A
Dissertation
On

"Improvement in Sensor Node Localization"

Submitted in Partial fulfillment of the requirement
For the award of Degree of

MASTER OF TECHNOLOGY
Computer Technology and Application
Delhi Technological University, Delhi

SUBMITTED BY

SATYENDRA YADAV
University Roll No: 16/CTA/2010

Under the Guidance of:

Mr. VINOD KUMAR

Associate Professor
Delhi Technological University

DEPARTMENT OF COMPUTER ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
BAWANA ROAD, DELHI-110042

2010-2012

CERTIFICATE

This is to certify that the work contained in this dissertation entitled "**Improvement in Sensor Node Localization**" submitted in the partial fulfillment, for the award for the degree of M. Tech. in Computer Technology and Applications at **DELHI TECHNOLOGICAL UNIVERSITY** by **SATYENDRA YADAV, Roll No. 16/CTA/2010** is carried out by him under my supervision. This matter embodied in this project work has not been submitted earlier for the award of any degree or diploma in any university/institution to the best of our knowledge and belief.

(Mr. VINOD KUMAR)
Project Guide
Associate Professor
Department of Computer Engineering
Delhi Technological University

ACKNOWLEDGEMENT

First of all, let me thank the almighty god, my parents and my dear friends who are the most graceful and merciful for their blessing that contributed to the successful completion of this project.

I feel privileged to offer sincere thanks and deep sense of gratitude to Mr. VINOD KUMAR, project guide for expressing his confidence in me by letting me work on a project of this magnitude and using the latest technologies and providing their support, help & encouragement in implementing this project.

I would like to take this opportunity to express the profound sense of gratitude and respect to all those who helped us throughout the duration of this project. DELHI TECHNOLOGICAL UNIVERSITY, in particular has been the source of inspiration, I acknowledge the effort of those who have contributed significantly to this project.

Satyendra Yadav

(Roll No.: 16/CTA/2010)

TABLE OF CONTENTS

List Of Figures

List Of Tables

Abstract

The most fundamental problem of wireless sensor networks is localization (finding the geographical location of the sensors). Most of the localization algorithms proposed for sensor networks are based on Sequential Monte Carlo (SMC) method. To achieve high accuracy in localization it requires high seed node density and it also suffers from low sampling efficiency. There are some papers which solve this problems but they are not energy efficient. Another approach the Bounding Box 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 thesis we propose an energy efficient approach which will further reduce the scope of searching the candidate samples, so now we can remove the invalid samples from the sample space and we can introduce more valid samples to improve the localization accuracy. We will consider the direction of movement of the valid samples, so that we can predict the next position of the samples more accurately, hence we can achieve high localization accuracy. Further we can also add the information about the speed of movement of the node so that we can measure the actual acceleration of the node.

Now as we have information about the direction and speed of movement of the node we can locate a sensor node more accurately and faster.

Chapter: 1 Introduction

1.1 Objective

The objective of this thesis is to design a new localization algorithm that can work effectively and accurately to locate the position of the sensor node. Our algorithm can provide an efficient range free solution for the localization problem of the sensor node.

The fundamental problem in designing sensor network is localization- determining the location of sensors. Traditional methods 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 in [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:

1) 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.

2) 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.

3) 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.

1.2 Problem Statement

The objective of this research work is to provide an energy efficient range-free localization algorithm which can achieve high localization accuracy in various scenarios. So that, this algorithm can be used in place of expensive GPS (Global Positioning System)

The desired algorithm must be generic and can be customized for any application. It must also be adaptable with the changing needs and requirements of the application.

1.3 Motivating Factor

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 localization problem is even more important in wireless sensor networks for the following reasons:

1. Many WSN protocols and applications simply assume that all nodes in the system are location-aware.
2. If a sensor is reporting a critical event or data, we must know the location of that sensor.
3. If a WSN is using a geographical routing technique, all of the nodes must be aware of their location.

Traditional methods 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.

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 thesis we will consider all above challenging aspects of localization problem, and we will propose a localization algorithm which will outperform the existing algorithms in terms of localization accuracy.

1.4 Organization of the Dissertation

This thesis work is organized as follows

Chapter 1 deals with providing the objective, problem statement, motivation of undertaking this research work as well as organization of this dissertation.

Chapter 2 deals with the concept of Wireless Sensor Networks. It also provides the basic knowledge of various challenges associated with Wireless Sensor Networks. The localization problem is also explained to helps us in better understanding of needs and requirement of an accurate localization algorithm.

Chapter 3 provides introduction to the network model and basic overview of the localization strategies, which are basis for our approach. It also provides knowledge about various steps involved in localization algorithm.

Chapter 4 begins with description of our research work. It describes the basic design of our approach. We have explained various steps involved in our approach in detail with an example.

Chapter 5 gives the basic overview of the simulation parameters used in our experiment. It also explains the outcomes of the experiments with the help of graphs for different iterations. We have calculated average localization error and standard deviation for each iteration. In the end all the iterations are compared with the help of a graph.

Chapter 6 gives the final findings and outcomes of the research. It lists the problems that we have solved and those that still remain to be tackled. It also lays the ground to the future work in this direction.

Chapter: 2

Literature Review

2.1 Wireless Sensor Network

A WSN is typically formed by deploying many sensor nodes in an ad hoc manner. These nodes sense physical characteristics of the world. The sensors could be measuring a variety of properties, including temperature, acoustics, light, and pollution. Base stations are responsible for sending queries to and collecting data from the sensor nodes.

Some of the main characteristics of a networked sensor include:

(1) Small physical size,
(2) Low power consumption,
(3) Limited processing power,
(4) Short-range communications, and
(5) A small amount of storage.

Individually, these resource-constrained devices appear to be of little value. Deploying these sensors in large scale across an area of interest, however, is when they can be most effective. Placing sensors in hostile or inaccessible regions may allow for data collection which was previously impossible. Spatial and temporal processing as well as dense monitoring is now feasible. The sensors must be able to form an ad hoc network, and use collaborative techniques to monitor environment and respond to users whenever required.

