International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 16 (2014), pp. 1701-1710
© International Research Publications House
http://www. irphouse.com

Model for Accurate Sensor Node Localization Using MCB with Speed & Direction Calculation

Vinod Kumar[1], Amit Kumar[2], Satyendra Yadav[3]

[1]Delhi Technological University, Delhi
[2, 3]Research Scholar, Delhi Technological University, Delhi

Abstract

Finding the geographical location of the sensors is the most common problem associated with the Wireless sensor networks. The Sequential Monte Carlo Localization (SMCL) algorithm is the base for most of the localization algorithms proposed. These localization algorithms requires high seed node density and they also suffers from low sampling efficiency. There are some papers which solves this problems but they are not energy efficient. Another approach The Monte Carlo Localization Boxed (MCB) method was used to reduce the scope of searching the candidate samples and thus reduces the time for finding the set of valid samples. In this paper we have proposed an energy efficient approach which will consider the direction of movement and the speed of acceleration of the sensor nodes. This additional information will help in further reducing the scope of searching the candidate samples. The valid samples will have more accurate location information and they are less likely to be filtered in the filtering step of SMCL. Thus it reduces the number of iterations the algorithm needs and hence achieves high localization accuracy in less time as compared to traditional SMCL and MCB.

Keywords: Localization; Wireless Sensor Networks; Sequential Monte Carlo Localization (SMCL); Monte Carlo Localization Boxed (MCB);

Introduction

Wireless Sensor Networks (WSNs) are composed of large number of sensors that are equipped with a processor, wireless communication capabilities, sensor capabilities, memory and a power source (Battery). WSNs have been used in many fields including environmental monitoring and habitat monitoring, precision agriculture, animal tracking and disaster rescue. In many applications, it is essential for nodes to know their position. In the most existing sensor networks, sensors are static but some modern applications have sensors that are mobile. For example in habitat monitoring applications like Zebra Net [5] sensors are attached to zebras and collect information about their behavior and migration patterns [6]. In other applications sensors are deployed on cellular phones to measure reception quality [6].

The fundamental problem in designing sensor network is localization- determining the location of sensors. Traditional method for obtaining the node's location information include attaching a GPS receiver in each node or manually configure each node's position. As the scale of sensor networks becomes larger and larger, these methods are becoming unfeasible for their high cost and inconvenience. Many localization algorithms for sensor networks have been proposed [8], [7], [12], [15], [16], [13], [14], [17], [10], [11], [18], [9]. These algorithms use some special nodes, called anchor or seed nodes, which know their positions to facilitate the determination of the positions of the other nodes (called common nodes). However these algorithms are designed for static sensor networks and are not applicable to mobile sensor networks. Most of these algorithms also require special costly hardware as they depend upon measuring ranging information from signal strength, time of arrival, time difference of arrival or angle of arrival. Adding the required hardware increases the cost and size of the nodes.

We are interested in performing localization in a more general network environment where the prior deployment of the seed node is unknown, node distribution is irregular, the seed density is low and where seeds and nodes can move uncontrollably. Although mobility makes other localization techniques increasingly less accurate, our technique takes advantage of mobility to improve accuracy and reduce the number of seeds required.

We consider a sensor network composed of seeds that know their locations and nodes with unknown locations. We are interested in following three scenarios:

***Nodes are static, seeds are moving*:**
For example, a military application where nodes are dropped from plane onto land and transmitters attached to soldiers in the area are used as moving seeds. Each node receives information from seeds and estimates its location more accurately.

***Nodes are moving, seeds are static*:**
For example, nodes are moving along the river and seeds are placed at fixed locations on the river banks. In this scenario the nodes location will change as the time passes, old location will become inaccurate since the node has moved. So the seed information is required to revise the location estimate.

***Both nodes and seeds are moving*:**
This scenario is most general in nature. It is applicable to any application where the nodes and seeds are deployed in an ad hoc way.

