CARI'12

Speed and Direction Prediction-based Localization for Mobile Wireless Sensor Networks

Imane Benkhelifa[*,**] — Samira Moussaoui[*]

* Departement of informatics
USTHB- Faculty of Electronics and Informatics
EL-Alia 16111, ALGIERS
ALGERIA

** CERIST Research Center
DTISI Division
Ben-Aknoun, ALGIERS
ALGERIA

i.benkhelifa@cerist.dz, moussaoui_samira@yahoo.fr

RÉSUMÉ. Dans cet article, nous proposons une technique de localisation efficace et pratique spécialement conçue pour les réseaux de capteurs mobiles. Cette méthode utilise une seule ancre mobile traversant la zone de déploiement avec une trajectoire prédéfinie tout en diffusant sa position à ses capteurs voisins. En utilisant les informations de positionnement précédentes, le nœud prédit sa vitesse et sa direction de déplacement. L'évaluation de notre solution montre que cette technique de prédiction bénéficie à la fois, de la prédiction et la trajectoire de l'ancre mobile, d'une manière plus efficace que des solutions précédentes. Les résultats de simulation montrent que notre algorithme dépasse la méthode de Monté Carlo conventionnel et sa variante MCB.

ABSTRACT. In this paper, we propose an efficient and practical localization technique especially designed for mobile sensor networks. It uses a single mobile anchor travelling with a predefined trajectory while periodically broadcasts its current location coordinates to the nearby sensors. Using previous location information, the node predicts its speed and direction. The evaluation of our solution shows that our technique takes benefit from both, prediction and anchor trajectory, in a more effective manner than previous solutions. The simulation results show that our algorithm outperforms conventional Monte Carlo localization schemes and its variant MCB.

MOTS-CLÉS : Réseaux de capteurs mobiles, Localisation, Prédiction, Ancre Mobile.
KEYWORDS: Mobile Sensor Networks, Localization, Prediction, Mobile anchor.

1. Introduction

Several methods have been proposed to allow sensors having their location estimations. Except probabilistic methods like Monte Carlo and its variants [1][2][3], all the other methods propose the use of methods dedicated to static sensor networks and then frequently refresh the estimations after a certain time [4]. As for probabilistic solutions, they turn out to be memory consumers when saving, holding and renewing location samples and they consider all - in their algorithms- maximum values more particularly, the maximum speed of the nodes [3]. These techniques need also an important number of anchors. An anchor is a node aware of its position (e.g., equipped with GPS). However, having a GPS receiver on every anchor is currently a costly proposition, in terms of power, volume and money. For this reason, other techniques propose the use of a single mobile anchor in order to minimize material coasts [5][6]. The mobile anchor may be a robot, a drone, a plane or a vehicle. In such methods, the anchor travels through the deployment area while broadcasting its location along the way. Sensors localize themselves by monitoring information coming from this anchor [6]. Thus, in sensor networks that already incorporate mobile anchors as part of the design, enabling localization through the mobile anchor can be a cost-effective way of achieving sensor network localization.

In some applications such animal tracking, nodes move with unknown velocities and random directions. A prediction of speeds and directions of nodes is proving to be a promising solution to move the estimated positions closer to the real positions.

In this paper, we propose a new localization method for mobile sensor networks called “Speed and Direction Prediction-based Localization” (SDPL) based on the prediction of node speeds and directions considering its previous estimations and the anchor information. This provides more positioning accuracy compared with other methods that do not consider any direction information and use only the maximum speed to predict positions.

2. Related work

In [1], Hu and Evans present a range-free anchor-based localization algorithm for mobile sensor networks based on the Sequential Monte Carlo method. The Monte Carlo method has been extensively used in robotics where a robot estimates its position based on its motion, perception and possibly a pre-learned map of its environment. The authors extend the Monte Carlo method (MCL) as used in robotics to support the localization of sensors in unmapped terrain. They assume that a sensor has little control and knowledge over its movement, in contrast to a robot. The only assumption that is made is that the sensors or anchors move with a known maximum speed and that the

radio range is common to the sensors and the anchors – or is distributed together with other messages. This latter point, however, is not described by the authors.

