Walking GPS: A Practical Solution for Localization in Manually Deployed Wireless Sensor Networks

Radu Stoleru, Tian He, John A. Stankovic
Department of Computer Science
University of Virginia
{stoleru, tianhe, stankovic@cs.virginia.edu}

Abstract

In this paper we present the design, implementation and evaluation of a simple, practical and cost effective localization solution, called Walking GPS, that can be used in real, manual deployments of WSN. We evaluate our localization solution exclusively in real deployments of MICA2 and XSM motes. Our experiments show that 100% of the deployed motes localize (i.e,. have a location position) and that the average localization errors are within 1 to 2 meters, due mainly to the limitations of the existing commercial GPS devices.

1. Introduction

Wireless Sensor Networks (WSN) have generated a lot of interest recently. The interest, initially in the academic community, was due to the challenging research problems posed by these devices with very limited resources (e.g. memory, processing power, radio bandwidth, energy). More recently, due to the tremendous potential of WSN in military, civil and industrial applications, real deployments of these networks have become imminent.

Many elegant and clever solutions have been proposed and evaluated in simulators and real system deployments, for several of the problems present in the WSN. Among these problems are energy conservation, efficient data placement and aggregation, programming paradigms and topology control. However, despite the attention it has received, accurate localization is a problem that remains unsolved in a real, ad-hoc deployment, without sophisticated, expensive hardware. In this paper we present a solution to this problem, when manual deployment is an option.

In many applications it is envisioned that WSN will be deployed from Unmanned Aerial Vehicles. In the meantime, manual deployments have been prevalent and the employed localization solutions have used some variant of associating the sensor node ID with prior knowledge of that ID's position in the field.

We propose in this paper a solution, called Walking GPS, in which the deployer (either person or vehicle) carries a GPS device that periodically broadcasts its location. The sensor nodes being deployed, infer their position from the location broadcast by the GPS device.

The main contribution of this paper is that we present the design, implementation and the real world evaluation of a solution for the localization of wireless sensor nodes that are deployed manually in the field. Our solution is simple, cost effective and has very little overhead. Despite the simplicity of the idea, many system design and implementation issues had to be addressed in order to make the solution work and be efficient. We further hypothesize that the lessons learned from our experience can be extended to aerial deployments, for an initial, coarse localization.

The rest of the paper is structured as follows. In the second section we present related work, emphasizing research that focused on real system implementations. We present the system design and architecture of our Walking GPS localization solution in section three and the implementation of the system in section four. We present our extensive experimental results in section five and conclude in section six by summarizing the main contributions and propose future extensions of our solution.

2. Related Work

Recent developments, driven by both industrial and academic research, continuously expand the applicability of sensor networks at an unprecedented momentum. The inherent interaction between sensor networks and the physical world makes location-awareness one of the essential services for many emerging applications such as location-based directory service (e.g., GHT) [1] and entity tracking [2]. In response, many algorithms have been proposed to address the localization problem in sensor networks.

Among these solutions, some of them are designed under certain assumptions and mostly evaluated in simulation environments. For example, the Amorphous positioning algorithm proposed in [3] uses offline hop-distance estimations and multilateration to estimate nodes' locations, assuming an isotropic RF radio. The APIT positioning algorithm [4] is a scheme in which a node infers its position based on the possibility of being inside or outside of a triangle formed by any three anchors . In this scheme, more powerful anchors, that cover the entire deployment area, are required. In the Received Signal Strength Indicator (RSSI) techniques (RADAR [5] and SpotOn [6]), either theoretical or empirical models are used to translate signal strength into distance estimates. These approaches show promising results in simulation and controlled laboratory environment. However, in practice, many empirical studies [7] [8] [9] demonstrate that in most environments, RF radio is not isotropic and there is, actually, no connection between the signal strength degradation and the distance an RF signal travels.

Another set of solutions use Time of Arrival (TOA) [10] and Time Difference of Arrival (TDOA) [1] [11] [12] techniques to obtain pair-wise distances. These techniques demonstrate high accuracy in localization in real deployment. However, they require extensive infrastructure support, such as GPS in [10], high per-node cost as in Cricket [11] [12] and AHLos [13]. In addition, the limited range of ultrasound (normally 7 – 10 meters) used in TDOA, imposes the requirement of a high-density anchor nodes, which makes the overall system cost for localization prohibitively high for large scale sensor networks.

Recently, mobile robots have been utilized as an effective instrument to localize sensor nodes [14] [15]. Although the ideas are similar to those in this paper, due to the significant discrepancy in the deployment configuration, their designs currently require a fine-tuning of parameters, in order to achieve a high accuracy. This tuning may prove to be problematic for a non-expert.

In view of various limitations exhibited by current localization schemes, we aim to find a practical solution that not only provides high localization accuracy in a running system, but also requires very low system cost.