Wireless sensor networks provide the means to link the physical world to the digital world. The mass production of integrated, low-cost sensor nodes will allow the technology to cross over into a myriad of domains. In the future, applications of wireless sensor networks will appear in areas we never dreamed. Listed below are just a few places where sensor networks can and will be deployed.

- Earthquake monitoring
- Environmental monitoring
- Factory automation
- Home and office controls
- Inventory monitoring
- Medicine
- Security

2.2 WSN Challenges

Although some applications have shown promise, the field of wireless sensor networks still provides many challenges to researchers:

Data storage – Sensors are sampling the environment continuously. With the limited storage capacity of the networked sensors, volumes of data cannot be stored permanently. Data has to be compressed, filtered and aggregated with data from other nodes, and stale data must be purged. Should the data be stored in the network or should it be routed offline to a central server?

Energy efficiency – Some form of battery typically powers networked sensors. When large networks of sensors are deployed, they are expected to run unattended for long periods of time. Writing energy-efficient algorithms that conserve the battery could extend the lifetime of an application by months. Energy conservation techniques are to be designed at all of the networking layers, from the physical layer to the application layer, and for various applications.

Fault tolerance – In early generations of networked sensors, there are high malfunction and failure rates. In most sensor applications, it is not feasible for a human to physically traverse a region to repair and replace nodes. A significant percentage of sensor nodes may fail when deployed in hostile environments. Therefore, techniques must be provided by the system, so that the application continues running without interruption when nodes become faulty or die.

Localization – Using wireless sensor networks to locate or track things is an application that is attracting much attention lately. There are many sensor network protocols and applications that assume every node knows its location. How is this possible? If every node were equipped with a GPS component, both the financial and energy cost of a large sensor network would become exorbitant. If a small fraction of the nodes are aware of their location, is it possible for the remaining nodes to discover their location?

Scalability – The applications that are envisioned for sensor networks in the near future will use thousands of sensors. How do you get thousands of nodes to self-organize and work together? Centralized algorithms must sometimes give way to distributed algorithms, when applications are being considered for networks of this scale. The deployment and management of thousands of tiny devices are issues that must be addressed.

Security – Any network application that uses a wireless medium inherently assumes a security risk. Eavesdropping to obtain information and jamming to deny service are a couple of ways that a sensor network system may be attacked. What can be done to make sure a wireless sensor network provides important features such as availability, reliability, freshness, and privacy?

2.3 Localization Problem

‘The procedure through which the nodes obtain their positions is called localization’

In many applications of wireless sensor networks, precise location information of sensor nodes is critical to the success of the applications. Most data collected from sensors are only meaningful when they are coupled with the location information of the corresponding sensors. Consider an application of habitat monitoring; thousands of sensors are dropped in the targeted region of a tropical rain-forest by an aeroplane. Nodes are equipped with sensing devices to monitor the changes of temperature and humidity of the environment. To make every measurement useful to scientists, the location where measurements are taken has to be known.

Localization in wireless sensor networks is to determine the geographical positions of sensors in a wireless sensor network.

The most trivial solution is manual configuration. The location of each sensor is predetermined before deployment. Sensors are installed to the assigned locations by human. Obviously, this solution is inscalable as much labour is required for the installation. Furthermore, it is sometimes infeasible to have manual configuration as the location information of sensors is unknown before actual deployment. Recalled the previous example of habitat monitoring, sensors are dropped from an aeroplane, in which exact locations are only known when sensors land on the forest.

Another solution for localization is equipping every sensor with a GPS receiver. Sensors can locate themselves individually using the GPS signals. However, installing a GPS receiver for every sensor node greatly increases the total cost of the sensor network. In addition, the introduction of GPS receiver increases the energy consumption of a node and hence shortens its life time. Lastly, the location obtained from GPS-receiver may not be precise enough for certain applications and the accuracy of GPS is affected by various environmental factors. Accuracy can be of tenths of meters for general GPS. The error can be lowered to less than ten meters for GPS augmentation systems like Differential GPS (DGPS) but with a higher cost.

Realizing the challenges for network localization, this dissertation aims to study the localization problem in sensor networks. We try to study the practical problem to find the tradeoff between the accuracy and energy cost for achieving excellent localization accuracy in sensor networks.

Chapter: 3

Research Background

3.1 Network Model

We have two 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.

3.2 Sequential Monte Carlo Localization

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.

Location Estimation Algorithm [1]:

Initialization: Initially the node has no knowledge of its location. N is the constant that denotes the number of samples to maintain

$$L_O = \{\text{set of N random locations in the deployment area}\}$$

Steps: Compute a new possible location set L_t based on L_{t-1} the possible location set from the previous time step, and the new observations, o_t

$L_t = \{ \}$
while (size(L_t) < N) **do**
  $R = \{l_t^i \mid l_t^i$ is selected from $p(l_t|l^i_{t-1}), l^i_{t-1} \in L_{t-1}$ for all $1 \le i \le N\}$ **Prediction**
  $R\text{filtered} = \{ l^i_t \mid l^i_t$ where $l^i_t \in R$ and $p(o_t|l^i_t) > 0\}$ **Filtering**
  $L_t = \text{choose}(L_t \cup R_{filtered}, N)$

The mobile localization problem can be stated in a state space form as follows. Let t be the discrete time, lt denote the position distribution of the node at time t, and o_t denote the observations from seed nodes received between time t-1 and time t. A transition equation $p(l_t \mid l_{t-1})$ describes the prediction of node's current position based on previous position, and an observation equation $p(l_t \mid o_t)$ describes the likelihood of the node being at the location in the given observations. We are interested in estimating recursively in time the filtering distribution $p(l_t \mid o_0, o_1, \ldots, o_t)$. A set of N samples L_t is used to represent the distribution l_t, and our algorithm recursively computes the set of samples at each time step. Since L_{t-1} reflects all previous observations, we can compute lt using only L_{t-1} and o_t.

Initially we assume the node has no knowledge about its position, so the initial samples are selected randomly from all possible locations. At each time step, the location set is updated based on possible movements and new observations. We estimate the location of the node by computing the average location of all possible locations in L_t. we assume locations are (x, y) positions in two dimensional Cartesian space, but the technique could be used equivalently for three dimensions or other location representations.