An improvement of MCL can be found in [3]. This new version called Monte Carlo localization Boxed (MCB) uses steps similar to those of MCL. The major differences lie in the way anchor information and the method for drawing new samples. The method used for constraining the area from which MCB draws samples is as follows. A node that has heard anchors – one-hop or two-hop anchors – builds a box that covers the region where the anchors' radio ranges overlap. This box is called *"anchor box"*. Once the anchor box is built, a node simply has to draw samples within the region it covers. In the case where samples already exist, a bounding box is built with an additional constraint, namely, for each old sample from the sample set, an additional square of size 2 * *vmax* centered at the old sample is built, called *"sample box"* where *vmax* is the maximum speed of all the nodes. This updated box delimits per old sample the area a node can move in one time interval at maximum. MCB is supposed to support mobility. However, in the evaluation, authors simulate two time units. In the first time units, the nodes move without localizing. For each subsequent time unit, the nodes first localize and then move. In other words, the time freezes and the whole network is localized using a snapshot. There is no movement while the nodes are localizing.

3. Speed and direction prediction-based localization

In this section, we describe the proposed localization algorithm for mobile sensor networks. First, we believe mobility should be taken into account directly when designing new localization algorithms.

3.1. Localization technique

Only the anchor can obtain its exact coordinates at any time (eg. equipped with GPS). At the beginning, nodes have no knowledge about their positions. The anchor broadcasts periodically location packets while travelling through the deployment area. The localization process is repeated periodically. At each period of location invocation ΔP, a node can be found in one of the three cases:

- Case 1: The node has never received any anchor message. In this case, the node draws *N* random samples from the whole deployment area and takes the mean as its estimated position.
- Case 2: The node has received anchor messages:

To better understanding, we divide the set of the received anchor positions in Δt into k sub-sets E_i where each E_i contains only interconnected circles formed by these positions. E_k is the last sub-set.

1) If there is no E_i before E_k that contains three or more circles, then, the location estimation is, according to E_k, as follows:

a) If E_k has one element, the estimated position is the mean the *N* samples drawn from the circle centered at the anchor position and has as radius the distance between the anchor and the node (this distance is converted from the RSSI of the received message).

b) If E_k has two elements, the node takes as estimation the gravity center of the intersection zone of the two circles formed by the two heard positions. In reality, the node position is one of the intersection points, but a node cannot decide in which side it belongs without third information.

c) If E_k has three elements or more, the node treats the formed circles three by three. The shortest distance between the four intersection points determines two successive node positions. The node considers the last point as its estimated position (see Figure 1).

We have taken the shortest distance to determine node positions because, as supposed, the anchor speed is greater than the node speed. This case is verified by simulation where in 91% of the cases, this technique has given successful results.

The case of receiving three messages or more is very important because it allows unknown nodes to calculate an estimation of their velocities and directions for future utilizations, especially, when they cannot receive anchor messages.

2) If there is a subset E_i ($i < k$) which contains three or more elements, then, the estimation is as follows:

The node estimates the positions in E_i similarly to the case 1).c, then, it draws a line T_i through these positions with the *linear regression*.

a) If T_i goes across all the circles of E_k, then, the node concludes that it has not changed its direction. If $|E_k| \leq 2$, then, the estimated position will be predicted from T_i. Else, the node executes 1).c for E_k and uses positions to refine the line of the previous linear regression.

b) If there is no connection between T_i and E_k, the node deduces that its direction has been changed. The node calculates its position only according to E_k.

- Case 3: The node has not received anchor messages in ΔP: if the node has already an estimation of its velocity and its direction, then the node uses them to predict its new position. Else, the node keeps its last estimation.

3.2. Speed and direction prediction

SDPL is mainly based on the prediction of the velocity and the direction of unknown nodes. To do so, we suppose that nodes follow a rectilinear movement (see Figure 2) where nodes have a constant velocity and direction during certain time periods (Δt).