3. System Architecture

In this section, we present the architecture of our system, and the design decisions we encountered. We support our decisions with background material and a description of the internals of our software components.

An alternative to the Walking GPS localization scheme is enabling each sensor node with GPS capabilities. This monolithic solution is both expensive and inefficient.

In the Walking GPS architecture, however, the system is decoupled into two software components: the GPS Mote and the Sensor Mote. A UML deployment diagram is shown in Figure 1.

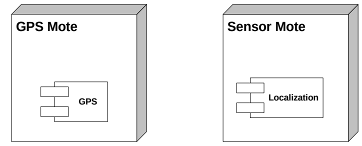

Figure 1. Software components for the Walking GPS scheme

The GPS Mote runs on a MICA2 mote. The mote is connected to a GPS device, and outputs its location information at periodic intervals. The Sensor Mote component runs on all sensor nodes in the network. This component receives the location information broadcast by the GPS Mote and infers its position from the packets received.

This architecture enabled us to push all complexity derived from the interaction with the GPS device to a single node, the GPS Mote, and to significantly reduce the size of the code and data memory used on the sensor node. Through this decoupling, a single GPS Mote is sufficient for the localization of an entire sensor network, and the costs are thus reduced.

A relatively simple design for the GPS Mote would have been to periodically broadcast the actual GPS location received from the GPS device. A GPS location is represented by a latitude and a longitude, which are angular measures from the Equator to North or South, and Prime Meridian to East or West, respectively. Due to the relatively small size of a sensor network (hundreds to a few thousand meters), the use of global (i.e. GPS) coordinates is very inefficient. The inefficiency stems from the size of the packets used for passing location information – a significant portion of the location is likely to be the same for all sensor nodes – as well as from the computational costs encountered when aggregating data, e.g., triangulation of several GPS coordinates for positioning a target.

In order to reduce the overhead incurred when exchanging data containing global GPS coordinates (the GPS coordinates take 11 bytes out of 29 bytes, which is the payload size of a TinyOS packet), we decided to use a local, Cartesian, coordinate system. This local coordinate system of reference, which uses linear units, is better suited for WSN, than a global coordinate system.

A local coordinate system is built from a global system, that uses GPS coordinates, in the following way: the local system of reference has an origin (called a Reference Point) specified in terms of global coordinates (GPS coordinates). The distance between this Reference Point (with coordinates λ_1 and φ_1) and another point, with a GPS location specified by λ_2 and φ_2 , can be computed as follows [16]:

$$\text{Distance} = \sqrt{\left(F_{lat} * \left(\varphi_1 - \varphi_2\right)\right)^2 + \left(F_{lon} * \left(\lambda_1 - \lambda_2\right)\right)^2} \qquad (1)$$

where:

$$F_{lat} = \frac{\pi}{180}\left(\frac{a^2 b^2}{(a^2\cos^2\varphi + b^2\sin^2\varphi)^{3/2}} + h\right) \quad (2)$$

$$F_{lon} = \frac{\pi}{180}\left(\frac{a^2}{\sqrt{a^2\cos^2\varphi + b^2\sin^2\varphi}} + h\right)\cos\varphi \quad (3)$$

are conversion factors that represent the distances for 1 degree change in latitude and longitude, respectively. The unit of measure is meter/degree. The parameters in the above formulas are: a=6,378,137 meters, b=6,356,752.3142 meters and h is the height over the ellipsoid. The influence of h on the conversion factors is minimal and a value of 200 meters is assumed. The X and Y coordinates of the point with a GPS location specified by λ_2 and φ_2 are given by the two additive terms in (1). The Y-axis of the local coordinate system is oriented in the North/South direction and the X-axis in the East/West direction. All variables specified in (1), (2) and (3) (i.e., λ, φ and h) can be directly obtained from a commercial GPS device. The result of our design is that the GPS Mote transforms the global coordinates received from the GPS device into local coordinates and broadcasts these local coordinates.

The localization scheme that makes use of the Walking GPS solution has two distinct phases:

- the first phase is during the deployment of the sensor nodes. This is when the Walking GPS solution takes place. The carrier (soldier or vehicle) has a GPS-enabled mote attached to it; the GPS-enabled mote periodically beacons its location; the sensor nodes that receive this beacon infer their location based on the information present in this beacon.
- the second phase is during the system initialization. If at that time, a sensor node does not have a location, it will ask its neighbors for their location information. The location information received from neighbors is used in a triangulation procedure by the requester, to infer its position. This second phase enhances the robustness of the scheme.

The internals of the two components running on the GPS Mote and on the Sensor Mote, are described further in the following subsections.

3.1. GPS Mote

A GPS Mote without a Reference Point is in an Uninitalized state. No messages are sent by the GPS Mote, as long as it is in this state. A Reference Point can be obtained either through radio communication, or from flash memory. Once a Reference Point is obtained through radio, it is also stored in the flash memory. A GPS Mote with a Reference Point is in an Initialized state.