Some localization algorithms specially designed for mobile sensor networks have also been proposed [1], [19], [2], [20], [4]. They all use the Sequential Monte Carlo (SMC) method. In mobile sensor networks the SMC methods are preferred as they are easy to implement and can exploit nodes mobility to improve localization accuracy. But the SMC methods need to keep sampling and filtering until obtaining enough valid samples. This is very time consuming and not suitable where nodes have limited computation capability. In this paper we will use Bounding Box method which will reduce the scope of searching candidate sample. We will further improve the location accuracy by adding the information about the direction of movement of the node with the help of a compass attached with each node. Now using this information we can predict the next position of the valid samples more accurately. Hence we can calculate the location of the nodes more accurately.

Background

Network Model:

We have 2 kinds of nodes, one is seed node who knows their exact position at any time and second is common nodes who needs to determine their position in each time unit. Both the seed node and common node only have limited knowledge of their mobility. We assume that a node is unaware of its moving speed and direction, other than knowing its maximum speed is v_{max}. Which means in each time unit a node can move in any direction with speed v where $0 < v \leq v_{max}$, but the exact value of v is unknown.

Initially nodes are deployed randomly over the network area. Two nodes can communicate with each other only if they are within the communication range defined by the radius r. The 1-hop neighbors of sensor p are those sensors that can communicate with it directly i.e. the sensors which are present within radius r. The 2-hop neighbors of sensor p are those who can communicate with the 1-hop neighbors of p directly but not with p. Let suppose a node q is there which can directly communicate with node p, If q is a seed node then we can say that q is p's 1-hop seed node and if q is a common node then we can say that q is p's 1-hop common node. Similarly if there is another node r which cannot communicate with p but can communicate with q directly, then we say r is 2-hop neighbor of p.

Sequential Monte Carlo Localization (SMCL)

The Sequential Monte Carlo (SMC) method [21] provides simulation based solutions to estimate the posterior distribution of non-linear discrete time dynamic models. The posterior distribution is represented using a set of weighted samples, and the samples are updated gradually as the time goes. In each time unit samples are updated using the previous samples and this updated samples are then validated using the observed seed nodes in current time unit.

The Sequential Monte Carlo Localization (SMCL) algorithm [1], is the first algorithm using SMC methods for localization in mobile sensor networks. We can consider SMCL as a 3 step operation for each common node:

Initialization:

Node has no knowledge about its location in the deployment area. N initial samples are selected randomly to represent p's possible positions.

$L_0 = \{l_0^1, l_0^2, \ldots\ldots.., l_0^N\}$

Here N is a constant which represents the number of minimum samples to maintain.

Prediction:

A node starts from the set of possible locations computed in previous step, L_{t-1} and computes a set of n new samples, L_t using the transition equation. The Transition equation $p\ (L_t^i|L_{t-1}^i)$ is determined by the mobility model or other constraints.

In SMCL[1] the Transition equation is given by:

$$P\ (l_t \mid l_{t-1}) = 1/\pi v^2_{max} \text{ if } d\ (l_t, l_{t-1}) < v_{max}$$
$$0 \text{ if } d\ (l_t, l_{t-1}) \geq v_{max} \qquad (1)$$

Where $d\ (l_t, l_{t-1})$ is the distance between two samples l_t and l_{t-1}. So the set of n new samples computed in prediction step contains one location selected randomly from the circle of radius v_{max} around every point in l_{t-1}.

Filtering:

Weights of the new samples found in previous step are computed as $p\ (l_t^i|o_t)$, where o_t is the newly observed seed node in the current time unit. Samples with 0 weight are dropped and if the number of samples after filtering is less that N, then go to step 2.