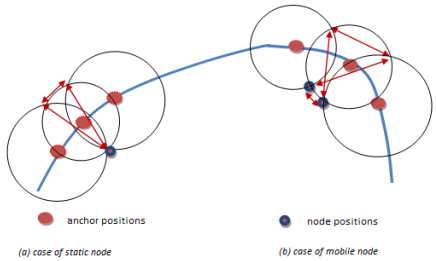

Figure 1: *Estimation with three circles*

This reflects the reality where nodes (e.g. Human beings, cars…etc) keep their speed and direction, at least, for moments. This allows nodes to predict positions at time $T = T_0 + \Delta T$ with the following equations:

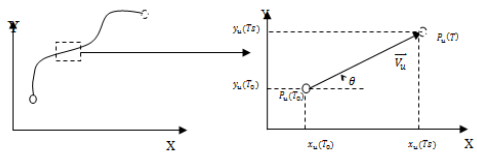

Figure 2: *Prediction of speed and direction*

$$P_u(T) = \overrightarrow{V_u} \times \Delta T + P_u(T_0) \rightarrow \begin{cases} x_u = V_u \times cos\theta \times \Delta T + x_u(T_0) \\ y_u = V_u \times sin\theta \times \Delta T + y_u(T_0) \end{cases} \quad (1)$$

Where $\theta$is the angle between the abscissa axe and the speed vector $\vec{V}$.
The speed is calculated as follows:

$$V_u = \frac{\sqrt{\left(x_u(T_0) - x_u(T_s)\right)^2 + \left(y_u(T_0) - y_u(T_s)\right)^2}}{T_s - T_0} \quad (2)$$

Where T_0 and T_S are times corresponding to two positions already estimated. If the calculated speed is equal to zero, the node deduces that it is static since Δt.
The angle θ defines the node direction.$\theta$is calculated as follows:

$$\tan\theta = \frac{y_u(T_0) - y_u(T_s)}{x_u(T_0) - x_u(T_s)} \quad (3)$$

In case where a node has already estimated many positions, it calculates the speed of each couple positions and takes the mean as its next speed. In addition, to predict the

direction angle, the node makes a linear regression with these positions to deduce, then, the line that provides a best fit for the data points using the *least squares* approach.

After the prediction of the speed and the direction, a node estimates its coordinates (x,y) as follows:

$$\begin{cases} x = x_{prev} + \cos\theta \times speed \times T_{dif} \\ y = y_{prev} + \sin\theta \times speed \times T_{dif} \end{cases} \quad (4)$$

Where (x_{prev}, y_{prev}) is the last estimated position; $(speed, \theta)$ are respectively the predicted speed and direction angle; $\boldsymbol{T_{dif}}$ is the time between current time and time of the last estimation.

4. Evaluation

To evaluate our proposed algorithm, we used the Network Simulator (NS) [7] version ns-allinone-2.34. NS2 is widely used in academic network researches.

To analyze the simulation results, the main metric is the localization error. As done in [1][2][3][5], the localization error is the distance between the real and the estimated position of a node. We consider the average localization error over all sensors. We compare the mean error of our method with one gives by MCB since it outperforms MCL [3]. The following table summarizes the simulation parameters

Mobility Model	Random Waypoint
Size of the deployment area	200 x 200 m²
Number of Nodes	100
Transmission Range	30 m
Anchor Trajectory	Waves [8]
Anchor Speed	20 m/s
Positioning invocation interval	5 s
Size of the sample set	50
Simulation runs	50

Table 1: *Simulation Parameters*

We note that the localization error does not depend on the node density; since nodes receive beacons directly from the mobile anchor. The mean error is calculated when the anchor finishes its trajectory one time. Obviously, when the anchor follows its trajectory many times, the mean error decreases because nodes have the chance to receive more beacons from the anchor whenever it passes close to them.

4.1. Variation of nodes' speed

For the first test, we set the broadcasting interval of the mobile anchor to 1s. Figure 3(a) shows that, for both methods, when the maximum speed increases, the mean error increases too. Indeed, the high speed (up to 54 km/h in the graph) allows the node to move quickly away from the anchor. This deprives the node of receiving anchor messages that help it localizing. On the other hand, when the speed is small (the smallest speed in the graph is 3.6 km/h), the node moves slowly and the anchor messages remain valid for certain time. This allows the node to minimize its localization error. In addition, in the case where the node cannot receive anchor messages, SDPL tries to predict the speed and the direction of the node movement. This enables the node to minimize the estimation error and thus outperforming MCB.