The state transition diagram for the GPS Mote is shown in Figure 2, below.

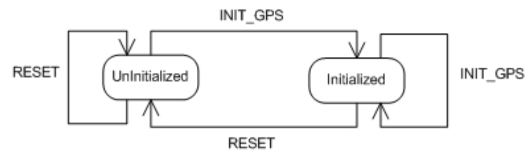

Figure 2. State transition diagram for the GPS Mote

The initialization of the GPS Mote is done by sending a packet of type INIT_GPS, with a format shown in Table 1. In addition to the latitude and longitude of the Reference Point, two more parameters are sent: the sending power and the sending period.

Table 1. INIT_GPS message format

Reference Point Latitude	Reference Point Longitude	Sending Power	Sending Period
4 bytes signed integer	4 bytes signed integer	1 byte unsigned	2 bytes unsigned integer

The sending power is used for limiting the communication range of the GPS Mote. The intent is that only the motes in the vicinity of the deployer receive the localization information. The sending period describes the frequency with which the localization packets are sent. This frequency is correlated with the speed of deployment.

The format for the latitude and longitude of the Reference Point is an optimized for space version of the more general d°m'ss.ss" format (degrees, minutes, seconds). The unit for the 4-byte format that we use is the 1/1000 of a second and the formula for computing it, is the following:

$$coord = d * 36 * 10^5 + m * 6 * 10^4 + ss.ss * 10^3$$

When an INIT_GPS packet is received, its information is stored in the flash memory, to be available after a system reset. A packet of type RESET puts a GPS Mote back into an Unitialized state, and erases the portion of the flash memory that stores the Reference Point, Sending Power and Sending Period.

The GPS Mote sends location information by broadcasting an INIT_LOCALIZATION packet, with the format shown in Table 2.

Besides the X and Y coordinates, the GPS Mote broadcasts the GPS coordinates of the Reference Point as well. We chose this design because the base stations are also deployed using the Walking GPS solution.

Table 2. INIT_LOCALIZATION message format

Reference Point Latitude	Reference Point Longitude	X Coord.	Y Coord.
4 bytes signed integer	4 bytes signed integer	2 bytes signed integer	2 bytes signed integer

The base stations can be queried for the reference point of the local coordinate system, hence that is the reason for broadcasting it as part of the INIT_LOCALIZATION packet. Thus, the Reference Point information, present in the INIT_LOCALIZATION packets, is only used by base stations.

3.2. Sensor Mote

The Sensor Mote can also be in one of two states: Initialized (if location information is present) or Uninitialized. The state transition diagram for the Sensor Mote is shown in Figure 3, below:

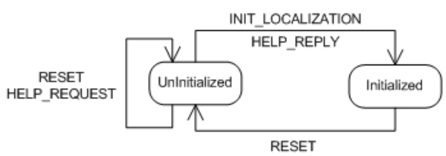

Figure 3. State transition diagram for the Sensor Mote

A Sensor Mote can become Initialized in one of two circumstances: a) if it receives an INIT_LOCALIZATION message (event that can happen during the Walking GPS phase of our localization scheme) or b) if the sensor node sent a HELP_REQUEST message and neighboring nodes reply with HELP_REPLY messages (event that can happen during the system initialization phase). If a sensor node enters the system initialization phase and it does not have a location (which was supposed to have been acquired as part of the Walking GPS), then the sensor node broadcasts a packet of type HELP_REQUEST. Neighboring nodes that have a location respond by broadcasting packets of type HELP_REPLY, with the format shown in Table 2. A sensor node that sent HELP_REQUEST messages, stores the HELP_REPLY responses in a buffer and computes its own location at the centroid of the locations received from its neighbors.

After obtaining the location information, a sensor node stores its location in flash memory, to be available after a system reset. If a RESET packet is received the location information is erased from the flash and the sensor node becomes Unitialized.

For Sensor Motes, we provided support for two methods of deployment:

- ON_AT_DEPLOYMENT, or the first deployment type, is the type of deployment in which a sensor node is powered on right before it is deployed (the deployer reaches the point of deployment, then turns on the power and places the mote on the ground).
- ON_ALL_THE_TIME, or the second deployment type, is the type of deployment where the sensor node is powered all the time. This second scenario is more convenient for the deployer (no mechanical switches to be turned on/off), and more likely to be used in a real world deployment, however it is more challenging, since the sensor node needs to infer its location from a set of beacons containing, most likely, different locations.

