

# A High-Accuracy, Low-Cost Localization System for Wireless Sensor Networks

Radu Stoleru, Tian He, John A. Stankovic, David Luebke

Department of Computer Science

University of Virginia, Charlottesville, VA 22903

{stoleru, tianhe, stankovic, luebke}@cs.virginia.edu

## ABSTRACT

The problem of localization of wireless sensor nodes has long been regarded as very difficult to solve, when considering the realities of real world environments. In this paper, we formally describe, design, implement and evaluate a novel localization system, called Spotlight. Our system uses the spatio-temporal properties of well controlled events in the network (e.g., light), to obtain the locations of sensor nodes. We demonstrate that a high accuracy in localization can be achieved without the aid of expensive hardware on the sensor nodes, as required by other localization systems. We evaluate the performance of our system in deployments of Mica2 and XSM nodes. Through performance evaluations of a real system deployed outdoors, we obtain a 20cm localization error. A sensor network, with any number of nodes, deployed in a 2500m<sup>2</sup> area, can be localized in under 10 minutes, using a device that costs less than \$1000. To the best of our knowledge, this is the first report of a sub-meter localization error, obtained in an outdoor environment, without equipping the wireless sensor nodes with specialized ranging hardware.

## Categories and Subject Descriptors

C.2.4 [Computer-Communications Networks]: Distributed Systems; C.3 [Special-Purpose and Application-Based Systems]: Real-Time and embedded systems.

## General Terms

Algorithms, Measurement, Performance, Design, Experimentation

## Keywords

Wireless Sensor Network, Localization, Event Distribution, Laser

## 1. INTRODUCTION

Recently, wireless sensor network systems have been used in many promising applications including military surveillance, habitat monitoring, wildlife tracking etc. [12] [22] [33] [36]. While many middleware services, to support these applications, have been designed and implemented successfully, localization - finding the position of sensor nodes - remains one of the most difficult research challenges to be solved practically. Since most emerging applications based on networked sensor nodes require location awareness to assist their operations, such as annotating sensed data with location context, it is an indispensable requirement for a sensor node to be able to find its own location.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SenSys'05, November 2-4, 2005, San Diego, California, USA.

Copyright 2005 ACM 1-59593-054-X/05/0011...\$5.00.

Many approaches have been proposed in the literature [4] [6] [13] [14] [19] [20] [21] [23] [27] [28], however it is still not clear how these solutions can be practically and economically deployed. An on-board GPS [23] is a typical high-end solution, which requires sophisticated hardware to achieve high resolution time synchronization with satellites. The constraints on power and cost for tiny sensor nodes preclude this as a viable solution. Other solutions require per node devices that can perform ranging among neighboring nodes. The difficulties of these approaches are twofold. First, under constraints of form factor and power supply, the effective ranges of such devices are very limited. For example the effective range of the ultrasonic transducers used in the Cricket system is less than 2 meters when the sender and receiver are not facing each other [26]. Second, since most sensor nodes are static, i.e. the location is not expected to change, it is not cost-effective to equip these sensors with special circuitry just for a one-time localization. To overcome these limitations, many range-free localization schemes have been proposed. Most of these schemes estimate the location of sensor nodes by exploiting the radio connectivity information among neighboring nodes. These approaches eliminate the need of high-cost specialized hardware, at the cost of a less accurate localization. In addition, the radio propagation characteristics vary over time and are environment dependent, thus imposing high calibration costs for the range-free localizations schemes. With such limitations in mind, this paper addresses the following research challenge: How to reconcile the need for high accuracy in location estimation with the cost to achieve it. Our answer to this challenge is a localization system called Spotlight. This system employs an asymmetric architecture, in which sensor nodes do not need any additional hardware, other than what they currently have. All the sophisticated hardware and computation reside on a single Spotlight device. The Spotlight device uses a steerable laser light source, illuminating the sensor nodes placed within a known terrain. We demonstrate that this localization is much more accurate (i.e., tens of centimeters) than the range-based localization schemes and that it has a much longer effective range (i.e., thousands of meters) than the solutions based on ultra-sound/acoustic ranging. At the same time, since only a single sophisticated device is needed to localize the whole network, the amortized cost is much smaller than the cost to add hardware components to the individual sensors.

## 2. RELATED WORK

In this section, we discuss prior work in localization in two major categories: the range-based localization schemes (which use either expensive, per node, ranging devices for high accuracy, or less accurate ranging solutions, as the Received Signal Strength Indicator (RSSI)), and the range-free schemes, which use only connectivity information (hop-by-hop) as an indication of proximity among the nodes.

The localization problem is a fundamental research problem in many domains. In the field of robotics, it has been studied extensively [9] [10]. The reported localization errors are on the order of tens of centimeters, when using specialized ranging hardware, i.e. laser range finder or ultrasound. Due to the high cost and non-negligible form factor of the ranging hardware, these solutions can not be simply applied to sensor networks.

The RSSI has been an attractive solution for estimating the distance between the sender and the receiver. The RADAR system [2] uses the RSSI to build a centralized repository of signal strengths at various positions with respect to a set of beacon nodes. The location of a mobile user is estimated within a few meters. In a similar approach, MoteTrack [17] distributes the reference RSSI values to the beacon nodes.