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 $p(l_t^i \mid l_{t-1}^i)$ is determined by the mobility model or other constraints. It is assumed that the node has no information about its speed and direction, but it knows its speed is less than v_{max}. So if l_{t-1}^{i} is one possible location of a node in previous step, then the possible current positions are contained in the circular region with origin l_{t-1}^{i} and radius v_{max}. The Transition equation $p(l_t^i \mid l_{t-1}^i)$ is determined by the mobility model or other constraints.

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

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

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 vmax around every point in l_{t-1}. The uncertainty about the node's location is very high because of unknown motion of the node. In case where some information is known about the node's motion like it is moving with some known speed or it is likely to move in a certain direction, then the probability distribution can be adjusted accordingly to make better predictions.

Filtering:

In filtering, we remove the invalid locations on the basis of the new observations. Here we assume that the time is discrete and all messages are received instantly. Hence a location announcement by a seed will be heard by every node within the radio range of the seed. In a realistic environment, we have to deal with network collisions and account for missed messages. 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 prediction step.

There are four types of seeds to consider:

a) Outsiders: The seeds that were not heard in either the current or the previous time quanta.

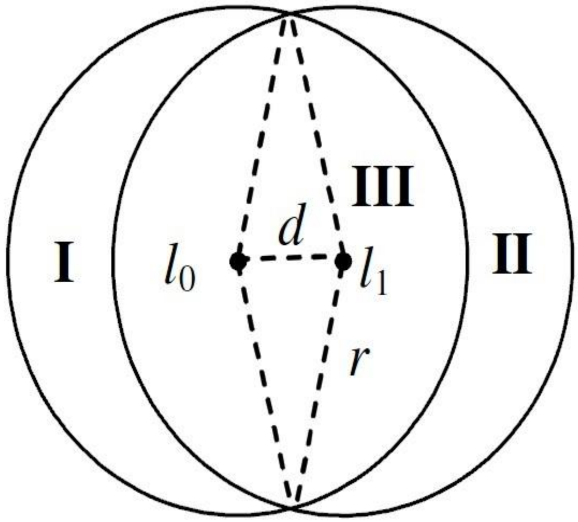

Fig.3.1. Seed Movement

b) Arrivers: The seeds that were not heard in previous time quanta but heard in the current one.
c) Leavers: The seeds that were not heard in current time quanta but heard in previous one.
d) Insiders: The seeds that were heard in both time quanta.

In fig 3.1 the seed moves from l_0 at time 0 to position l_1 on time 1. The seed is an arriver for nodes in region II, an insider for nodes in region III, an outsider for all other nodes, and a leaver for nodes in region I. Arrivers and Leavers provide the most useful information that the node was within distance r of l_0 at time t_0, but not within distance r of l_1 at time t_1.

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:

$$filter(l_t) = \forall s \in S, d(l, s) \leq r \wedge \forall s \in T, r < d(l,s) \leq 2r \tag{2}$$

The probability distribution is zero, if the filter condition is false and evenly distributed otherwise. 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.

3.3 Monte Carlo Localization Boxed

Despite being quite accurate, especially in low-anchor configurations, MCL's efficiency can be improved. Drawing samples is a long and tedious process that could easily drain a lot of energy from a sensor node. Furthermore, the way MCL makes use of anchor information leaves room for improvement. This version of the Sequential Monte Carlo Localization called Monte Carlo Localization Boxed (MCB) uses steps similar to those of MCL. The major differences lie in the way we use anchor information and the method we use for drawing new samples.

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.

In SMCL we have two areas, Candidate sample area and Valid sample area (fig 3.2). The Candidate sample area is used to draw new candidate samples into the deployment area whereas the Valid sample area is used to filter out the invalid candidate samples drawn in the prediction step. If the Candidate sample area is large and the Valid sample area is small, the candidate samples drawn in prediction phase have high probability to be filtered out in the filtering phase. Now from the transition equation (1) and filtering condition (2) we know that the candidate samples area will be large when v_{max} is large and the valid samples area will be small when s_d is large. So SMCL will be very time-consuming when v_{max} and s_d is large.

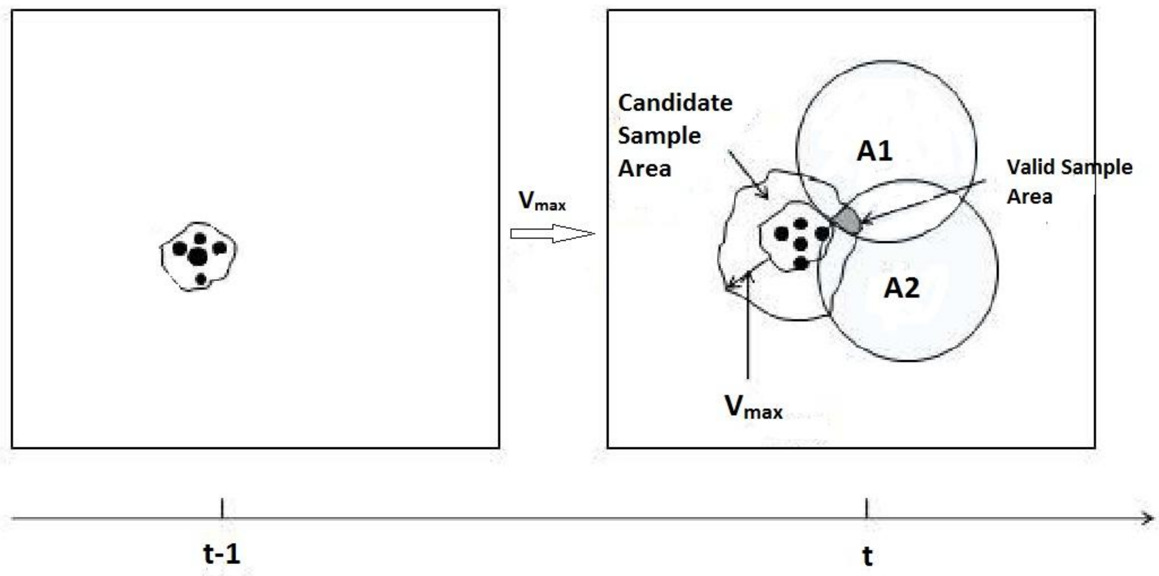

Fig.3.2 How SMCL[1] works

To overcome above problem, we can use a Bounding Box Method [2] to reduce the candidate sample area. The main idea of this method is to constrain samples into much smaller area. The perfect solution is to draw candidate samples from the valid samples area only. However, the valid sample area is very hard to obtain but we can construct an approximation of that area using bounding box.