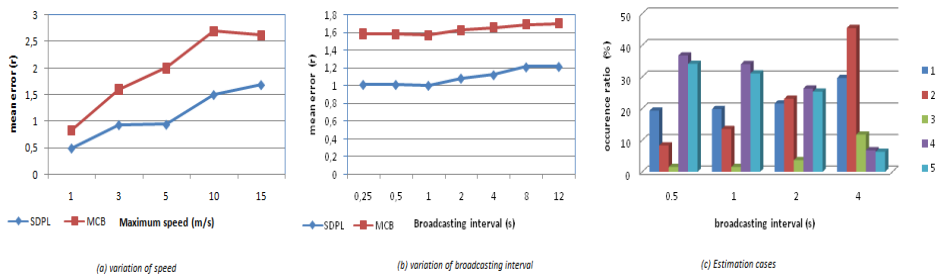

Figure 4: *Evaluation of SDPL*

where :

1- Estimation from the whole deployment area.
2- Estimation from one anchor circle.
3- Estimation from the intersection of two circles.
4- Estimation from the intersection of three circles.
5- Estimation using prediction of speeds and directions

4.2. Variation of the broadcasting interval

Now, we set the maximum speed to 5m/s and we study the impact of the broadcasting interval on the localization error. The broadcasting interval is the time between two consecutive messages sent by the anchor. Varying this interval has a direct influence on the number of the diffused messages.

In Figure 3(b), we notice that when decreasing the anchor broadcasting interval, the localization error decreases. In fact, if this interval is small, this means that the anchor diffuses more location messages in each positioning invocation period. Thus, nodes receive more messages that help them to determine their positions. In the case of MCB, the high number of messages enables building smaller anchor boxes and sample boxes

which allows drawing samples closer to the real position. As for SDPL, the high number of messages enables getting more speed and direction-prediction cases. This makes estimated positions closer to the real positions. Figure 3(c) presents in details the occurrence ratio of each occurrence case. When the broadcasting interval is small, the probability of being in case n° 4 and n° 5 is high. This explains the decrease of the localization error in Figure 3(b) for SDPL method.

5. Conclusion

In this paper, we proposed a new method especially designed for mobile sensor networks called Speed and Direction Prediction -based Localization which is mainly based on the prediction of the speed and the direction that a node moves with. To help nodes getting their positions, one single mobile anchor is deployed instead of many static anchors. This allows reducing material costs. The results of simulations of our algorithm show that it allows a node to get an improved accuracy more than 56% over Monte Carlo localization Boxed which is an improvement of Monte Carlo localization. Most importantly, it ensures node not being able to receive information from the anchor to be localized thanks to an efficient prediction.

6. Bibliography

[1] Evans, D. and Hu, L.2004. Localization for Mobile Sensor Networks. in Proc. of *ACM MOBICOM*, pp. 45-57.

[2] Huber, M. , Karmangar, F.A and Zaruba, G.V. 2004. Monte Carlo Sampling Based In-Home Location Tracking with Minimal RF Infrastructure Requirements. in Proc. of *IEEE GLOBECOM*, vol. 6, pp. 3624-3629.

[3] Baggio A., Langendoen,K..2008. Monte Carlo Localization for Mobile Wireless Sensor Networks. In *ELSEVIER Ad Hoc Networks 6*, pp. 718-733.

[4] Abu-Ghazaleh, N.B., Kang, K.D., Kolar, V., Tilak, S. 2005. Dynamic localization control for mobile sensor networks. In: Proceedings of *the IEEE International Workshop on Strategies for Energy Efficiency in Ad Hoc and Sensor Networks*.

[5] Das, S. M., Hu, Y. C., Koutsonikolas, D. 2006. Path Planning of Mobile Landmarks for Localization in Wireless Sensor Networks. in Proc. of *IEEE Distributed Computing Systems Workshops*, pp. 86-86.

[6] Ramadurai, V., Sichitiu, M. L. 2004. Localization of Wireless Sensor Networks with a Mobile Beacon. in Proc. of *the First IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pp. 174-183.

[7] NS-2 Simulator, http://www.isi.edu/nsnam/ns/

[8] Benkhelifa,I., Moussaoui, S. 2011. APPL: Anchor Path Planning based Localization for Wireless Sensor Networks. In Proc. *MIC-CCA*, Istanbul, Turkey.