The complexity associated with the two deployment types, is different. For the case in which the sensor nodes are turned on right before the deployment, the solution is trivial. The first received INIT_LOCALIZATION packet provides the actual location. When the sensor nodes are on all the time, they need to infer from the RSSI value the time when they were deployed. In order to achieve this, we use a circular buffer with a window size of 4, which stores location information received in the INIT_LOCALIZATION messages. Two configurable parameters are used: a minimum RSSI value (called Lower Bound RSSI), below which the sensor mote no longer accepts localization packets (this reduces the risk of receiving messages from GPS Motes that are far away) and an interval RSSI (called Delta RSSI). We employ a moving average computation for the packets present in the circular buffer. A subsequent INIT_ LOCALIZATION message is accepted if its RSSI is in the interval [Avg. – Delta RSSI, Avg. + Delta RSSI]. In order to obtain its location, a sensor node goes back in the circular buffer, a configurable number of entries (currently two) and retrieves the location present in that entry in the buffer.

4. System Implementation

The Walking GPS localization scheme requires that the deployer has a GPS Mote attached to it. We built a prototype, called the GPS Mote assembly, that can be worn during the deployment. This prototype consists of a GPS device mounted on top of a bicycle helmet. The GPS device is connected through and RS232 cable to the GPS Mote that is attached with a velcro to a wristband. Figure 4 illustrates the prototype.

For the GPS device, we used the eTrex Legend [17] device produced by Garmin. The GPS device is WAAS (wide-area augmentation system) enabled, and it provides updated location information with high accuracy (error < 3 meters), at a rate of 1Hz. Our choice to use a commercial GPS device for experiments was due to its ease of use and seamless integration. We also implemented a miniaturized version of the GPS Mote using the MTS420CA sensor board from Crossbow Inc.

[18]. We have not performed extensive performance evaluation experiments to assess the accuracy of location information obtained from the miniaturized version of the GPS Mote. For our deployment method (used in the Walking GPS scheme) the miniaturization was not an important factor.

Figure 4. GPS Mote assembly

We performed tests on MICA2 motes and on the newer generation of motes, called XSM, from Ohio State University and Crossbow [19]. We used both platforms as Sensor Motes. For the GPS Mote, we only used MICA2 motes.

Several parameters were used for performance tuning. For the GPS Mote, the sending power of the radio was set for all experiments to 0xA and the rate at which the beacons were sent was set to 3Hz, unless the experiment (described in the following section) indicates a different value. For the Sensor Mote, two parameters were used: the Lower Bound on the RSSI value (packets with a signal strength smaller than the threshold were discarded), and the Delta RSSI, both explained in the preceding section. We provide the values of these parameters in an ADC_Count unit, which is a measure of the signal strength associated with a received radio packet. Both, MICA2 and XSM motes use a Chipcon CC1000 radio and the formulas for converting the ADC_Count unit to S.I. unit (dBm) is [20]:

$$V_{RSSI} = V_{battery} \times ADC\_Counts/1024$$
$$RSSI_{dBm} = -51.3 \times V_{RSSI} - 49.2$$

The values of the Lower Bound RSSI and Delta RSSI parameters, in ADC_Count units, are given in Table 3.

We implemented the Walking GPS localization scheme in nesC (approximately 1500 lines of code) for the TinyOS operating system. For the GPS Mote, the total code size was approximately 17KB and the data size was 595 bytes. The code size for the Sensor Mote module was 972 bytes and the data size was 117 bytes. The code has been released and it is available at [21] [22] .

Table 3. Sensor Mote parameter values used in experiments

	Lower Bound RSSI	Delta RSSI
MICA 2	200	50
XSM	35	5

5. Performance Evaluation

In this section we present the experimental results obtained from the evaluation of the Walking GPS localization scheme. In order to better understand the performance of each individual component of our system, we first evaluated each component separately. In the second part of our experiments, we evaluated the system in its entirety.

5.1. GPS Mote

The purpose of this experiment was to evaluate the accuracy of the GPS device and the precision of the transformation performed on the global GPS coordinates, in order to obtain the local coordinates (i.e., Cartesian). In this experiment we did not evaluate the Sensor Mote part of the Walking GPS scheme. The results are shown in the Figure 5.

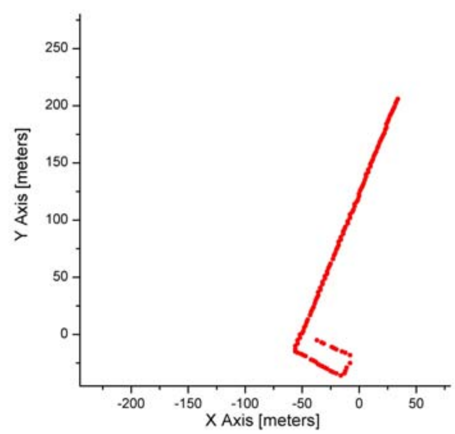

Figure 5. Walking in a straight path, then a loop

The experiment consisted in walking with the GPS Mote assembly in a straight alley, for approximately 215 meters, then turning left and following a U-shape path, going around a parking lot. The walk was performed at a speed of about 1.3 meters/sec and the rate with which the initialization beacons were sent was 2Hz. The average accuracy, as indicated by the Garmin GPS device, was approximately 4 meters. The beacons were received by a base station attached to a laptop, which were carried during the entire experiment. The starting point was in the upper right part of the graph:

In Figure 5, as well as in the figures that follow, the Cartesian coordinate system is aligned to the North-South and East-West directions. The X-axis represents the East-West direction and the Y-axis represents the North-South direction.

