is also different from that of traditional MCL-based algorithms. Firstly, instead of using only the radio range of anchors to do the filtering, we use ring areas with three different distances around anchors as the filter area to filter particles. This helps to improve the localization accuracy. Secondly, each particle has a different contribution to the final position estimate of the unknown node, as we weight each particle using a distance error penalty and a range free based distance estimation method failure penalty.

A. Notation

For reference purposes, all symbols used in the following description of our scheme will be listed in table I.

B. Initialization

Before having ran once, Tri-MCL is initialized with a set of N particles, distributed randomly over the experimental area.

C. Range-free distance estimation

Tri-MCL uses distance estimates between unknown nodes and anchor nodes to aid in the process of estimating locations. For this reason, three different schemes with different strengths are employed to estimate distances leading to the synergistic qualities of Tri-MCL. All of the different schemes are range-free and do not require additional hardware.

To allow penalizing the range estimates made with methods that are unsuitable for a given situation, each method will either return a result or a failure state.

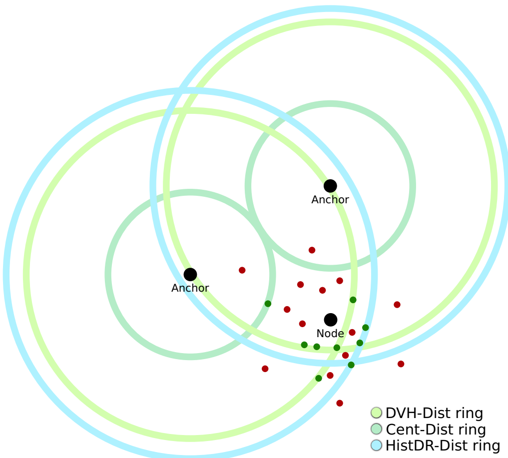

Fig. 1: Sample situation with two anchor nodes and one unknown nodes. Red are filtered particles and green are admissible particles.

1) DVH-Dist: DVH-Dist is a distance estimation algorithm built on the principles of the well known DV-Hop localization algorithm [18]. In DV-Hop first the minimum hop count from anchor nodes to unknown nodes are determined. Then the distance between the nodes and anchor nodes are computed by multiplying the minimum hop count and average distance of each hop. At last, the node estimates its position through triangulation algorithm or maximum likelihood estimators. However, since only the distances are relevant in the context of Tri-MCL, the costly calculations required to calculate the positions from the distance estimates are left out. We call this simplification of DV-Hop DVH-Dist.

Like DV-Hop, this approach works well in scenarios with a high density of unknown nodes, so that fine grained multi-hop distance estimates to a smaller number of anchor nodes can be made.

If no seed nodes have been heard, even indirectly, this method returns a failure state.

2) Cent-Dist: Cent-Dist works by calculating positions of nodes according to the Centroid localization scheme [2], where the center position between the received anchor node beacons is used to calculate a location estimate on unknown nodes. Calculating this position estimate only incurs little computational costs.

For Cent-Dist, this position estimate is calculated and used to determine estimated distances towards all known anchor nodes (i.e. also those received through DVH-Dist flooding). This approach works best, when there are multiple anchor nodes in the immediate vicinity of the unknown node attempting to calculate its position.

If less than one direct seed node has been heard, this method returns a failure state.

3) HistDR-Dist: HistDR-Dist, short for Historical Dead Reckoning Distance, is the final method employed in Tri-MCL to calculate range estimates.

For HistDR-Dist, the last three position estimates made by Tri-MCL are stored and used to derive an estimate of the current acceleration and angular acceleration of the node. Using these values, by means of dead reckoning, the current position of the node is estimated.

Let $e_{t-1} = (x_{t-1}, y_{t-1})$ be the previous estimate generated by Tri-MCL, $e_{t-2} = (x_{t-2}, y_{t-2})$ the one before that and so on. Using these values, HistDR will estimate the current position, which can then be used to determine a distance estimate $e_t = (x_t, y_t)$.

$$v_{t-1} = \mathrm{d}(e_{t-1}, e_{t-2}), v_{t-2} = \mathrm{d}(e_{t-2}, e_{t-3}) \tag{1}$$

$$\gamma_{t-1} = \angle(e_{t-1}, e_{t-2}), \gamma_{t-2} = \angle(e_{t-2}, e_{t-3}) \tag{2}$$

$$\Delta v = v_{t-1} - v_{t-2}, \Delta\gamma = \gamma_{t-1} - \gamma_{t-2} \tag{3}$$

$$x_t = x_{t-1} + (v_{t-1} + \Delta v)\cos(\gamma_{t-1} + \Delta\gamma) \tag{4}$$

$$y_t = y_{t-1} + (v_{t-1} + \Delta v)\sin(\gamma_{t-1} + \Delta\gamma) \tag{5}$$

This approach can give good results if the previous estimates are reasonably accurate. It is not reliant on other nodes for the current time step, so it bridges short intervals without connectivity to the rest of the network.

