Wireless Sensor Network Location Algorithms

Nuno Ricardo Gago Pinto
nuno.gago.pinto@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2016

Abstract

Localization of sensor nodes is one of the key issues in Wireless Sensor Networks. It is a precondition for a variety of applications, as well as geographic clustering and routing. Previous research on localization sensors presents [1, 2] an integrated implementation for localization algorithm for mobile wireless sensor networks based on the well-known range-free Centroid Location (CL) method, firstly presented by [3] and assumed regularly arranged beacons. The CL method [1, 2] proposes an architecture that splits the original sampling period of the Centroid algorithm into temporal windows in order to maintain a record of past information during movement, allowing for the weighting of the anchors' coordinates. In this work, we investigate the range-free location method and its impacts on localization error, keeping in mind the power consumption during the algorithm development. With our based CL algorithms, we present possibilities to reduce the location error and improve their accuracy using CL. However, since our new approaches introduce more complexity to CL, an analysis of the algorithms usage is considered.

Keywords: mobile sensor network, range-free localization algorithm, centroid location

1. Introduction

Today sensors are everywhere. Smart grid, smart homes, smart water networks, intelligent transportation, are infrastructure systems connected to our world more than we ever thought possible. The common vision of such systems is usually associated with one single concept, the internet of things (IoT), where through the use of sensors, the entire physical infrastructure is closely coupled with information and communication technologies; where intelligent monitoring and management can be achieved via the usage of networked embedded devices. In such a sophisticated dynamic system, devices are interconnected to transmit useful measurement information and control instructions via distributed sensor networks.

A wireless sensor network (WSN) is a network formed by a large number of sensor nodes where each node is equipped with a sensor to detect physical phenomena such as light, heat, pressure, etc. A typical sensor node consists of four basic components: sensing, processing, transceiver (transmitter and receiver) and power units (usually, a battery), as illustrated in Figure 1.

In WSNs the location estimation may enable a myriad of applications such as inventory management, transport, intrusion detection, road traffic monitoring, health monitoring, reconnaissance and surveillance.

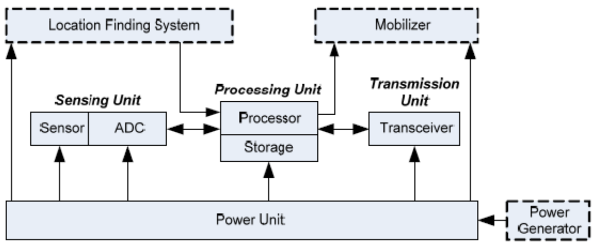

Figure 1: Typical wireless sensor node with their components.

One of the most know and common methods is the commercial positioning system GPS (Global Positioning System) [4] as broadly used in our cars, although GPS based localization estimation has well known limitations in terms of size and high energy consumption, making localization within the sensors network the preferred method over the GPS. Moreover, it cannot be used indoors, because of insufficient GPS coverage.

Therefore, assuming the localization within the wireless sensor network is the preferred method and since the wireless links between the sensors have highly irregular and probabilistic properties. This way, link quality can vary significantly over time, especially in indoor environments due to human activity and multi-path or reflexion effects. Consequently, a typical approach is to assume that only a small number of selected nodes know their exact position a-priori or obtain the position using common positioning systems. Then, all other nodes calculate their position with the help of these beacon nodes.

Localization within the wireless sensor network algorithms estimate the locations of sensors with initially unknown location information by using knowledge of the relative positions for a few sensors and inter-sensor measurements such as distance and bearing measurements.

The accuracy of localization techniques ranges from high precision, commonly based on solving a set of nonlinear equations, to low precision. Localization methods can be divided into coarse-grained and fine-grained algorithms, here we propose a coarse-grained localization algorithm, which needs only a minimum of computations, called Centroid Localization. In the Centroid Localization method, all non-localized nodes calculate their own position based on the centroid of all received positions from the other beacons, where the most reliable ones are the Base Stations, where their position is fixed and known in advance.

The only consideration used by the Centroid Location method to calculate the distance information in Centroid Location is the binary information whether the unknown node is in the communication range of a beacon or not, where the location in known of this beacon in advance. Centroid Location assumes a circular area with the center being the beacon's location as communication range.

This approach has some pitfalls that can occur as the following extremes: the beacons listens to all the possible neighbors, since their radio signal covers all the map, in this case the estimated position for all sensors is in the center of the map; the beacons cannot listen to any neighbor or beacon, meaning they are without any coverage, in a called "dark" area, in this case the sensors cannot calculate their position.

To overcome these extreme case scenarios and even estimate the location of sensors which are out of a coverage from any beacon, a proposed approach, called Modified Centroid Location, that consists on introducing historical information on the calculated position of the beacons and is verified using different densities on the number Base Stations nodes (the nodes which are on fixed and known positions), and different coverage areas of the beacons. It has also been verified the impact of having different weights (linear and exponential) on the historical information for the study performed.