To further verify the validity of the experimental data, we superimposed the trajectory obtained in the experiment onto an aerial map [23] of the area where the experiment took place. The result in shown in Figure 6. A very good match can be observed, between the experimental data and the reality in the field.

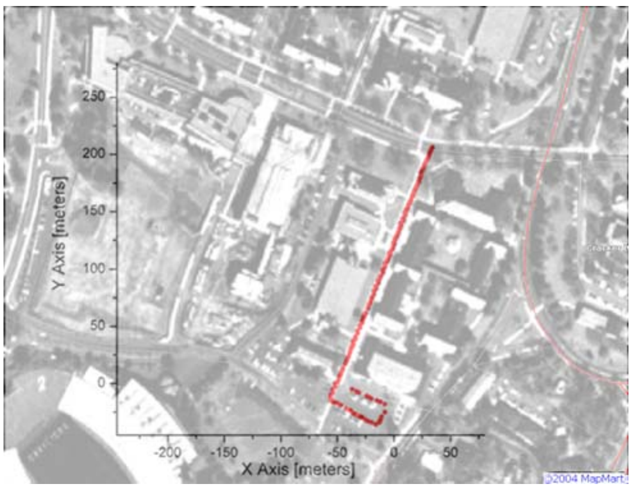

Figure 6. Super-imposing Figure 5 on an aerial map

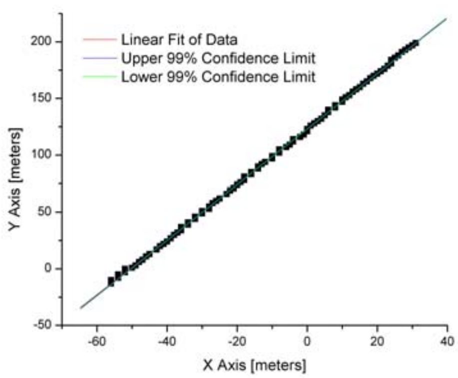

Figure 7. Linear fit for a portion of the path, shown in Figure 5.

For estimating the average error present in the localization, the portion of the experimental data, shown in Figure 5, that represented the walking in a straight path (not including the U-shape), was used in a regression linear fit. The result of the linear fit as well as the upper and lower 99% confidence limits, are shown in Figure 7.

From the linear regression analysis we obtained a coefficient of determination (R^2) of 0.99952, i.e., 99.9% of the variability can be explained by the linear model. The value for Mean Square Error (MSE) was 1.782m^2. This indicates an approximate error in localization on the order of 1 – 1.5 meters.

5.2. Sensor Mote

The purpose of this experiment was to evaluate the performance of the second component of the Walking GPS scheme, namely the Sensor Mote. In this experiment we did not evaluate the GPS Mote part of the Walking GPS scheme.

For this experiment we used 30 MICA2 motes as Sensor Motes. The test was performed outdoors. For this experiment, we configured the GPS mote to not use the actual readings of the GPS device. In this mode, called the "debug" mode, the X-coordinate is incremented with each broadcast, and the Y-coordinate has a value of 0 at all times. This allowed us to better identify the exact packet that is used by the Sensor Mote for inferring its location.

We performed the experiment 30 times, for each receiving mote. The beacons, containing location information, were sent at a rate of 1Hz. We used a lower rate because we wanted to control the timing when each mote was deployed. Using a lower rate enabled us to better synchronize the actual deployment with the occurrence of a particular beacon. The Sensor Motes were deployed at the same physical location, at approximately the time when the 18th initialization beacon was sent. In Figure 8, we show the number of motes for all X-coordinates that were obtained during the experiment.

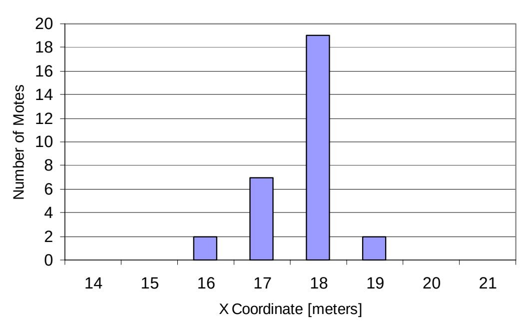

Figure 8. Number of motes for each X-coordinate

From Figure 8 it can be seen that, with a relatively good accuracy, the sensor motes inferred the correct location.