Let S denotes the set of all 1-hop seed neighbors of N and T denotes set of all 2-hop seed neighbors of N, then the filtering condition of l_t is:

filter (l_t) = *s*□*S*, *d* (*l*, *s*)□*r*□*s*□*T*, *r*□*d* (*l*, *s*)□2*r*

Thus we eliminate the inconsistent locations from possible locations. After filtering if the possible locations are less than N then prediction and filtering process repeats till we obtain N valid samples. After obtaining N valid samples, p calculates its position as the weighted average of all the samples.

Monte Carlo Localization Boxed (MCB)

The Monte Carlo Localization Boxed (MCB)[2] is another version of Sequential Monte Carlo Localization (SMCL). The steps in MCB are similar to those in MCL with difference in the use of seed information and in method for drawing new samples. The MCL algorithm uses 1-hop and 2-hop neighbor information for rejection of impossible samples in filtering step only. In MCB the seed information is used to constrain the sample area, so this method is easy and fast as compared to MCL as the samples are less likely to be filtered in the filtering step. Thus it reduces the number of iterations the algorithm needs to fill the sample set entirely.

Building the Bounding Box : The bounding box is the region of the deployment area where the node is localized. A node that has seed nodes as its 1-hop or 2-hop neighbors, builds a bounding box that covers the region where the neighboring seeds radio range overlaps.