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. We can also call the bounding box as the region of the deployment area where the node is localized.

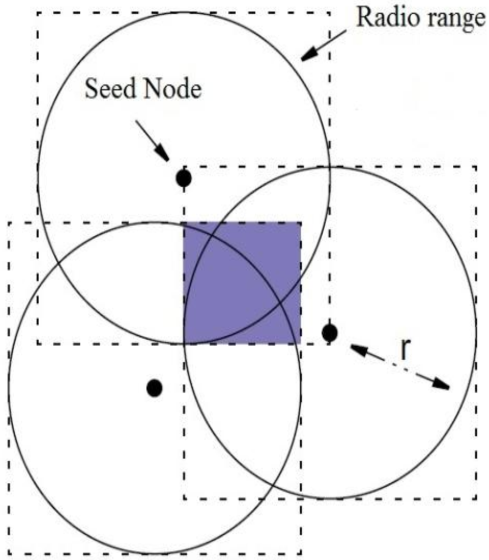

Fig.3.3. MCB[2] shaded region is the valid sample area.

The bounding box reduces the candidate samples area. It constraints candidate samples into much smaller area called as valid sample area, and the valid samples are drawn in this valid sample area only. Fig 3.3 shows an example of a bounding box, where three one-hop sees were heard. A node builds a square of size 2*r* centered at the seed position for each one-hop anchor heard and *r* being the radio range. Building the bounding box simply consists of calculating coordinates (x_{min}, x_{max}) and (y_{min} , y_{max}) as follows:

$$x_{min} = \max^{n}_{i=1} \{x_i - r\} ,$$
$$x_{max} = \min^{n}_{i=1} \{x_i + r\},$$
$$y_{min} = \max^{n}_{i=1} \{y_i - r\} ,$$
$$y_{max} = \min^{n}_{i=1} \{y_i + r\} \qquad (3)$$

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. Using this method, the probability for a candidate sample to be reserved in the final set increases very much, so the computation cost is reduced. 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 SMCL. MCB can also obtain enough samples where SMCL is not able to obtain enough samples, thus achieves higher location accuracy than SMCL.

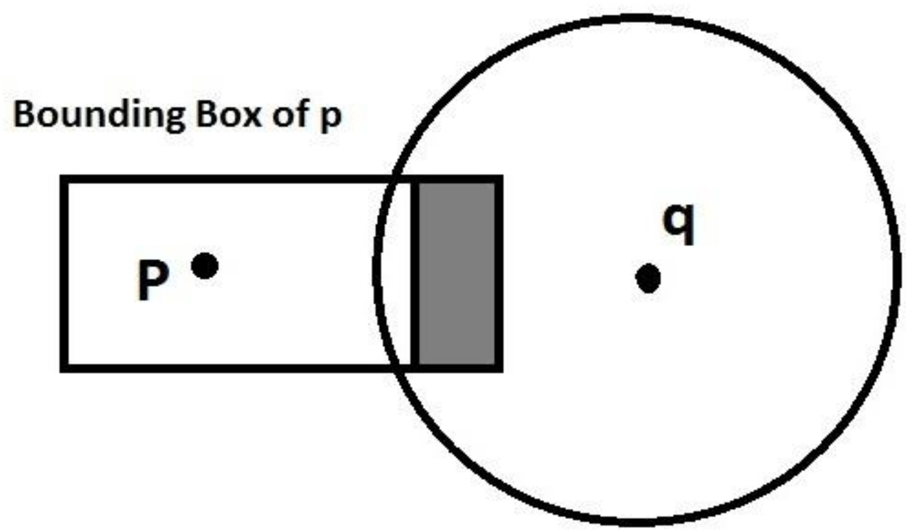

Fig.3.4. Reducing size of Bounding Box

We can further reduce the size of bounding box as follows: Suppose that a bounding box (x_{min}, x_{max}, y_{min}, y_{max}) has been built as above. We can reduce the size of bounding box by using two-hop beacon's negative effect (Fig.3.4.). Assuming *q* is *p's* two-hop beacon neighbor, then the shadowed region doesn't contain *p* otherwise *q* will be *p's* one-hop neighbor. So we can eliminate the shadowed region without any loss of valid samples.

Chapter: 4

Proposed 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 of the common node with the help of navigational instrument compass. The information about direction of movement of common node provided by compass will be used in prediction step of MCL to predict N new samples more accurately, hence it will improve localization accuracy.

4.1 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, 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.

Fig.4.1.: A HTC Desire S showing a compass app

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.

We will attach this navigational device compass with each sensor node. So that it will have information about its direction of movement. Now as the direction information is available, the next position of the sensor node can be predicted more accurately. In MCL and MCB both we don’t have any information about the direction, so there is a lot of inaccuracy while predicting the current location based on the previous location. This inaccuracy in prediction affects the localization results badly for both MCL and MCB.

In our approach we will remove this inaccuracy in prediction of the current location by using the direction of movement information provided by the compass. Hence we can achieve high localization accuracy.

4.2 Our Approach

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 of the node. In MCB we do not have any information about the direction of movement, so MCB takes any random direction for the samples. Hence it gives less accurate localization results as compared to our approach.

Steps for localization:

As mentioned above the steps for localization are same as MCB, but the difference comes in the prediction phase. The detailed steps are as follows:

1. **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.