The centroid algorithm can be used to estimate coordinates from the coordinates of the close nodes within the radio signal range. The variant studied by this dissertation considers the nodes are moving within a known area where some nodes are fixed with known positions considered by this as Base Stations providing the coordinates for the rest of the cluster network, thus giving an indication of the direction of movement.

For this dissertation the modified centroid algorithm was implemented in TinyOS and sensors, also known as motes, are simulated in TOSSIM networks (TinyOS mote SIMULATOR). TinyOS is an operating system designed for event-driven sensor networks and has very limited resources. TinyOS is open source and is designed for wireless devices with low power consumption, such as sensor networks, ubiquitous computing and intelligent buildings. TinyOS is often used by the sensors, including in Wireless Sensor Networks, mainly because of the characteristics oriented to a low power usage.

1.1. Outline

In this work, we review the state of art for node localization in Wireless Sensor Networks and show how the localization protocol has the potential to provide the robust solution that is needed. The research challenge that we face is how to obtain a highly accurate node locations in large scale sensor networks deployed in complex environments, at the lowest cost possible. Since our major focus is on the Centroid Location Methodologies it was evaluated the impact when introducing historical information to this method.

In the next Section, we emphasize the main objectives we would like to archive an the methods used as base for our approach as the references analyzed and state of art for node localization in Wireless Sensor Networks algorithms.

In Section 3 is presented our proposed modifications to the Centroid algorithm for both static and mobile networks and described the architecture used to develop and perform our experiments to verify our Centroid Location approach.

In Section 4 we explain the experiments performed and their results in detail.

Section 5 presents the conclusions retrieved from the experiments performed and the we discuss the future developments to improve the results.

2. Background

The work that has been done on sensor localization algorithms can be classified in two main categories: the range-based approaches and the range-free approaches. These approaches differ from each other essentially in the way distance information is obtained. Range-based approaches are based in distance or angle measurements, requiring the installation of specific and expensive hardware (e.g.,

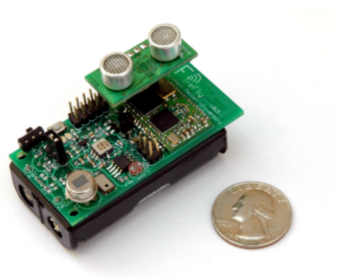

Figure 2: Sensor node example.

directive antennas). Range-free approaches only consider connectivity information between adjacent nodes [5, 6], as in including the Centroid Location [1, 2].

However, there are also hybrid solutions, which combine the advantages of range-based approaches with the advantages of range-free approaches. Moreover, there is either range-based or range-free approaches combined with the use of anchor nodes, based in MDS (Multidimensional Scaling), centralized or distributed, or mobile-assisted [5] and proximity based map (PDM) [6] or MDS and Ad-hoc Positioning System (APS) [7]. These techniques contribute to new directions in Wireless Sensors Networks localization as these schemes give high accuracy in low communication and computation cost.

Furthermore, [1, 2] proposed a method based on the Centroid algorithm that targets mobile networks, where the original sampling period of the Centroid algorithm is divided into temporal windows in order to maintain a record of past information of the node during movement. The selection of the anchor nodes is based on the recorded history, allowing for the weighing of the anchor's coordinates in the localization process. For example, if the number of messages received from an anchor is decreasing that is an indication that the node is moving away from it. Even if the target were mobile networks, with or without control movement, the method also proved to increase the accuracy of the Centroid algorithm in static networks.

2.1. Localization Method

Extensive work has also been done on modeling RF propagation and deploying systems based on received signal strengths [8, 9, 10, 11]. These systems require mapping signal strengths to distances based on known radio characteristics. This type of approach is highly dependent on signal propagation in the environment as well as individual antenna and receiver specific properties.

The research reported in [3] describes a range-free localization technique where each mobile node derives the position by calculating the center of the location of all beacons it hears. If beacons are placed well, localization errors can be decreased [12], but this is not possible in all environments. One option to improve the pure centroid algorithm would be to add an RSSI factor to the location computed.

Utilizing information from the environment would perhaps be the most basic and frequently overlooked source of localization information. Many of these environmental features such as background noises slowly changes over time, hence it becomes significantly difficult to uniquely calculate a large number of locations. Therefore, it requires the use instantaneous time synchronized sensor features to correlate with nearby locations. Hence, we do not require a database and the approach does not suffer from slow changes in the environment over time, and this way reduces the error.

The accuracy of localization techniques ranges from high precision, commonly based on solving a set of nonlinear equations, to low precision. In the methodology proposed by [3] the localization is divided into coarse-grained and fine-grained algorithms and proposed a coarse grained localization algorithm, which needs only a minimum of computations, called Centroid Localization (CL) [4]. In CL, all non-localized nodes calculate their position as the centroid of all received beacons positions. In [12] studied the precision of CL.