There is little variability in the value of the X-coordinate and this can be explained by our deployment process – we attempted to deploy each mote at the correct beacon number (the 18th) – however this procedure was not very rigorous. The small variability in the X-coordinate values, present in Figure 8, indicates a relatively good control in choosing which packet (among the ones present in the circular buffer) is used for determining the actual location. Empirical evaluations can provide insights regarding the length of the circular buffer and which entry from the buffer would best approximate the actual deployment location. For these evaluations, various deployment scenarios need to be considered (e.g., the speed with which the deployer moves, the rate at which the GPS device computes a new location and locations are broadcast, type of motes used in the deployment).

5.3. Deployment

The Walking GPS solution was evaluated in an open field, as shown in Figure 9. For an easier estimate of the localization error, we marked a 6x5 grid on the ground and we deployed the sensor motes in this grid. We want to emphasize the fact that the deployment being done in a grid was not used in any way during our localization. A deployment in any other regular geometric shape could have been performed. We used a grid because it was easy to create and it was easier to visually assess the performance.

Figure 9. Field where the experiments took place

We performed an experiment to verify, using only the GPS Mote, the grid marked on the ground. This experiment was similar in nature with the one shown in Figure 5. The walk was done on the grid marked in the field. This time the walk with the GPS assembly was for shorter distances and alternating directions, at a speed of about 1 meter/second. The localization packets were stored on a laptop, carried during the walk. The result of the experimental evaluation is shown in Figure 10.

The starting point was the upper-left corner, approximately at the (0, 0) coordinate. The path shown in Figure 10, was followed in all the following experiments as well. From Figure 10 we can observe a good fit between the experimental data and the path that was followed. In the experiments that follow, we provide numeric localization errors by performing a manual best fit of a strict grid with unit 10 meters, to the experimental data. The average localization error is defined as follows:

$$error = \frac{\sum_{i=1}^{N} \sqrt{(x_i - x_i^{grid})^2 + (y_i - y_i^{grid})^2}}{N}$$

where (x_i, y_i) is the coordinate obtained experimentally for the *i-th* mote, (x_i^{grid}, y_i^{grid}) is the coordinate of the *i-th* mote on the fitted grid, and N = 30 in our experiments.

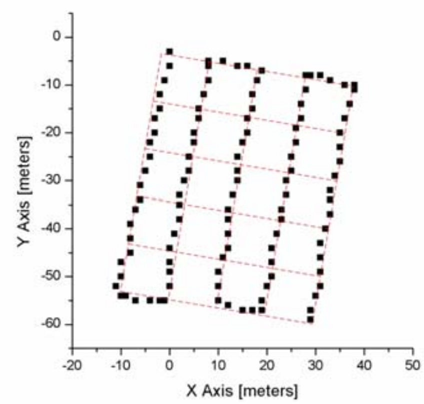

Figure 10. Walk with the GPS Mote along the grid

It is critical in understanding the following experimental results to note that the average location errors are not with respect to the "ground truth" location, but rather are relative to the known geometry of the deployment grid.

5.4. Integrated System - First Deployment Type

In this experiment we evaluated the entire system, consisting of 30 MICA2 motes that were deployed in the aforementioned grid. Each mote was turned on at its place of deployment, right before being deployed. This was described above as the First Deployment Type. The first localization packet provided the location of the receiving mote. The experimental results are shown in Figure 11.

The average localization error obtained from fitting a grid to the experimental data is 0.8 meters with a standard deviation of 0.5 meters. From Figure 11, as well as from the numerical results of the localization error, it can be observed a remarkably good fit. In this deployment type the errors are only due to the estimation of the global coordinate, done by the GPS hardware.

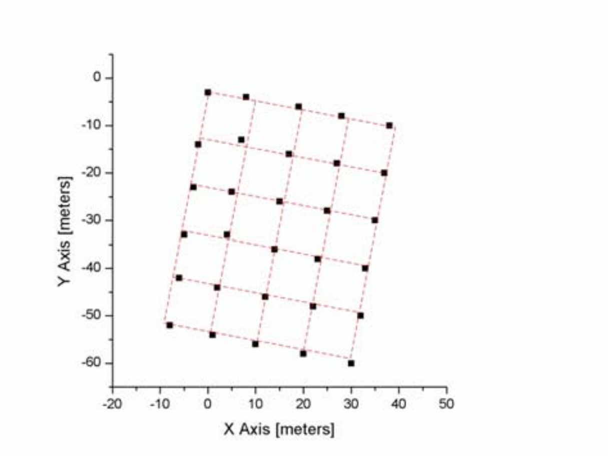

Figure 11. System deployment using the first deployment type

5.5. Integrated System - Second Deployment Type

The purpose of this experiment was to evaluate the performance of the second deployment type. In this experiment the sensor nodes were powered on all the time. The experimental results are depicted in Figure 12.