2. **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 $p(l_t^i \mid l_{t-1}^i)$ is determined by the mobility model or other constraints. The node has no information about its speed but it knows that its speed is less than v_{max} and also the direction of its movement. So, if l_{t-1}^i is one possible location of a node in previous step, then the possible current positions will be in the same direction as information provided by the navigational device compass attached with the node and it must be contained in the circular region with origin l_{t-1}^i and radius v_{max}.

The uncertainty about the node’s location is very less as we have direction of node’s motion. Now we can predict node’s current position in its actual direction of movement, which is not possible in case of MCL or MCB.

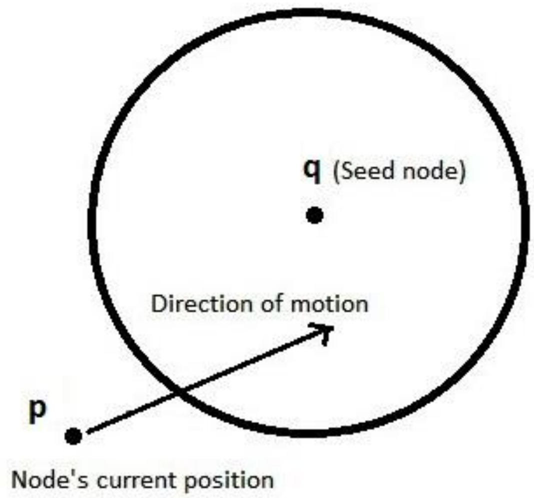

Fig.4.2. Prediction Example

In fig 4.2, current position of node p is shown, now to predict next position we have information about its direction of motion. So we will predict its next position in that direction only.

Here we will have two possible scenarios about node N’s next position:

a) Node p is not in radio range of Seed node q: If in the next position Node N is not in radio-range of q then we can predict p’s next position outside the radio range of q in its direction of motion.
b) Node p is in radio-range of Seed node q: In this case as node p is in radio-range of q we will predict p’s next position inside the radio-range of q in its direction of motion.

3. **Filtering:** In filtering, we remove the invalid samples which are inconsistent with the current observation. In our approach as we will have information about the direction of motion, we will remove all those samples which are not in the same direction as provided by the navigational device compass. 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.

Chapter: 5
Experimental Results

In this section, we will evaluate the performance of proposed approach through extensive simulations by measuring how its estimated location errors vary with various network and algorithm parameters. The key metric for evaluating a localization approach is the location estimation accuracy versus the deployment and communication cost. The location accuracy can be improved by increasing the density of the seeds or the frequency of the location announcements, but to determine appropriate deployment parameters the tradeoffs need to be understood.

5.1 Simulation Parameters

In our experiments, we vary parameters of both the sensor network and sensor nodes.
The various simulation parameters are as follows:

1) **The terrain area:**

For all our experiments, the deployment area is of 500m x 500m rectangular region. The sensor nodes are randomly distributed in this rectangular region.

2) **The node information:**

Node Density (n_d) - Node density is the average number of nodes including both nodes and seeds in one hop transmission range. For our experiments, we have taken Node Density $n_d = 10$.

Sees Density (s_d) – The average number of seed nodes in one hop transmission range is called as Seed Density. We have taken Seed Density $s_d = 1$

In our experiments, as per the above Node and Sees density we have taken 320 Nodes which includes both the common node and the seed nodes, so we have 32 seed nodes.

3) The Mobility Model:

We have implemented the movement of sensors using the modifies version of random waypoint mobility model[23] used by Hu and Evans[1]. This model prevents nodes from pausing at way-points.

Speed of Nodes and Seeds (v_{max} , v_{min} , s_{max} , s_{min}) :

We have represented speed as the moving distance per time unit. We have taken minimum speed for both nodes and seed as 0 and the maximum speed is taken as 10. A nodes speed is randomly chosen from [v_{max} , v_{min}] and the seeds speed is randomly chosen from [s_{max} , s_{min}].

4) The Radio Model:

We have set the communication range as 50 for both nodes and seeds for our experiments. We have assumed that a node can judge that it is in radio range *r* of seed node or not, but it cannot more precise distance information like measuring distance from received radio signal strength. We have also assumed that the radio range of nodes and seeds as a perfect circle, however it is not realistic.

5.2 Simulation Results

To analyze the simulation results, we will analyze the localization error. The localization error is calculated by measuring the distance between the estimated location of the node and the actual location of the node. As we have used the random waypoint mobility model for our experiments the movement of the nodes is random in the deployment area, hence the results of our experiments is also random. So we have taken results for 5 iterations, in which each iteration is of 100 steps. In each iteration we have considered result for 100 steps of the node, we have calculated error in each step, and after 100 steps we have calculated average of all 100 steps for each iteration and considered it as the average localization error for that iteration. Now when all the 5 iterations are finished then we have calculated average error for all the iteration, and considered it as the average localization error for our algorithm.

In order to show the simulation results we have plot a graph for each iteration. The graph is plotted between localization error and the steps for each iteration. In the end of this section we have shown results in tabular form. In which we have shown result of each step for different iterations.