2.2. Centroid Localization

The pure CL does not utilize the Received Signal Strength Indicator (RSSI) or any other parameter, indicating the distance between a beacon node and an unknown. The only kind of distance information used in CL is the binary information whether the unknown is in the communication range of a beacon or not. CL assumes a circular area with the center being the beacons location as communication range, i.e. unit disc graph model. Figure 3 depicts the communication ranges of four beacons, arranged as described above. It is shown that thirteen Intersection Areas (IAs) can be distinguished where an unknown node can be localized.

The CL uses the location information of all beacons in range to calculate the position as the centroid of the received beacon positions, as shown in equation 1. Here, $P_i(x, y)$ indicates the position of unknown node i given by its two dimensional coordinates. The known position of beacon j is given by $B_j(x, y)$. The number of beacons which are within the communication range of the unknown node is indicated by m.

$$P_i\left(x,y\right) = \frac{1}{m}\sum_{j=1}^{m} B_j\left(x,y\right) \qquad (1)$$

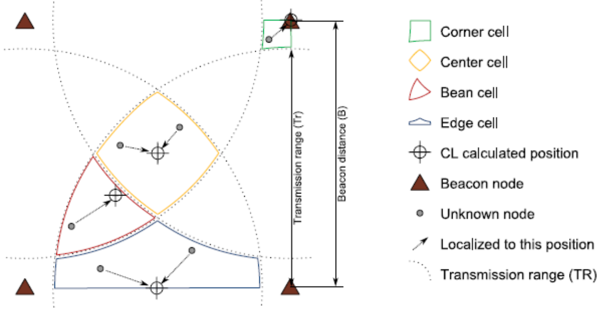

Figure 3: On Centroid Location nodes assign themselves to a cell by listening to messages from the beacons. Then, the node localizes itself at the estimated centroid of the according cell.

A node which is situated within one of the interception areas will calculate its position at one single point, regardless of its exact position within the interception area. For each interception areas exists such an localization point which is the centroid of the beacon positions in range. This behavior leads to a relatively high localization error, given as the Euclidian distance between the exact position of a sensor node and its calculated position. This is the main concept of the model proposed by us.

3. Modified Centroid Location

The CL methodology can only be used when the sensors are static, although when the sensors are moving the CL method must implements samples on the position during the movement. The method proposed in the current dissertation approaches the introduction of historical information for the computation of the position for the motes to overcome the limitations of the pure Centroid Location, referred here as Modified Centroid Location. To verify the approach different densities on the number Base Stations (beacons) nodes (the nodes which are on fixed and known positions broadcasting their location) within the map, and using different coverage areas (signal power), also was included. Furthermore, it has also been verified the impact of having different historical weights values (uniform, linear and exponential) on the study performed.

The test environment used in all the cases was created on a limited area where the nodes perform the movement and calculates the position relative to the beacons with known position which are considered the Base Stations. This limited map area has dimensions 650×650. Also, in all the simulations has been considered 9 nodes moving and on several densities for the Base Stations (number of beacons): 8, 16, 25, 36, 49, 56, 64, 72, 81, 100; using different coverage areas: 100, 200, 300, 400.

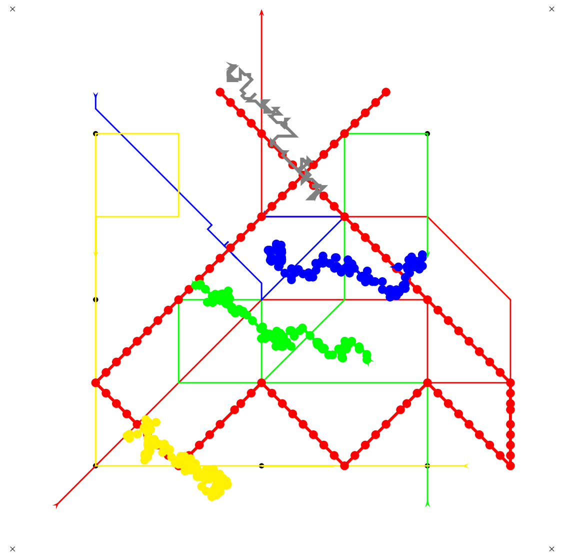

Figure 4: Movement of the motes on the simulation environment: Mote 1 in red (thin); Mote 2 in green (thin); Mote 3 in blue (thin); Mote 4 in yellow (thin); Mote 5 in red (thick); Mote 6 in green (thick); Mote 7 in blue (thick); Mote 8 in yellow (thick); Mote 9 in gray; The $\times$ are the square limits and the $\bullet$ are Base Stations examples.

Furthermore, it has been verified the impact of the historical information on the error of the calculated location and also impacts when implementing the intercommunication between motes making by this way all the nodes beacons.

3.1. Model Used

To perform the simulation 9 different nodes that move within the limited area, were used. This limited area is a square with dimensions of 650×650 where the Base Stations are placed. The Base Stations were equally distributed during the simulations cases, as much as possible, within the square to grant the maximum coverage area possible within the simulation area and this way maximize the efficiency of the expected results. The motes movements path were predefined and can be seen on Figure 4. The same path of the motes was used in all the cases in order to be able to compare always the same movements and reduce the variables changed when comparing the results.