The localization error obtained from fitting a grid to the experimental data is 1.5 meters with a standard deviation of 0.8 meters. The average localization error is larger than in the experiment where the nodes were turned on right before deployment.

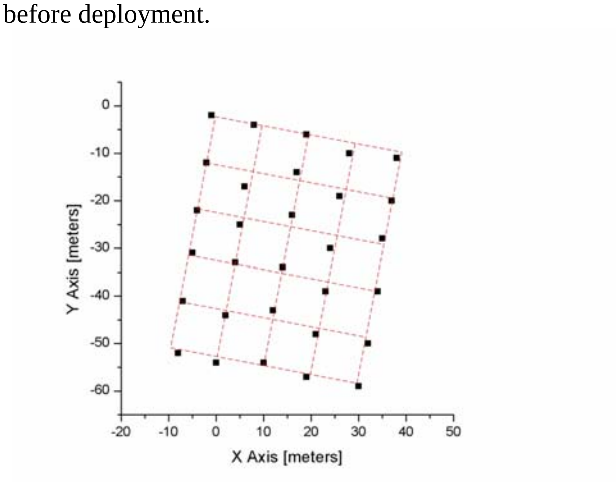

Figure 12. System deployed using the second deployment type

The less accurate location is due to the incorrect inference of the exact moment a sensor node was deployed. The same effect was observed and depicted in Figure 8, where there was some variance in the value of the X-coordinate. Nevertheless, an average localization error of only 1.5 meters is very good for deployments of sensor nodes not equipped with specialized hardware.

5.6. Integrated System – Dual Deployer

The purpose of this experiment was to evaluate the performance of the Walking GPS localization scheme when using two commercial GPS devices (the same model however). A GPS device, as any other hardware device is dependent on calibration. Even after stringent calibration procedures, some variability in the indicated location is expected. From the direct reading of the global GPS location as shown by two GPS devices positioned next to each other, differences on the order of 1/1000 of a minute and sometimes even 1/100 of a minute, were observed. It was anticipated that these differences will contribute to an even larger localization error.

The deployment in this experiment was done along the length of the grid field (lines containing 6 motes). Three of the vertical lines (the middle and the two extreme ones) were deployed using one of the GPS devices, the other two vertical lines were deployed using the second GPS device. The experimental results are shown in Figure 13.

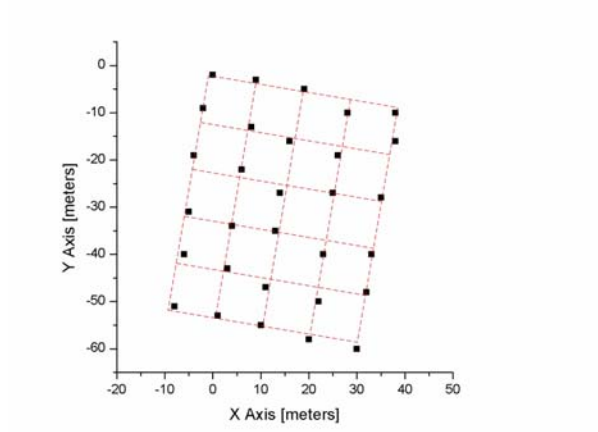

Figure 13. System deployment using two GPS devices

The localization error obtained from our fitting of a grid to the experimental data is 1.6 meters with a standard deviation of 0.9 meters. In this deployment scenario, the average localization error is the largest. In addition to the errors encountered in previous experiments, here, the GPS device calibration has an additional contribution. When comparing the results of this experiment with the previous one, in which only one GPS device was used, it can be observed that the effect the device calibration has on location error was relatively small, of about 0.1 meters.

6. Tracking Application

The proposed Walking GPS localization solution has been integrated and tested with a target tracking application [2] developed in our research group. A screenshot of the tracking application is shown in Figure

14. The experiment used 32 XSM motes, deployed in a parking lot, approximately 8 meters apart (spacing was not rigorous).

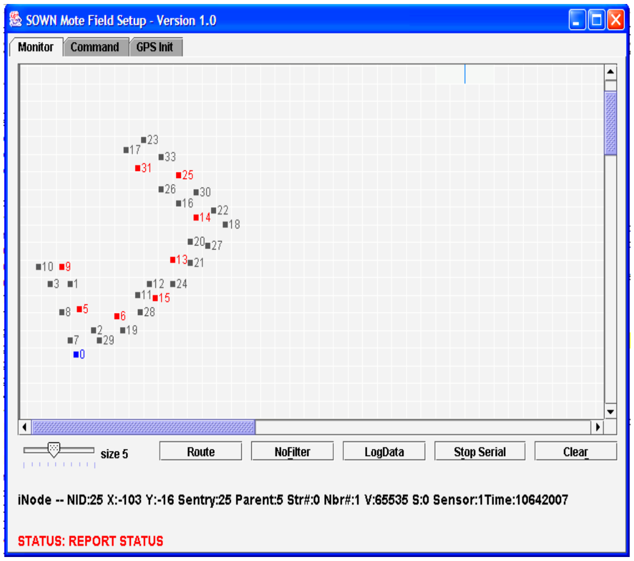

Figure 14. Target tracking application using Walking GPS

We observed a very good match between the locations reported by the tracking application and the real deployment of the nodes. We do not provide exact localization errors, due to the irregular deployment (the spacing was only approximate).