1) Iteration 1

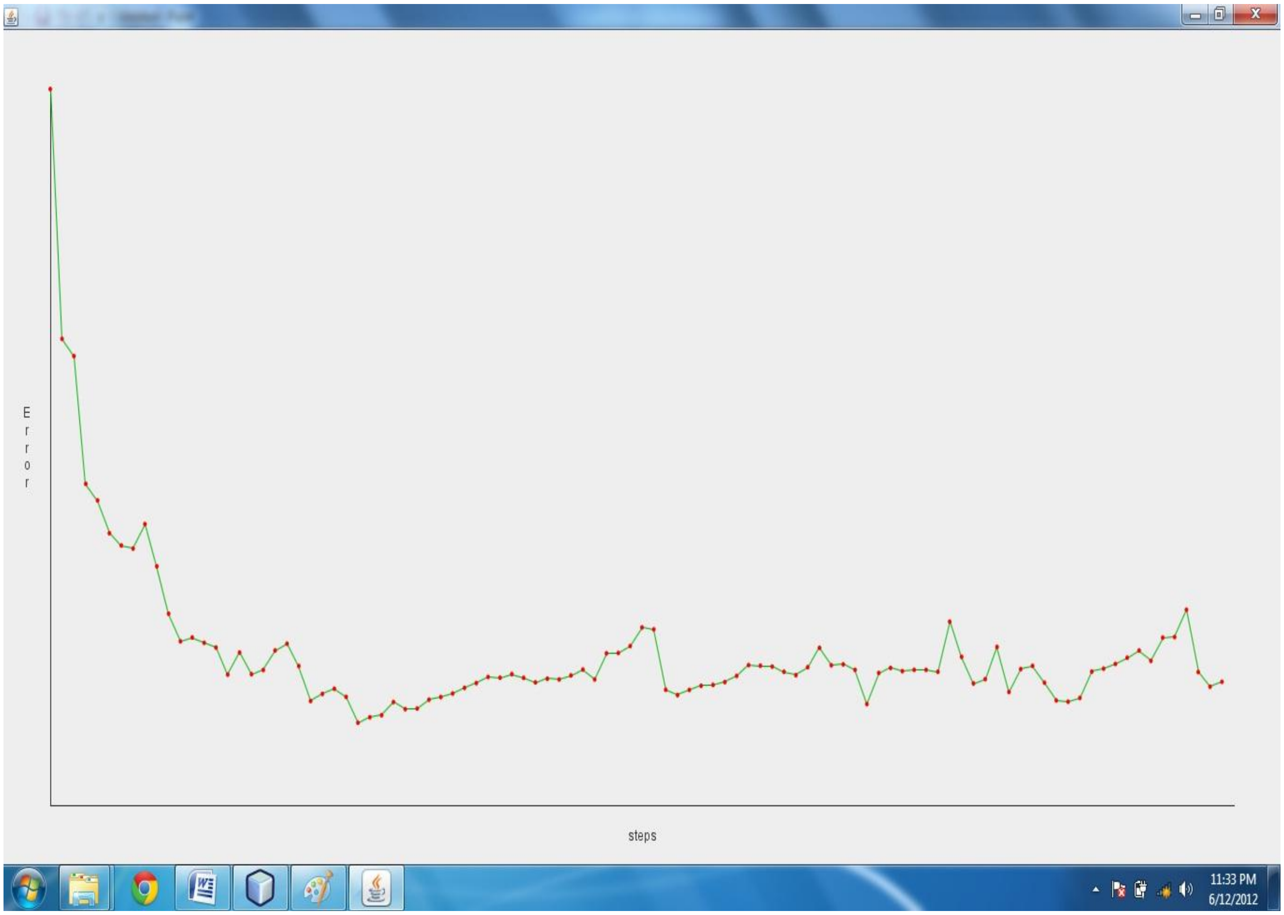

Fig.5.1 Simulation Result: Iteration 1

In iteration 1, the average estimation error in Modified MCB algorithm is: 25.61311861092864

And, the standard deviation in modified MCB algorithm is: 16.12

2) Iteration 2

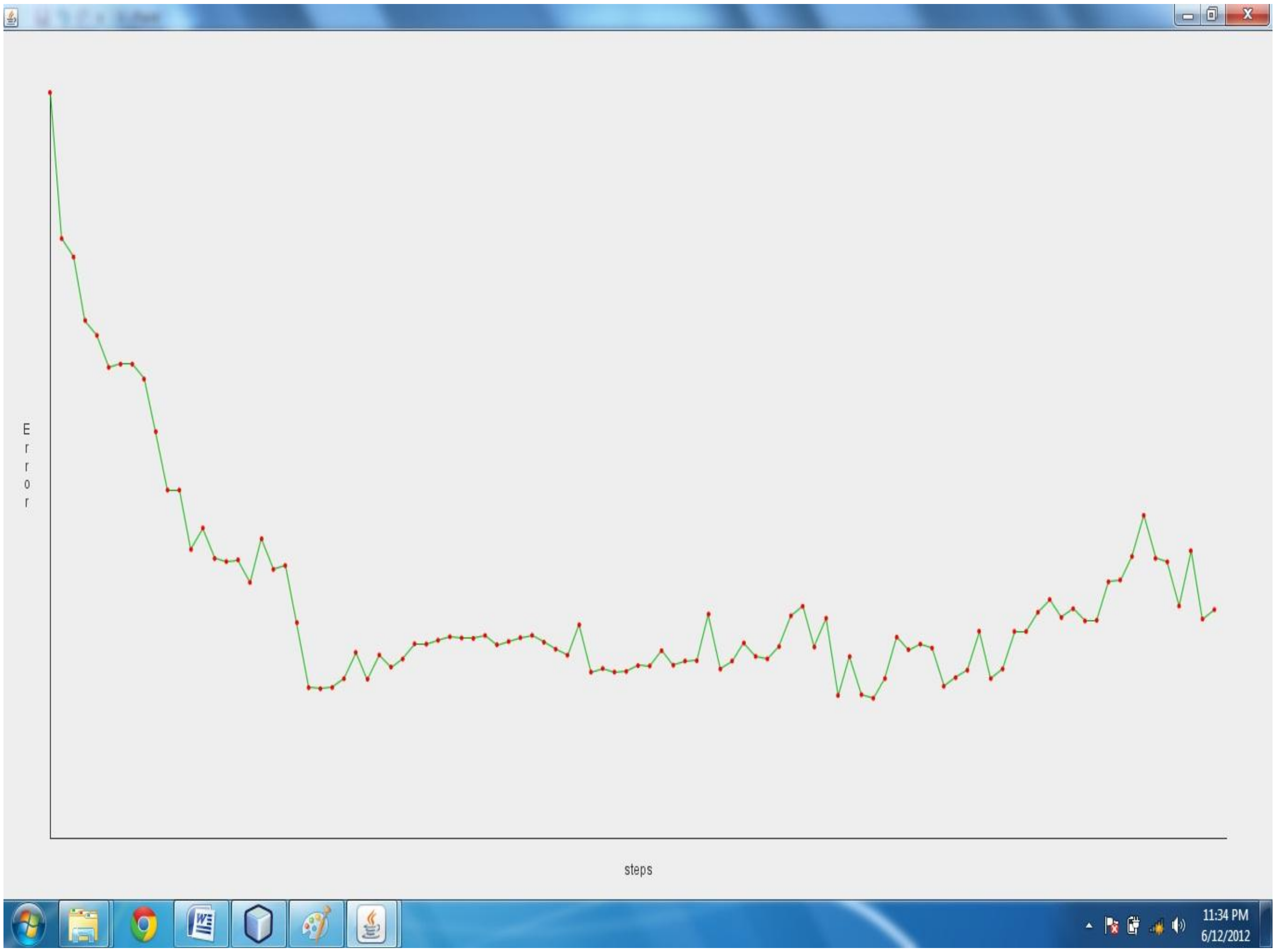

Fig.5.2 Simulation Result: Iteration 2

In iteration 2, the average estimation error in Modified MCB algorithm is: 27.626264805403828