The performance of the approach depends on the mobility model. The performance of HistDR-Dist is optimized for more realistic models such as the random waypoint mobility model, rather than the random walk mobility model. HistDR-Dist can be further extended by going one derivation deeper and working with the differential of acceleration and angular acceleration. Such a modification should allow the model to perform better in a simulation using a Gaussian mobility model as well as in a real world implementation.[3]

Since the computation is simple and only depends on a fixed number of components, the computational cost of HistDR-Dist is very low. Taken together, DVH-Dist, Cent-Dist and HistDR-Dist can be assumed to have less than the computational cost of DV-Hop alone.

If less than three samples exist in the history, this method returns a failure state.

D. Prediction

A parameter V_{Max} given in $\mathrm{m/s}$ defines the maximum speed any node in the network can attain. For the prediction step, similar to the original MCL, the set of particles L_t at the current time t is determined by iterating over the set of previous particles L_{t-1} and for each particle $l_i \in L_{t-1}$, a new particle is drawn from its surroundings within a radius of V_{Max}, reinterpreted in meters, around it. If the prediction is not performed once per second, the radius has to be adjusted correspondingly, both in MCL and in Tri-MCL, e.g. for a $0.5\,\mathrm{s}$ interval, the radius in which particles may move should be halved.

E. Filtering in MCL

In the original MCL, filtering is done by discarding particles that do not lie within one radio range r_{Range} around any of the directly heard anchor nodes and within the ring from one to two radio ranges around indirectly heard (2-hop) anchor nodes.

This means that particles $p \in L_t$ are kept, when the following condition holds:

Let MCL_A be the set of directly heard anchor nodes in MCL and let MCL_I be the set of 2-hop anchors.

$$\exists a \in \text{MCL}_A : \text{d}(p, a) \le r_{\text{Range}} \tag{6}$$
$$\wedge \exists a \in \text{MCL}_I : r_{\text{Range}} < \text{d}(p, a) \le 2 * r_{\text{Range}} \tag{7}$$

Regarding the resampling efficiency, in the original MCL, filtering is implemented in such a way, that a particle is drawn from L_{t-1}. Then the prediction step is run on this particle, and finally the decision is made whether to keep the particle or not. This process is repeated until the new set of particles L_t is full. This can lead to a high number of iterations of the costly prediction step in order to get enough admissible particles. In Tri-MCL, we simply run the prediction step once for each particle in L_{t-1} and once per missing particle.

F. Filtering in Tri-MCL

For Tri-MCL, we eschew the first part and extend the second part, keeping only particles $p_k \in L_t$, for which the following condition holds, with A being the set of anchor nodes heard over any number of hops, $a_i \in A$ being the ith anchor node and $r_{\{DVHDist, CentDist, HistDRDist\},i} \in RF_i$ being the corresponding distance estimate according to the three different distance estimation methods:

$$i \in \{1, ..., |A|\} \tag{8}$$
$$k \in \{1, ..., N\} \tag{9}$$
$$\exists a_i \in A, \exists r \in RF_i : \tag{10}$$
$$r - r_{\text{RingWidth}} \le \text{d}(p_k, a_i) \le r + r_{\text{RingWidth}} \tag{11}$$

Where $\text{d}(p_k, a_i)$ refers to the euclidean distance between the position of the particle's and anchor node's position and $r_{\text{RingWidth}}$ is one of the parameters of the algorithm, referring to the tolerance with which particles are kept, even if their range does not exactly match that of any range estimates.

After filtering, the set L_t may contain less than N particles, which can be remedied depending on the situation:

1) If it contains no particles at all, it will be reinitialized with the positions of all directly heard anchor nodes as particles.
2) If no directly heard anchor nodes are available, the positions of indirectly (2-hop) anchor nodes are used to seed the set of particles.
3) If still no particles are in the set, it is reset to its state before filtering took place.