As visible by the map image on Figure 4, some of these motes have a more predictable movement while others motes have a more random movement. This random movement has been created by a script, which gives higher priority to the forward movement, since is assumed that an animal (a possible real utilization for the sensors) moves more often forward then backwards. By these variety on the movements there has been created to certain extent a broad representation for a sensor movement.

Furthermore, as is visible in the map image, the movement of the motes does not have uniform behavior in terms of velocity, meaning that the speed is not the same in all the cases. Moreover, the definition of speed in TinyOS is not clear since the Operating System is event driven, and does not relies on predictable number of CPU cycles, that represents time. Therefore we use displacement between samples for the mote movement, as further explained on Section 4.1, although the displacement gives us the idea of speed during the paths of the different motes.

Our experiments were based on the pure Centroid Model, which has been explained in Section 2.2 and implemented according to Formula 1. For our case, the Modified Centroid Location method introduces the history of the previous positions, which is included in the formula for the Modified Centroid Location. In our experiments three different computing weights of the historical information have been considered: uniform historical weight, linear historical weight and exponential historical weight.

This way, to perform the calculation for mote position, the Modified Centroid Method includes the historical information where the following formulas are used. Here $P_i^{'}(x, y)$ indicates the position of unknown node i given by its two dimensional coordinates. The calculated positions using the pure Centroid where calculated using the Formula 1 and are given by $P_h(x, y)$, where h are previous calculated positions, starting from the oldest position (meaning $P_0(x, y)$ is the oldest position and $P_h(x, y)$ is the last position calculated). This way the more recent positions have higher weights.

Based on the previous, the formula for the case when the historical information is used with uniform weight, is given by:

$$P_i^{'}(x, y) = \frac{1}{h+1} \sum_{k=0}^{h} P_k(x, y) \tag{2}$$

In this case all the previous historical positions have the same weight and even the same as the last computed position.

For the case when the historical information is used with linear weight, the formula becomes the following:

$$P_i^{'}(x, y) = \frac{1}{\sum_{k=0}^{h}(k+1)} \sum_{k=0}^{h} (k+1) \times P_k(x, y) \tag{3}$$

For the linear historical weight, the current computed position has higher weight then the previous and so forth.

In the case when the historical information is used with exponential weight:

$$\begin{aligned} P_i^{'}(x, y) = & \frac{1}{\sum_{k=0}^{h}(k+1)^{k+1}} \times \\ & \times \sum_{k=0}^{h} (k+1)^{k+1} \times P_k(x, y) \end{aligned} \tag{4}$$

The exponential historical weight increases even more the weight of the last position over the previous computed one(s).

In all the cases, the node, which is situated on the interception area, computes the position at one single point, based on Centroid Location method and adds to this calculation the historical positions of the mote.

4. Experimental Evolution

The simulation to perform our experiments was implemented using TinyOS, with the help of TOSSIM that easily allows debugging, testing, and analyzing the algorithms implemented in a controlled and repeatable environment that allowed extended analysis on the results. As TOSSIM runs on a normal PC, the TinyOS code can be tested using debuggers and other regular development tools, such as python and C++.

In our case python [13] was chosen to interact with TOSSIM, due to its solid and powerful functionality and a relative small quantity of lines of code, which makes it less prone to issues and is easy to debug. This way, the python program language allowed to compute and process the mote position information for each single iteration step, making it possible to simulate the movement of the motes relatively to the beacons nodes within the closed environment map area as specified in Figure 4.

Python is part of the automated simulation implemented and, using the interaction with the TOSSIM, allows verification if the mote is within the coverage area of the various Base Stations or beacon motes, as required for the Centroid method. This way, triggering the TOSSIM accordingly and for each interaction allowed the mote to calculate its position based on the communication messages exchanged with the Bases Stations and with the historical data to implement our Modified Centroid method. At the end of the simulation the error is calculated by the difference between the real position where the mote is and his computed position, based on the Centroid Modifed Method as described in Section 3, using the Formulas 2, 3 and 4, considering uniform, linear and exponential historical weight, respectively.

Mote	1	2	3	4	5	6	7	8	9
Max	17.68	17.68	7.07	87.5	24.75	14.14	13.97	12.91	14.02
Min	12.5	12.5	0	0	7.5	1.04	1.04	1.16	1.06
Ave	13.89	12.89	5.83	13.22	17.28	6.42	6.47	6.22	6.6

Table 1: Motes displacement to give idea about the mote velocity.

4.1. Experiments Description

In the simulation model used the Base Stations send multicast (without acknowledge message) messages with their location, which is received by the motes within their coverage area. In our simulation it has not been introduced any field topology effects resulting in variations on the radio noise, only a simple white noise is considered for communication.