7. Conclusions and Future Work

In this paper we presented the design, implementation and the evaluation of a localization solution that can be used in situations in which WSN are deployed manually. The method to deploy sensor nodes manually is currently used in several projects and there are scenarios of real system deployments, where the manual deployment is, still, the only viable solution. Our proposed solution has very little overhead and it is cost effective.

The experience from the development of the current system can be further used in future research that will address the aerial deployment. Considering the sensor deployment rate, deployment altitude, sensor trajectory and the actual location at the beginning of deployment, some coarse location information can be inferred using our solution, giving a starting point for finer and more granular localization schemes.

8. Acknowledgements

This work was supported by the DARPA IXO office under NEST project (grant number F336616-01-C-1905) and by the NSF grant CCR-0098269. The authors would like to thank the anonymous reviewers for their invaluable feedback.

9. References

[1] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan and S. Shenker, "GHT: A Geographic Hash Table for Data-Centric Storage", *1st International Workshop on Wireless Sensor Networks and Applications (WSNA)*, Atlanta, GA, 2002

[2] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, B. Krogh, "Energy-Efficient Surveillance System Using Wireless Sensor Networks", *2nd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Boston, MA, 2004.

[3] R. Nagpal, H. Shrobe and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network", *2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, Palo Alto, CA, 2003.

[4] T. He, C. Huang, B. Blum, J. A. Stankovic, T. Abdelzaher, "Range-free localization schemes for large scale sensor networks", *9th International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, CA, 2003

[5] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System", *19th Anual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM)*, Tel Aviv, Israel, 2000.

[6] J. Hightower, G. Boriello and R. Want, "SpotON: An indoor 3D Location Sensing Technology Based on RF Signal Strength", *University of Washington CSE Technical Report #2000-02-02*, 2000.

[7] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin and S. Wicker "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks", *UCLA Technical Report UCLA/CSD-TR 02-0013*, 2002

[8] Y. J. Zhao and R. Govindan "Understanding Packet Delivery Performance in Dense Wireless Sensor Network", *1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, CA 2003.

[9] G. Zhou, T. He and J. A. Stankovic, "Impact of Radio Asymmetry on Wireless Sensor networks", *2nd International Conference on Mobile Systems, Applications, and Services (Mobisys)*, Bostan, MA, 2004

[10] B. Hofmann-Wellenhoff, H. Lichtenegger and J. Collins, "Global Positions System: Theory and Practice", *Fourth Edition. Springer Verlag*, 1997.

[11] N. B. Priyantha, A. Chakraborty and H. Balakrishnan, "The Cricket Location-Support System", *6th International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, 2000.

[12] A. Smith, H. Balakrishnan, M. Goraczko, N. Priyantha, "Tracking Moving Devices with Cricket Localization System", *2nd International Conference on Mobile Systems, Applications, and Services (*Mobisys), Boston, MA, 2004

[13] A. Savvides, C. C. Han and M. B. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors", 7th *International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, 2001.

[14] P. Corke, R. Peterson and D. Rus, "Networked Robots: Flying Robot Navigation using a Sensor Net", *11th International Symposium of Robotics Research (ISRR)*, Siena, Italy, 2003

[15] P.N. Pathirana, A.V. Savkin, S. Jha and N. Bulusu, "Node Localization Using Mobile Robots in Delay-Tolerant Sensor Networks", *IEEE Transactions on Mobile Computing*, 2004

[16] Viacheslav Adamchuk, "Global Positioning System Data Processing", url: pasture.ecn.purdue.edu/~abegps/web_ssm/web_GPS.html

[17] Garmin International Inc., "Owners's Manual and Reference Guide" url: www.garmin.com/manuals/eTrexLegend_OwnersManual.pdf

[18] Crossbow Technologies, "MTS/MDA Sensor and Data Acquisition Boards User's Manual", Document 7430-0020-02, October 2003

[19] P. Dutta, "Echelon Documentation", url: www.cse.ohio-state.edu/~duttap/echelon/

[20] Crossbow Technologies, "MPR/MIB Mote Hardware Users Manual", Document 7430-0021-05, December 2003

[21] TinyOS CVS tree, url: www.sourceforge.net/projects/tinyos

[22] University of Virginia SOWN CVS tree, url: www.sourceforge.net/projects/uvasown

[23] IntraSearch Inc., url: www.mapmart.com