And, the standard deviation in modified MCB algorithm is: 17.63

3) Iteration 3

Fig.5.3 Simulation Result: Iteration 3

In iteration 3, the average estimation error in Modified MCB algorithm is: 21.197533907504333

And, the standard deviation in modified MCB algorithm is: 18.26

4) Iteration 4

Fig.5.4 Simulation Result: Iteration 4

In iteration 4, the average estimation error in Modified MCB algorithm is: 25.3396640168027

And, the standard deviation in modified MCB algorithm is: 16.34

5) Iteration 5

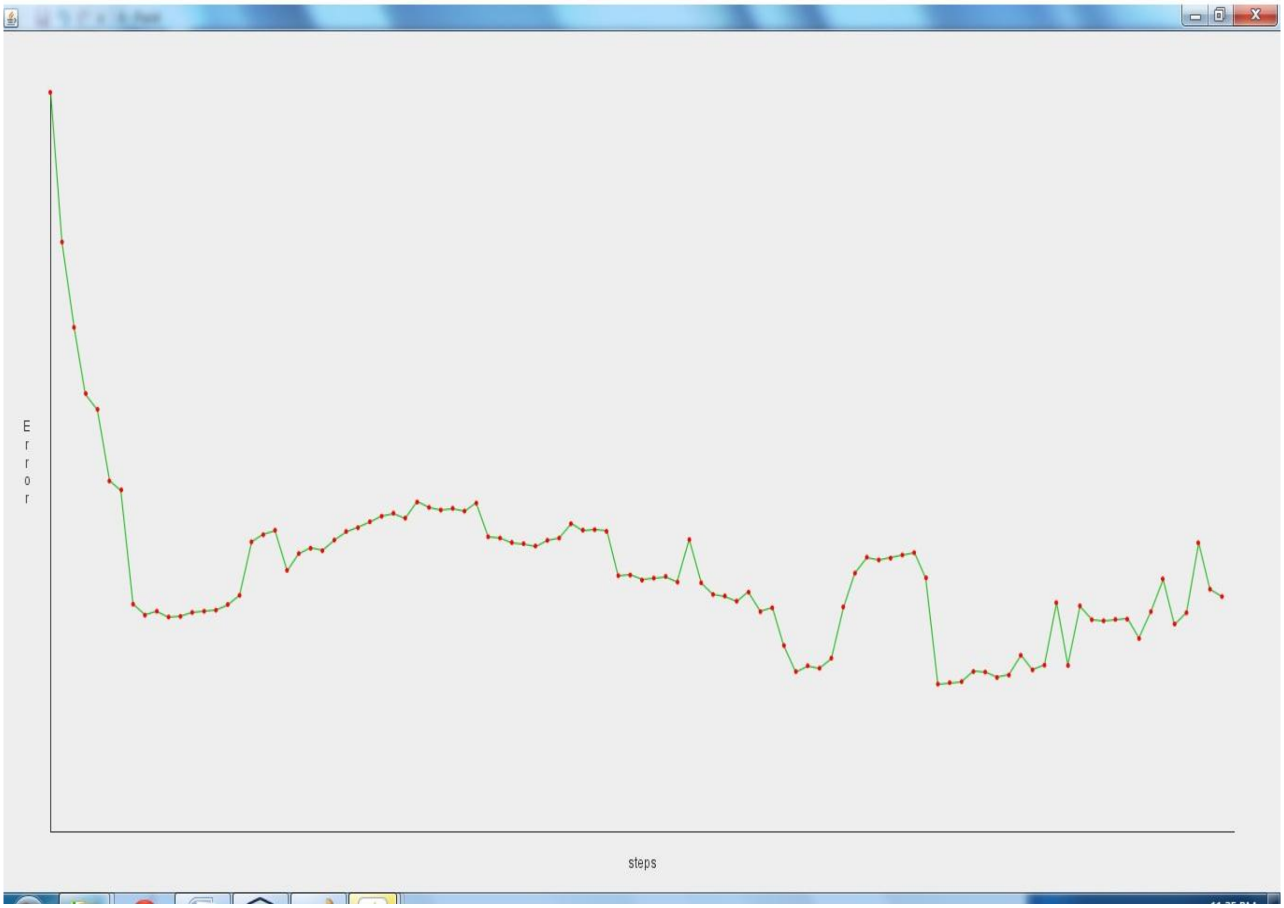

Fig.5.5 Simulation Result: Iteration 5

In iteration 5, the average estimation error in Modified MCB algorithm is: 25.601242017568072