During all the simulations performed, the motes movements have always been the same, as described on the Figure 4 from Section 3.1. The tests performed were based on variations of the Base Stations power, to have different coverage areas, and/or Base Stations density in map (the number of the Base Stations), this way makes possible to understand the effect of these changes (power and density of the Base Stations) on the error rate of motes calculated positions using the pure Centroid and the Modified Centroid algorithms when comparing to their real mote position.

To make the motes movement as much random as possible, a script has been developed that generates a pseudo random movement with higher probability for the forward movement, since normally the animals have higher probability to move forward. In Figure 4 can be seen that some of the motes have different velocity for their movements and some of then even have a almost random movement (where the backward movement has less probability to occur). Since TinyOS is an event driven OS, we consider the different movement displacement (the length measure between each step or iteraction) to give an idea about the mote's velocity. The motes displacement can be seen in high level on the Table 1.

The simulations have been executed with different values on the base stations power that covers the an area of 100, 200, 300, 400, and the number of base stations 8, 16, 25, 36, 49, 56, 64, 72, 81, in the square placed at known positions, on the map with 650×650. We have also centroid simulation where it has been included the case when the moving motes also broadcast their position (with the position the mote has calculated for it self, which due to location errors might differ from their real position) and, this way, become beacons also with a power that covers the area of 200. As final it has been simulated the effect of inclusion of the history of the previous 1, 3, 5 positions the motes have calculated, using 3 different weights for the history of the locations. Considering all these possibilities, it gives 504 different test cases for all the 9 motes, 4536 different cases in total to consider, all performed on a automated script to generate the results consolidated on a Excel file and on a database.

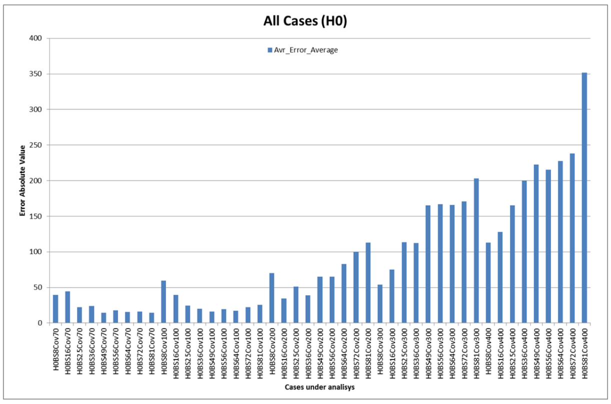

Figure 5: Error evolution for all the cases under study, with 8, 16, 25, 36, 49, 56, 64, 72, 81 number of Base Station beacons with 70, 100, 200, 300, 400 Coverage.

*Note the x axis mentions the case considered, where **H0** means no history (pure Centroid Method), the number after **BS** means the number of Base Stations, and the number after **Cov** means the coverage value considered.*

Due to the different cases used on the movement, the different locations of the Base Stations, and with the different coverages used some times the beacons can get to "dark" areas, areas where the beacons do not receive any broadcast information from any neighbor (either other beacon or Base Station). These cases are also verified and considered on the script developed, in order to include information on the historical data.

As mentioned, to implement all these cases, we have developed python scripts that make use of the algorithm implemented in NesC (that runs in TinyOS). The simulation developed considered the movement of the motes in at least 100 different positions, each of them considered a different interaction of the historical information, and their position retrieved and evaluated in each single one of these positions, by getting the relative position compared to the visible base stations on that particular position using the Modified Centroid methodology as described on the previous chapters.

4.2. Experimetal Results

The graphic on Figure 5 presents the error for all the cases studied (8, 16, 25, 36, 49, 56, 64, 72, 81 number of Base Station beacons with 100, 200, 300, 400 Coverage), just to give an high level idea how the error evolves to guide us to focus on more detail the particular case. These cases are pure Centroid Location Method, meaning no historical information has been considered.

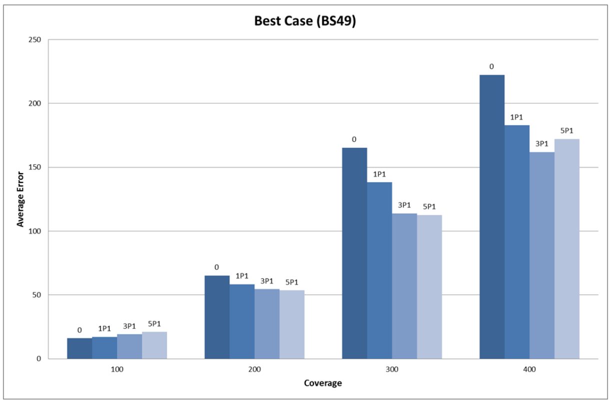

Figure 6: Best Case (49 Base Stations) with linear historical effect for different Base Stations coverage. *Note the **0** means no history (pure Centroid Method), **1P1** means 1 previous historical position, **3P1** means 3 previous historical positions and **5P1** means 5 previous historical position, always using linear weight.*