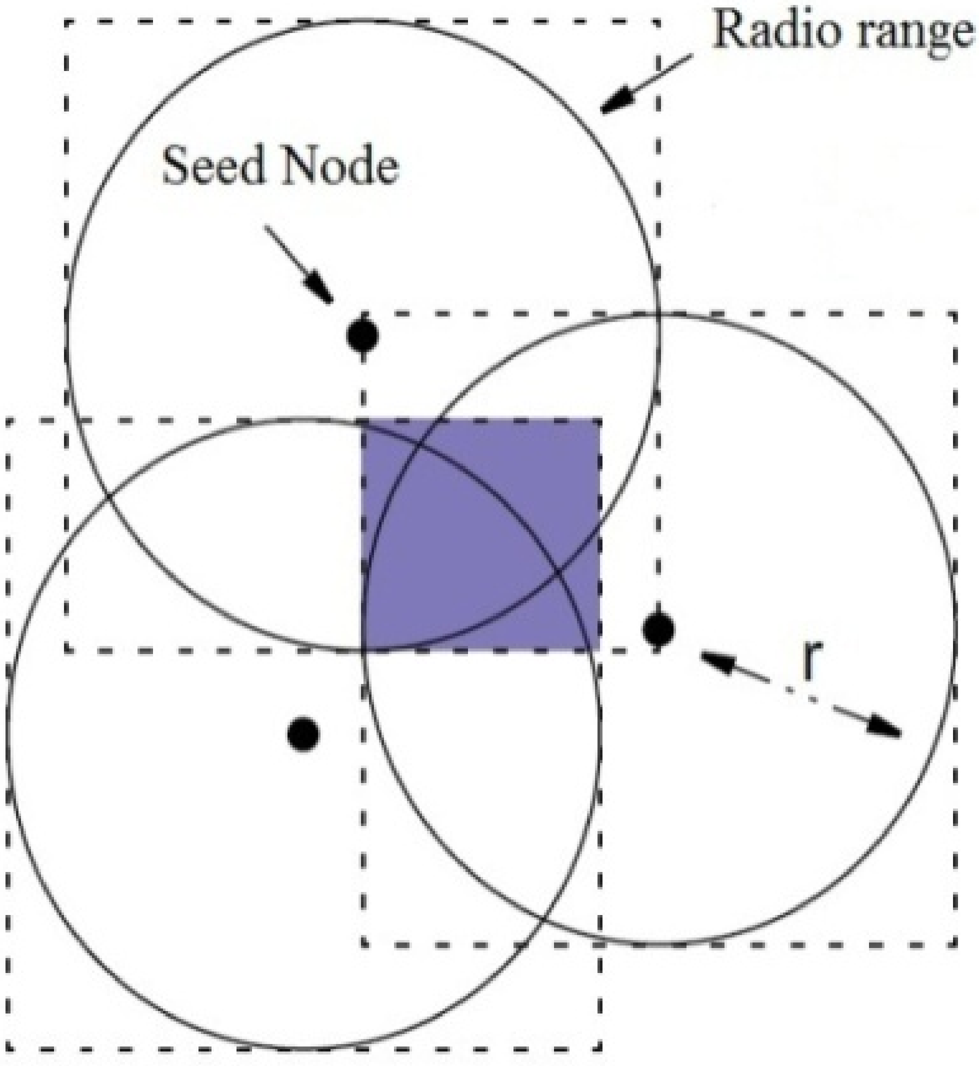

Fig.1. MCB[2] shaded region is the valid sample area.

The bounding box reduces the candidate samples area. It constraints candidate samples into much smaller area called as valid sample area (fig.1.). The valid samples are drawn in this valid sample area only. Building the bounding box simply consists of calculating coordinates (x_{min}, x_{max}) and ($y_{min,}y_{max}$) as follows:

$$x_{min} = max^{n}_{i=1}\{x_i - r\},$$
$$x_{max} = min^{n}_{i=1}\{x_i + r\},$$
$$y_{min} = max^{n}_{i=1}\{y_i - r\},$$
$$y_{max} = min^{n}_{i=1}\{y_i + r\} \qquad (2)$$

where (x_i, y_i) is the coordinate of the i'^{th} 1-hop seed neighbor. 2-hop seed neighbor can be used to reduce the bounding-box further. When using 2-hop seed nodes we should replace *r* with *2r* in the above formula.

Once the bounding box is built a node simply has to draw samples within the region it covers. MCB tries to make best possible use of all information a node has received. During the initialization if the sample set is empty then it allows a node to use 2-hop seed neighbor information even if it has no 1-hop seed. This means that a node that heard only 2-hop seed neighbor can still draw samples using these and produce a location estimate, which is not possible in case of MCL. MCB can also obtain enough samples where SMCL is not able to obtain enough samples, thus achieves higher location accuracy than SMCL.

Our Approach

In this section we will present our approach which is based on MCB and will reduce the computation cost and increases location accuracy. Our approach utilizes the information about the direction of movement and the speed of motion of the common node. The direction of movement is calculated with the help of navigational instrument compass and the speed of motion is calculated with the help of Accelerometer. The information about direction of movement and direction of motion of common node provided by compass and accelerometer respectively will be used in prediction step of MCL to predict N new samples more accurately, hence it will improve localization accuracy.

Compass:

A compass is a navigational instrument that measures directions in a frame of reference that is stationary relative to the surface of the earth. The frame of reference defines the four cardinal directions (or points) – north, south, east, and west. Intermediate directions are also defined. Usually, a diagram called a compass rose, which shows the directions (with their names usually abbreviated to initials), is marked on the compass. When the compass is in use, the rose is aligned with the real directions in the frame of reference, so, for example, the "N" mark on the rose really points to the north. Frequently, in addition to the rose or sometimes instead of it, angle markings in degrees are shown on the compass. North corresponds to zero degrees, and the angles increase clockwise, so east is 90 degrees, south is 180, and west is 270. These numbers allow the compass to show azimuths or bearings, which are commonly stated in this notation.

There are two widely used and radically different types of compass. The magnetic compass contains a magnet that interacts with the earth's magnetic field and aligns itself to point to the magnetic poles. The gyro compass (sometimes spelled with a hyphen, or as one word) contains a rapidly spinning wheel whose rotation interacts dynamically with the rotation of the earth so as to make the wheel process, losing energy to friction until its axis of rotation is parallel with the earth's.