If at this point $|L_t| < N$, until $|L_t| = N$, a particle $p_k \in L_t$ is drawn and from it a particle p'_k is sampled from its V_{Max} surroundings, as in the prediction step, and then inserted into L_t:

$$d(p'_k, p_k) \le V_{\text{Max}} \tag{12}$$
$$L_t := L_t \cup \{p'_k\} \tag{13}$$

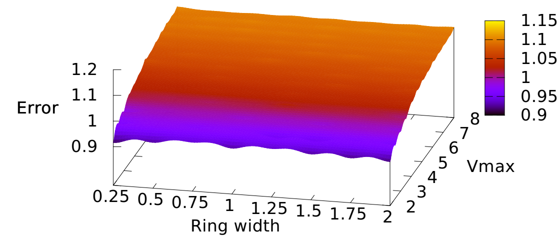

Fig. 2: Tri-MCL error relative to radio range over varying ring width and V_{Max}.

Figure 1 gives a graphical representation of our approach. The shown scenario consists of one unknown node and two anchor nodes. The red and green points represent the particles of the unknown node. Red particles are filtered and green particles are kept. Each of the six colored rings, one per method and anchor node, is one of the admissible areas for particles and corresponds to the three different types of distance measurement methods we employ. The green particles lie only on the colored rings.

G. Position estimation

During a final step, the position is estimated as a weighted average of the particles. Each particle is used once per anchor node that is heard by the unknown node and each such instance of a particle is weighted according to two penalty-factors:

$$\forall a_i \in A, \forall p_k \in L_t, \text{ let } p_{i,k} := p_k \tag{14}$$

1) Distances estimation method failure penalty: The distance estimation method failure penalty factor $\omega_{\text{Fail},i,k}$ relates to the number ϕ of distance estimation methods that succeeded in estimating a distance to an anchor node associated with certain particles $p_{i,k}$:

$$\epsilon_{i,k} = \frac{|RF_i| - \phi}{|RF_i|} \tag{15}$$
$$\beta_{i,k} = \frac{\epsilon_{i,k} + \sigma}{1 - \epsilon_{i,k} + \sigma} \tag{16}$$
$$\omega_{\text{Fail},i,k} = \frac{1}{\beta_{i,k}} \tag{17}$$

With a small $\sigma > 0$, used to avoid dividing by zero.

2) Range error penalty: The range error penalty factor $\omega_{\text{Range},i,k}$ represents how well the particle's position matches the estimated ranges. It is computed as the average distance error over all three of the range free distance estimation methods for the given particle $p_{i,k}$, as follows:

$$\xi_{i,k}(r) = \begin{cases} 1, & \text{if } r \text{ failed} \\ \frac{|\text{d}(p_{i,k}, a_i) - r|}{\text{maxDistance}}, & \text{otherwise} \end{cases} \tag{18}$$
$$\omega_{\text{Range},i,k} = \frac{1}{3} * \sum^{r \in RF_i} \xi_{i,k}(r) \tag{19}$$

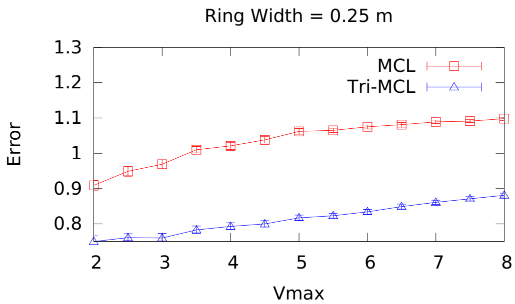

Fig. 3: Tri-MCL error compared to MCL error over varying V_{Max} with a ring width of $0.25\,\text{m}$.

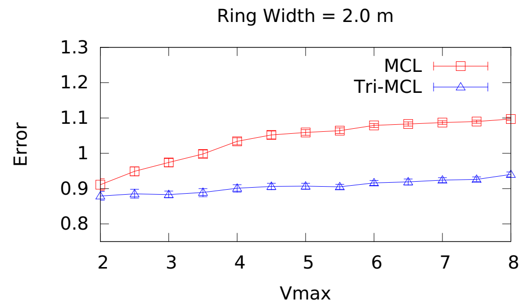

Fig. 4: Tri-MCL error compared to MCL error over varying V_{Max} with a ring width of $2\,\text{m}$.

3) Overall weight: Finally, the two weights are combined to form the final weight of the particle $p_{i,k}$:

$$\omega_{i,k} = \omega_{\text{Range},i,k} * \alpha + \omega_{\text{Fail},i,k} * (1 - \alpha) \tag{20}$$

Here, α is a weighting factor, chosen after some informal trial runs as 0.75.