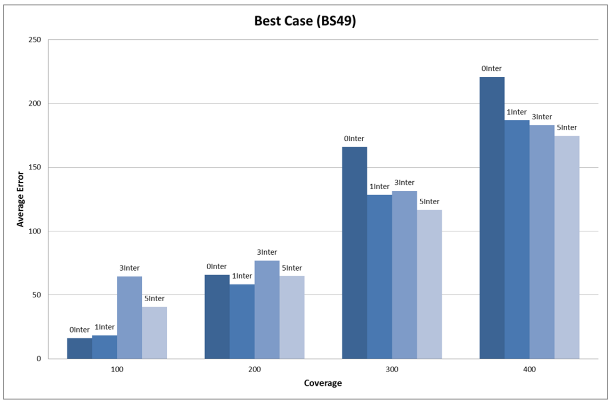

Figure 7: Best case (49 Base Stations) with historical effect including intercommunication between mote, for different Base Stations coverage. *Note the **0Inter** means no history (pure Centroid Method), **1Inter** means 1 previous historical position, **3Inter** means 3 previous historical positions and **5Inter** means 5 previous historical position, always using linear weight.*

From the graphic on Figure 5 is visible the increase of the Coverage (coverage power the Base Station beacons) increases the error. This error increase is caused by the fact the Centroid Method tends to assume the motes are in the center of the square area, by definition. It is also visible that the increase of the density of Base Station beacons does not always reduces the error on the position calculation, showing that there is a optimal number amount of Base Stations which provides a best result. From the graphic, *the best results occurs when the Coverage is 100, and the density for the number of Base Station beacons is 49.*

Considering the best case for the Centroid method, where the density (the number) of Base Stations is 49 within the square area and Coverage of the Base Stations is 100, further analysis has been focused in more detail this scenarios with different perspectives.

At first, it can be seen the effect of our modification of the Centroid Location Method where is taken in consideration linear weighted history information of the previous steps positions. Here we have considered 1, 3, and 5 previous steps and those are mentioned as 1P1 - 1 previous historical step, 3P1 - 3 previous historical steps and 5P1 - 5 previous historical steps. This evaluation can be seen in the graphic of Figure 6.

In this case, the error reduces for higher coverage values on the Base Station beacons and increases for lower coverage because, since in the pure Centroid Method the motes tend to get to the center of the square with higher Coverage of the beacon, by introduction the historical information of previous positions on their calculation, the error is reduced.

The introduction of intercommunication between the moving motes, makes all the nodes become beacons, reducing the error when the coverage of the Base Stations is higher, because the mote calculates a position more distant than the center of the square (where it tends to be in the pure Centroid Location). At the same time, if the coverage of the Base Stations has similar level as the coverage of the (also moving) modes, the error is increased by their negative influence, and gets even more if the history of the previous positions is also taken in consideration, since it increases the cumulative error, as is seen by the graphic on Figure 7.

Looking to these two cases from the opposite perspective (checking the evolution of the coverage), as expected, the error increases with the increase of the coverage. As already mentioned the centroid method intends to position the motes on the center of the map, but the introduction of the historical and intercommunication reduces the error level as is seen by graphics on Figures 8 and 9, respectively.

A different perspective is the evaluation that can be seen when the coverage of the Base Station beacons is fixed to 100, but the density (the number) of Base Station density varies, this effect can be seen on the graphic of Figure 10. This case shows there is an optimal value for the amount of base stations that provides the best result.

This effect happens because the pure Centroid Location methodology makes the motes to be in the center of the visible beacons and, therefore, from one hand, too few Base Stations beacons provide a huge distance in average, since it tends to increases the distance between the beacons and the mote itself; and on the other hand, too many Base Stations beacons in number tends to make the mote to be considered, again, in center of the square.

Based on all this analysis, the optimal case hap-

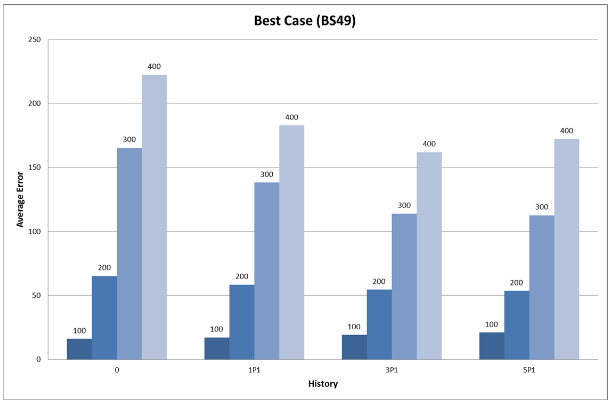

Figure 8: Best Case (49 Base Stations) verifying the coverage effect.
*Note the **0** means no history (pure Centroid Method), **1P1** means 1 previous historical position, **3P1** means 3 previous historical positions, **5P1** means 5 previous historical position, always using linear weight, the values 100, 200, 300, 400 are the different coverages considered.*