Fig.2.: A HTC Desire S showing a compass app

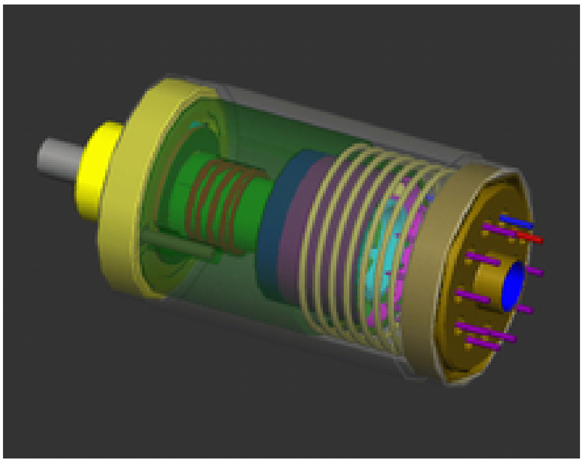

Fig.3: A depiction of an accelerometer

Accelerometer:

An Accelerometer is a device that measures proper acceleration. The proper acceleration measured by an accelerometer is not necessarily the coordinate acceleration (rate of change of velocity). Instead, the accelerometer sees the acceleration associated with the phenomenon of weight experienced by any test mass at rest in the frame of reference of the accelerometer device. For example, an accelerometer at rest on the surface of the earth will measure an acceleration g= 9.81 m/s2 straight upwards, due to its weight. By contrast, accelerometers in free fall or at rest in outer space will measure zero. Another term for the type of acceleration that accelerometers can measure is g-force acceleration.

An accelerometer alone is unsuitable to determine changes in altitude over distances where the vertical decrease of gravity is significant, such as for aircraft and rockets. In our approach the Accelerometer will measure the speed of movement of the node in the prediction step to predict N new samples more accurately, hence it will improve localization accuracy.

Working:

Our approach is based on MCB, all the steps for localization calculations is same as MCB. The difference come in the prediction phase where a node starts from the set of possible locations computed in the previous step, L_{t-1}, and applies the mobility model to each sample to get a set of new samples $L_{t.}$ The set of new samples obtained in the prediction phase are more accurate as compared to MCB as we have information about the direction of movement and the speed of motion of the node. In MCB we do not have any information about the direction of movement and the speed of motion, so MCB takes any random direction for the samples. Hence it gives less accurate localization results as compared to our approach.

Performance Evaluation

In this section, we evaluate the performance of the proposed approach. We consider following two parameters to evaluate the performance:

Localization accuracy:

Localization accuracy is the most important parameter for evaluating localization algorithms. As we have explained earlier that our approach gives better performance in prediction phase as compared to MCB and MCL, so definitely it will give accurate localization results as compared to MCB and MCL. Our approach takes advantage of information about the direction of movement and the speed of motion of the node to improve the localization accuracy.

Cost:

Our approach require navigational instrument compass and Accelerometer to measure acceleration for each node so the set up cost is high but the computational cost is less as compared to MCB or MCL. The computational cost consists of two parts: the cost in generating candidate samples and the cost in evaluating the candidate samples. The cost in generating the candidate sample is less as we know the direction of movement and the speed of motion of the node so we can generate the candidate samples easily in less computation cost and this candidate samples are more accurate as we have information about direction of movement and the speed of motion, so cost is also less for evaluating this candidate samples.

Conclusion and Future work

In this paper, we have presented an accurate range-free localization approach for mobile sensor network. Our approach is based on Monte Carlo Localization Boxed (MCB) and it improves the performance of existing MCB algorithm.

We will further prove that our approach improves the performance of existing MCB algorithm with the help of simulation. We have used Compass and Accelerometer to get high localization accuracy but It has increased our cost. We have to find some method to decrease the number of external devices like Compass and Accelerometer being used. Many issues remain to be explored in future work including the most appropriate compass and accelerometer devices for mobile sensor networks. If we can get information about accurate speed and direction then we can easily achieve high location accuracy. We have to think of a solution which will produce accurate localization and it must be energy and cost efficient as well.