And, the standard deviation in modified MCB algorithm is: 17.08

6) Average for all 5 Iteration

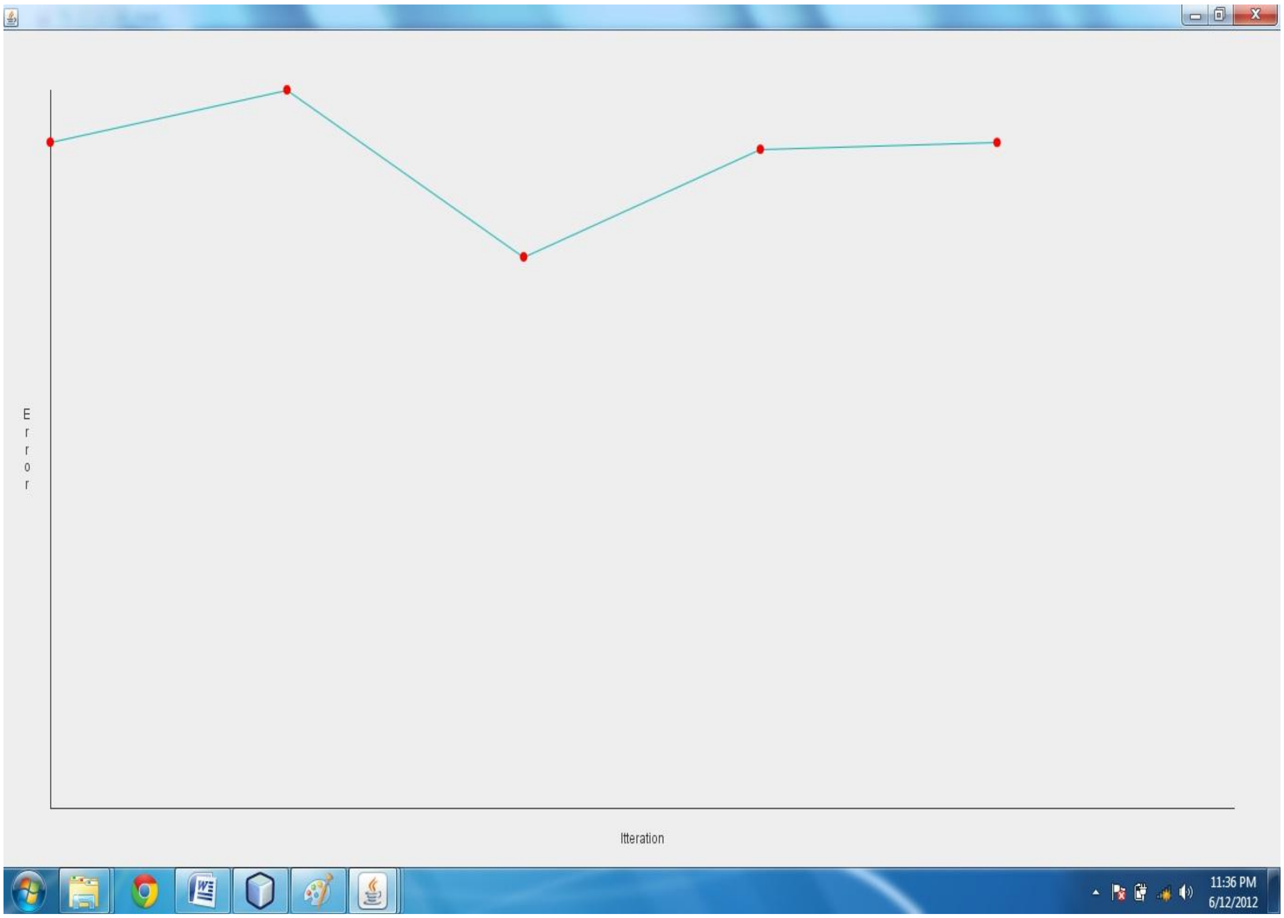

Fig.5.6 Simulation Result: Average for all Iterations

Average estimation error in Modified MCB algorithm is: 25.075564671641512

We can show our simulation results in the tabular form as follows:

Serial No	Iteration Number	Estimated Error
1	Iteration 1	25.61311861092864
2	Iteration 2	27.626264805403828
3	Iteration 3	21.197533907504333
4	Iteration 4	25.3396640168027
5	Iteration 5	25.601242017568072
6	Average for all 5 Iteration	25.075564671641512

Table 5.1. Estimated error for all Iterations

Serial No	Iteration Number	Standard Deviation
1	Iteration 1	16.12
2	Iteration 2	17.63
3	Iteration 3	18.26
4	Iteration 4	16.34
5	Iteration 5	17.08

Table 5.2. Standard Deviation for all Iterations

Chapter: 6 Conclusion and Future Work

6.1 Conclusion

Localization in wireless network has received much interest in the past years. In this thesis, we have presented an efficient and accurate range-free localization algorithm for wireless sensor networks which works well in both static and mobile wireless sensor networks. The main drawback of the existing localization algorithms for localization is there high computation cost and poor localization accuracy.

The proposed algorithm improves the performance of existing Monte Carlo Boxed (MCB) Localization algorithm. In our algorithm we have used information about direction of the movement of the node, so that we are able to predict the next position of the node more accurately and faster as compared to existing Monte Carlo Boxed Localization algorithm. We have attached a device called Compass to each node so that, we can easily get this information about direction of the movement of every node. Our algorithm outperforms existing Sequential Monte Carlo (SMC) based algorithms in terms of location accuracy. Our algorithm can produce more accurate localization results under high node density. Also, even when there are a very few seed nodes, most nodes still get accurate position estimations.

6.2 Future Work

In the future, we are planning to further enhance the performance of our algorithm. We will try to include some more information about the node's movement like speed using some additional equipment like accelerometer which will further improve the performance of our algorithm. But we need to understand the tradeoff between the accuracy and energy cost for proposing excellent localization algorithms for sensor networks.

We can also try to improve the performance of our algorithm by considering the information of the sensors mobility patterns. If we know the mobility pattern, then it will help us a lot in the prediction phase. This will make our prediction more accurate, hence we can achieve high localization accuracy.

We have to think of a solution which will produce accurate localization results and it must be energy and cost efficient as well.

References

[1] 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 Networks,” Proc. Sixth Int’l 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] Y. Shang, W. Ruml, Y. Zhang, and Fromherz MPJ. Localization from mere connectivity. In Proceedings of the 4th ACM Internatioal Symposium on Mobile Ad Hoc Networking and Computing(Mobihoc'03), 2003.

[17] Chen Wang and Li Xiao. Locating Sensors in Concave Areas. In Proceedings of 25th IEEE International Conference on Computer Communications(Infocom'06), pages 1–12, April 2006.

[18] 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.

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

[20] 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.

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

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

[23] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research,Trends and Applications, 2(5):483-502, 2002.