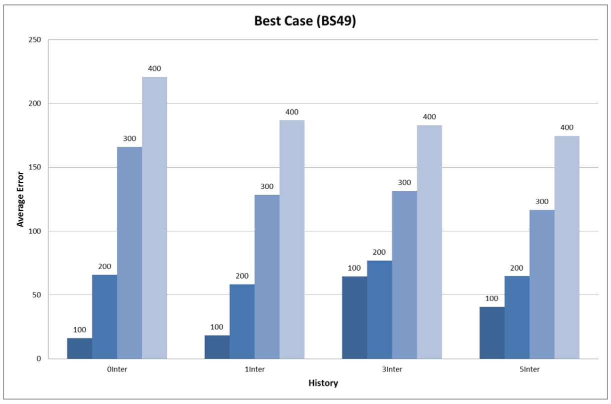

Figure 9: Best Case (49 Base Stations) verifying the coverage effect with intercommunication.
*Note the **0Inter** means no history (pure Centroid Method), **1Inter** means 1 previous historical position, **3Inter** means 3 previous historical positions, **5Inter** means 5 previous historical position, always using linear weight, the values 100, 200, 300, 400 are the different coverages considered.*

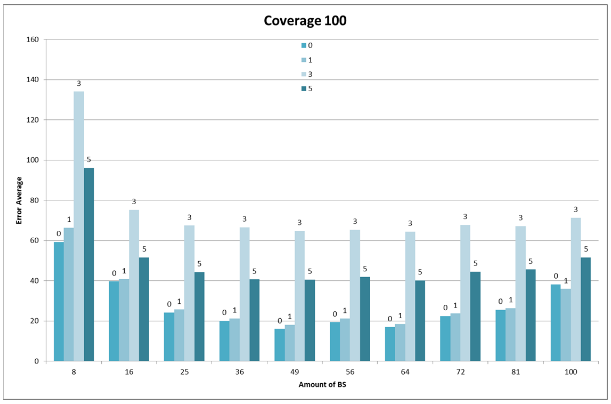

Figure 10: Best case (coverage 100) verifying the Base Stations density.
*Note the **0** means no history (pure Centroid Method), **1** means 1 previous historical position, **3** means 3 previous historical positions and **5** means 5 previous historical position, always using uniform weight.*

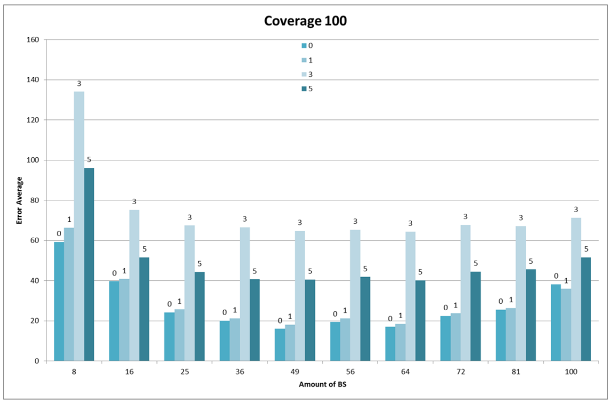

Figure 11: Best case (coverage 100) verifying the Base Stations density with intercommunication.
*Note the **0** means no history (pure Centroid Method), **1** means 1 previous historical position, **3** means 3 previous historical positions and **5** means 5 previous historical position, always using uniform weight.*

pens in the case the number of Base Stations is 49. More or less around this case same optimal amount of 36, 49, 56 Base Stations is the optimal value of density for the other Coverages values of the Base Stations.

By introducing intercommunication between the moving motes, making all the moving motes also become beacons, increases the error. This effect is visible on graphic of Figure 11 and the error gets even worst if the historical information is also considered. This negative impact is explained because it increases the propagation of the error over the next position steps.

Although by introducing weighted historical information the error does not increase that much, as expected, the propagation of the error over the next position steps is reduced by the lower weight of the previous positions, as can be seen on the graphic of

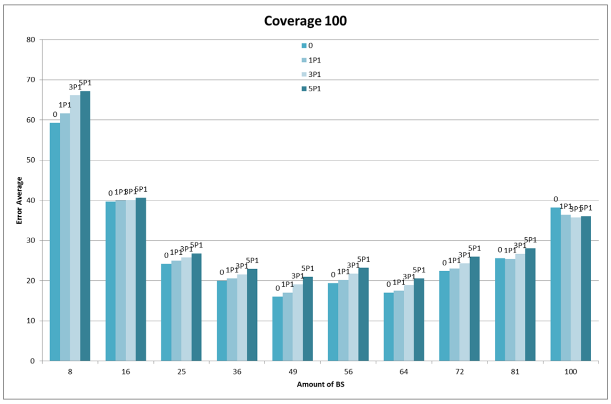

Figure 12: Best Case (Coverage 100) effect of the Weighted Historical Information

*Note the **0** means no history (pure Centroid Method), **1P1** means 1 previous historical position, **3P1** means 3 previous historical positions and **5P1** means 5 previous historical position, always using linear weight.*

Figure 12.