References

[1] L. Hu and D. Evans. Localization for mobile sensor networks. In Tenth International Conference on Mobile Computing and Networking (MobiCom’04), pages 45–57, Philadelphia, Pennsylvania, USA, Sept. 2004.

[2] Aline Baggio and Koen Langendoen. Monte-carlo localization for mobile wireless sensor networks. In MSN06, pages 317–328, 2006.

[3] Z. Shigeng, J. Cao, C. Lijun, and C. Daoxu, “Location Nodes in Mobile Sensor Networks More Accurately and Faster, ” Proc. Ann. IEEE Conf. sensor and Ad Hoc Comm. and Networks, 2008.

[4] M. Rudafshani and S. Datta, “Localization in Wireless sensor Networksl Conf. Information Processing in Sensor Networks (IPSN ’07), pp. 51-60, 2007.

[5] 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 in zebranet. In proceedings of the 10th international conference on Architectural support for programming languages and operating systems (ASPLOS), pages 96-107, 2002.

[6] S. Tilak, V.Kolar, N.B. Abu-Ghazaleh, and K.D. Kang. Dynamic localization control for mobile sensor networks. In 24th IEEE International Performance, Computing and Communications Conference (IPCCC), pages 587-592, 2005.

[7] David K. Goldenberg, Pascal Bihler, Ming Cao, Jia Fang, Brian D.O. Anderson, A. Stephen Morse, and Y. Richard Yang. Localization in sparse networks using sweeps. In Proceedings of the 12th annual international conference on mobile computing and networksing (MobiCom’ 06), 2006.

[8] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In Proceedings of the 9th Annual International Conference on Mobie Computing and Networking (Mobicom’03), pages 81–95, 2003.

[9] P. Biswas and Y Ye. Semidefinite programming for ad hoc wireless sensor network localization. In Proc. of 3rd International Symposium on Porcessing in Sensor Networks (IPSN), 2004.

[10] Lance Doherty, Kristofer S.J. Pister, and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom'01), volume 3, pages 1655–1663, 2001.

[11] Mo Li and Yunhao Liu. Rendered path: Range-free localization in anisotropic sensor networks with holes. In Proceedings of the 13th annual ACM international conference on Mobile computing and networking (Mobicom'07), pages 51–62, Montreal, Que´ebec, Canada, September 2007. ACM.

[12] H. Lim and J.C. Hou. Localization for anisotropic sensor networks. In Proc. of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), pages 138–149, 2005.

[13] Dragos Niculescu and Badri Nath. Ad-hoc positioning system (aps). In Proceedings of 2001 IEEE Global Telecommunications Conference (Globecom'01), pages 2926–2931, San Antonio, TX, USA, May 2001. IEEE Communications Society.

[14] Andreas Savvides, Chih-Chieh Han, and Mani B. Srivastava. Dynamic fine-grained localization in ad-hoc sensor networks. In Proceedings of the 7th Annual international Conference on Mobile Computing and Networking (Mobicom'01), pages 166–179, Rome, Italy, 2001. ACM.

[15] Y. Shang and W. Ruml. Improved MDS-Based localization. In Proc. of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom'04), pages 2640–2651, 2004.

[16] Hongyi Wu, Chong Wang, and Nian-Feng Tzeng. Novel selfconfigurable positioning technique for multihop wireless networks. IEEE/ACM TRANSACTIONS ON NETWORKING, 13 (3):609–621, June 2005.

[17] Bram Dil, Stefan Dulman, and Paul Havinga. Range-based localization in mobile sensor networks. In ewsn06, pages 164–179, 2006.

[18] Jiyong Yi, Sungwon Yang, and Hojung Cha. Multi-hop-based monte carlo localization for mobile sensor networks. In Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'07), pages 162–171, June 2007.

[19] J. E. Handschin. Monte Carlo Techniques for Prediction and Filtering of Non-Linear Stochastic Processes. Automatica 6. pp. 555-563.1970.

[20] http://en.wikipedia.org/wiki/Compass.