4) Weighted average: In the last step, the position estimate is calculated as a weighted average over all the particles left in the set.

With $p^x_{i,k}$ referring to the x component of the particle $p_{i,k}$ and $p^y_{i,k}$ referring to the y component, the final coordinates are calculated as such:

$$\omega_\Sigma = \sum_{i=0}^{|A|} \sum_{k=0}^{N} \omega_{i,k} \tag{21}$$

$$x_t = \frac{1}{\omega_\Sigma} \sum_{i=0}^{|A|} \sum_{k=0}^{N} \omega_{i,k} p^x_{i,k} \tag{22}$$

$$y_t = \frac{1}{\omega_\Sigma} \sum_{i=0}^{|A|} \sum_{k=0}^{N} \omega_{i,k} p^y_{i,k} \tag{23}$$

IV. Evaluation

To evaluate the effectiveness of our new approach, we have performed a set of simulations in an especially built simulation software while varying two simulation parameters.

The simulations are run with 150 nodes, of which 15 are anchor nodes, distributed randomly over a $100\,\text{m} \times 50\,\text{m}$ simulation area. To balance out bias introduced by the random distribution, we run each simulation of $300\,\text{s}$ simulation time fifty times. The radio communication range of all nodes is set $20\,\text{m}$.

Mobility is introduced into the simulation by having nodes move around the simulation area on the basis of a modified random waypoint model. To avoid a loss of velocity as described by Yoon et al., the model constrains the admissible combinations of newly picked speed and waypoint combinations in such a way, that the picked combination must be reachable within five simulation seconds. Otherwise a new speed and waypoint combination is chosen until an admissible combination is found.[27]

Over the simulation runs, we vary the maximum speed V_{Max} attainable by nodes and the Tri-MCL parameter $r_{\text{RingWidth}}$. V_{Max} was varied within $[2, 8]$ with a step size of 0.5, while $r_{\text{RingWidth}}$ was varied within $[0.25, 2]$ with a step size of 0.25.

For the sake of comparison, we evaluated the results of MCL[10] and SA-MCL[8] at the same time as Tri-MCL.

A. Results

The error values for Tri-MCL over our simulation runs are shown in Figure 2. The error is given relative to the communication range.

For figures 3 and 4, $99\,\%$ confidence intervals are given for each sample.

We found Tri-MCL delivers the highest improvement upon MCL at high values for V_{Max} and low values for $r_{\text{RingWidth}}$, with a maximum improvement of $28\,\%$ during one simulation run. The highest, average improvement at $25\,\%$ over a simulation batch was found with $r_{\text{RingWidth}} = 0.25\,\text{m}$ and $V_{\text{Max}} = 5\,\text{m/s}$. In the worst batch there is still some slight but significant improvement over MCL of $3.5\,\%$ at maximum tested $r_{\text{RingWidth}}$ and minimum tested V_{Max}.

Overall it can be seen that lower values of $r_{\text{RingWidth}}$ lead to better location estimates due to higher precision during the filtering step.

It should be noted that the best accuracy is achieved with both low values for V_{Max} and low values for $r_{\text{RingWidth}}$. However, Tri-MCL appears to be more robust than MCL against higher speeds, with its performance not detoriating as quickly, which is why the improvement over MCL is higher with higher values for V_{Max}.

Figure 4 shows that even with a higher $r_{\text{RingWidth}}$ value of $2\,\text{m}$, Tri-MCL performs significantly better than MCL.

SA-MCL is apparently not well suited for the given scenario, giving results nearly identical to MCL, which is why it has been omitted from figures 3 and 4 to enhance readability.

V. Conclusion and Future Works

In this paper, we present a new localization scheme called Tri-MCL, which improves localization accuracy and increases the efficiency of sampling during the prediction step. Our method employs three different distance measurement approaches based on range-free methods to estimate distances between unknown nodes and anchor nodes. These distances are then used to filter out particles not lying within rings around the anchor nodes with a radius corresponding the distance estimates. We also consider the weight of different particles, which means that the weight of each particle is related to the distance between anchor node and unknown node. The results from our simulations and experiments validate the effectiveness of our proposed algorithms in improving localization accuracy and reducing computational costs during resampling. In the future, we aim to implement our proposed algorithm in mobile ad hoc networks with real world deployments.

ACKNOWLEDGMENT

We would like to note that the contributions of both Arne Bochem and Yali Yuan for this paper are equal. Therefore, Yali Yuan is contributing as the co-first author. Yali Yuan would like to thank for the scholarship support from the China Scholarship Council (CSC).