From the examples shown before, it is clear the best results come from the Pure Centroid Method. The introduction of either linear or weighted historical information or even intercommunication between the closest modes has a negative impact on the calculation of their error. The best scenario tested is when using the Coverage 100 on the Base Stations beacons and 49 Base Stations on the density of the number of Base Stations.

5. Conclusions

The objective of this thesis was to enhance and improve the pure Centroid Method during the movement of the beacons on a controlled environment, which was partially accomplished as seen in Section 3, where the best results for the method used occurs when the Coverage is 100 and the number of Base Stations is 49, on a field of 650 × 650, and in this case the error is on average 10%.

It got clear form the experiments performed the introduction of historical information increases in certain extent the error and the historical information has a negative impact, therefore from our point of view should not be used in any of the cases experimented. The reason behind is related to the accumulated error that is kept on several samples.

During this analysis it has been noticed that the Centroid Method used on moving elements brings higher error than the Centroid Method used on static elements. This is caused by the fact that the Centroid Method cannot be used during several iterations since when a later iteration would be calculated the beacon position had changed already, even though the error is in average 10%. This error is calculated when comparing to the size of the square limited area, where the motes are moving.

From the experiments performed the minimal errors occur when the Coverage is 100, and the density for the number of Base Station beacons is 49. In this best scenario case the error has considerable good value and drops to around 2%. During the experiments it must be mentioned that the great power of the language chosen, python, which proved to help the experiments performed that allowed to create graphics, excel files, integrations in SQL, in a automated way. Based on all the experiments performed during the development of this work it was proven the Centroid Method has the best performance when is used on its own, without including any other experiments, such as introduction of historical information or intercommunication with neighbors.

5.1. Future Work

Since in our believe the major reason of the high level value is caused by the fact the beacons are moving and this way it cannot be implemented several interpolations of the pure Centroid Method, it would be suggested to understand the impact of the speed when compared to the number of samples when implementing more interpolations on the pure Centroid Method, since the pure Centroid Method seams to have good results when the beacons are stopped. An example of a practical usage of this work would be to know the location of cow on the field, since the cows rarely move backwards; a second future possible approach is to include vectorial information on the movement to reduce the error of the results.

Acknowledgements

I would first like to thank my thesis advisor Professor José Costa of the Instituto Superior Técnico at Lisboa University. The door to Prof. Costa office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to acknowledge Professor José Monteiro of the Instituto Superior Técnico at Lisboa University as the contributor to finalize this thesis, and I am gratefully indebted for his very valuable comments on this thesis.

Finally, I must express my very profound gratitude to my parents and to my partner Telma for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you!

My acknowledgment would never be complete

without the special mention of my two lovely children Duarte and Alice for those important moments on your growth during the years of study. Many thanks!

References

[1] J. Monteiro, L. L. de Oliveira, J. B. Martins, and G. Dessbesell. Hardware implementation of a centroid-based localization algorithm for mobile sensor networks. *Symposium on Circuits and Systems*, pages 2829–2832, May 2011.

[2] J. Monteiro, L. L. de Oliveira, J. B. Martins, and G. Dessbesell. Centroidm: a centroid-based localization algorithm for mobile sensor networks. *Symposium on Integrated Circuits and Systems Design*, pages 204–209, September 2010.

[3] N. Bulusu, J. Heidemann, and Estrin. D. gps-less low cost outdoor localization for very small devices. *IEEE PerCom*, 2000.

[4] J. D. Gibson. *The Mobile Communications Handbook*. IEEE PerCom, 1996.

[5] J. Bachrach and C. Taylor. *"Localization in Sensor Networks," in Handbook of Sensor Networks: Algorithms and Architectures.* I. Stojmenovic, Ed., 2005.

[6] King-Yip Cheng, King-Shan Lui, and Vincent Tam. Localization in sensor networks with limited number of anchors and clustered placement. In *Wireless Communications and Networking Conference, 2007*, pages 4425–4429. IEEE WCNC 2007, March 2007.

[7] A. A. Ahmed, H. Shi, and Y. Shang. Sharp: A new approach to relative localization in wireless sensor networks. In *IEEE ICDCS*. IEEE ICDCS 2005, 2005.

[8] J. Krumm, L. Williams, and Smith G. Smartmovex on a graph - an inexpensive active badge tracker. *UbiComp*, 2002.

[9] T. Christ and P. Godwin. A prison guard duress alarm location system. *IEEE ICCST*, 1993.

[10] A. Smailagic, J. Small, and D. P. Siewiorek. Determining user lcation for context aware computing through the use of a wireless lan infrastructure. *IEEE*, 2000.

[11] M. Youssef, A. Agrawala, and A. U. Shankar. Wlan location determination via clustering and probability distributions. *IEEE PerCom*, 2003.

[12] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann. Weighted centroid localization in zigbee-based sensor networks. In *IEEE International Symposium on Intelligent Signal Processing*, Madrid, October 2007. WISP 2007, IEEE PerCom.

[13] Python is a multi-paradigm programming language. https://www.python.org.