REFERENCES

[1] A. Baggio and K. Langendoen, "Monte-carlo localization for mobile wireless sensor networks," in *Mobile Ad-hoc and Sensor Networks*. Springer, 2006, pp. 317–328.

[2] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *Personal Communications, IEEE*, vol. 7, no. 5, pp. 28–34, 2000.

[3] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[4] H. Chen, G. Wang, Z. Wang, H.-C. So, and H. V. Poor, "Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 1, pp. 108–116, 2012.

[5] Y.-S. Chen, S.-H. Chang, and C.-C. Teng, "Location estimation based on convex overlapping communication regions in wireless ad hoc sensor networks," in *Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014 4th International Conference on*. IEEE, 2014, pp. 1–5.

[6] H. Chenji and R. Stoleru, "Toward accurate mobile sensor network localization in noisy environments," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 6, pp. 1094–1106, 2013.

[7] B. Dil, S. Dulman, and P. Havinga, "Range-based localization in mobile sensor networks," in *Wireless Sensor Networks*. Springer, 2006, pp. 164–179.

[8] S. Hartung, A. Bochem, A. Zdziarstek, and D. Hogrefe, "Applied sensor-assisted monte carlo localization for mobile wireless sensor networks," in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing, 2016, pp. 181–192.

[9] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*. ACM, 2004, pp. 270–283.

[10] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 45–57.

[11] R. Huang and G. V. Zaruba, "Incorporating data from multiple sensors for localizing nodes in mobile ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 9, pp. 1090–1104, 2007.

[12] 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," *ACM Sigplan Notices*, vol. 37, no. 10, pp. 96–107, 2002.

[13] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 243–254.

[14] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," *Wireless networks*, vol. 6, no. 4, pp. 307–321, 2000.

[15] S. Kumar and D. Lobiyal, "An advanced dv-hop localization algorithm for wireless sensor networks," *Wireless personal communications*, vol. 71, no. 2, pp. 1365–1385, 2013.

[16] T.-H. Lin, P. Huang, H.-H. Chu, and C.-W. You, "Energy-efficient boundary detection for rf-based localization systems," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 1, pp. 29–40, 2009.

[17] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *Network, IEEE*, vol. 15, no. 6, pp. 30–39, 2001.

[18] D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, vol. 5. IEEE, 2001, pp. 2926–2931.

[19] ——, "Ad hoc positioning system (aps) using aoa," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3. Ieee, 2003, pp. 1734–1743.

[20] M. Rudafshani and S. Datta, "Localization in wireless sensor networks," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*. IEEE, 2007, pp. 51–60.

[21] J.-P. Sheu, W.-K. Hu, and J.-C. Lin, "Distributed localization scheme for mobile sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 4, pp. 516–526, 2010.

[22] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. LaHusen, "Air-dropped sensor network for real-time high-fidelity volcano monitoring," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 2009, pp. 305–318.

[23] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 34–40, 2004.

[24] S. Tilak, V. Kolar, N. B. Abu-Ghazaleh, and K.-D. Kang, "Dynamic localization control for mobile sensor networks," in *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*. IEEE, 2005, pp. 587–592.

[25] J. Xu, F. Bu, W. Si, Y. Qiu, and Z. Chen, "An algorithm of weighted monte carlo localization based on smallest enclosing circle," in *Internet of things (iThings/CPSCom), 2011 international conference on and 4th international conference on cyber, physical and social computing*. IEEE, 2011, pp. 157–161.

[26] J. Yi, S. Yang, and H. Cha, "Multi-hop-based monte carlo localization for mobile sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on*. IEEE, 2007, pp. 162–171.

[27] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, March 2003, pp. 1312–1321 vol.2.

[28] S. Zhang, J. Cao, L. Chen, and D. Chen, "Locating nodes in mobile sensor networks more accurately and faster," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on*. IEEE, 2008, pp. 37–45.

[29] S. Zhang, J. Cao, C. Li-Jun, and D. Chen, "Accurate and energy-efficient range-free localization for mobile sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 6, pp. 897–910, 2010.

[30] Y. Zhang, L. Cui, and S. Chai, "Energy-efficient localization for mobile sensor networks based on rss and historical information," in *Control and Decision Conference (CCDC), 2015 27th Chinese*. IEEE, 2015, pp. 5246–5251.

[31] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 11, pp. 1529–1541, 2005.