Solutions that use RSSI and do not require beacon nodes have also been proposed [5] [14] [24] [26] [29]. They all share the idea of using a mobile beacon. The sensor nodes that receive the beacons, apply different algorithms for inferring their location. In [29], Sichitiu proposes a solution in which the nodes that receive the beacon construct, based on the RSSI value, a constraint on their position estimate. In [26], Priyantha et al. propose MAL, a localization method in which a mobile node (moving strategically) assists in measuring distances between node pairs, until the constraints on distances generate a rigid graph. In [24], Pathirana et al. formulate the localization problem as an on-line estimation in a nonlinear dynamic system and proposes a Robust Extended Kalman Filter for solving it. Elnahrawy [8] provides strong evidence of inherent limitations of localization accuracy using RSSI, in indoor environments.

A more precise ranging technique uses the time difference between a radio signal and an acoustic wave, to obtain pair wise distances between sensor nodes. This approach produces smaller localization errors, at the cost of additional hardware. The Cricket location-support system [25] can achieve a location granularity of tens of centimeters with short range ultrasound transceivers. AHLoS, proposed by Savvides et al. [27], employs Time of Arrival (ToA) ranging techniques that require extensive hardware and solving relatively large nonlinear systems of equations. A similar ToA technique is employed in [3].

In [30], Simon et al. implement a distributed system (using acoustic ranging) which locates a sniper in an urban terrain. Acoustic ranging for localization is also used by Kwon et al. [15]. The reported errors in localization vary from 2.2m to 9.5m, depending on the type (centralized vs. distributed) of the Least Square Scaling algorithm used.

For wireless sensor networks ranging is a difficult option. The hardware cost, the energy expenditure, the form factor, the small range, all are difficult compromises, and it is hard to envision cheap, unreliable and resource-constraint devices make use of range-based localization solutions. However, the high localization accuracy, achievable by these schemes is very desirable.

To overcome the challenges posed by the range-based localization schemes, when applied to sensor networks, a different approach has been proposed and evaluated in the past. This approach is called range-free and it attempts to obtain location information from the proximity to a set of known beacon nodes.

Bulusu et al. propose in [4] a localization scheme, called Centroid, in which each node localizes itself to the centroid of its proximate beacon nodes. In [13], He et al. propose APIT, a scheme in which each node decides its position based on the possibility of being inside or outside of a triangle formed by any three beacon nodes heard by the node. The Global Coordinate System [20],

developed at MIT, uses apriori knowledge of the node density in the network, to estimate the average hop distance. The DV-\* family of localization schemes [21], uses the hop count from known beacon nodes to the nodes in the network to infer the distance. The majority of range-free localization schemes have been evaluated in simulations, or controlled environments. Several studies [11] [32] [34] have emphasized the challenges that real environments pose. Langendoen and Reijers present a detailed, comparative study of several localization schemes in [16].

To the best of our knowledge, Spotlight is the first range-free localization scheme that works very well in an outdoor environment. Our system requires a line of sight between a single device and the sensor nodes, and the map of the terrain where the sensor field is located. The Spotlight system has a long effective range (1000's meters) and does not require any infrastructure or additional hardware for sensor nodes. The Spotlight system combines the advantages and does not suffer from the disadvantages of the two localization classes.

### 3. SPOTLIGHT SYSTEM DESIGN

The main idea of the Spotlight localization system is to generate controlled events in the field where the sensor nodes were deployed. An event could be, for example, the presence of light in an area. Using the time when an event is perceived by a sensor node and the spatio-temporal properties of the generated events, spatial information (i.e. location) regarding the sensor node can be inferred.



**Figure 1. Localization of a sensor network using the Spotlight system**

We envision, and depict in Figure 1, a sensor network deployment and localization scenario as follows: wireless sensor nodes are randomly deployed from an unmanned aerial vehicle. After deployment, the sensor nodes self-organize into a network and execute a time-synchronization protocol. An aerial vehicle (e.g. helicopter), equipped with a device, called Spotlight, flies over the network and generates light events. The sensor nodes detect the events and report back to the Spotlight device, through a base station, the timestamps when the events were detected. The Spotlight device computes the location of the sensor nodes.

During the design of our Spotlight system, we made the following assumptions:

- the sensor network to be localized is connected and a middleware, able to forward data from the sensor nodes to the Spotlight device, is present.
- the aerial vehicle has a very good knowledge about its position and orientation (6 parameters: 3 translation and 3 rigid-body rotation) and that it possesses the map of the field where the network was deployed.
- a powerful Spotlight device is available and it is able to generate

spatially large events that can be detected by the sensor nodes, even in the presence of background noise (daylight).

- a line of sight between the Spotlight device and sensor nodes exists.

Our assumptions are simplifying assumptions, meant to reduce the complexity of the presentation, for clarity. We propose solutions that do not rely on these simplifying assumptions, in Section 6.

In order to formally describe and generalize the Spotlight localization system, we introduce the following definitions.

### 3.1 Definitions and Problem Formulation

Let's assume that the space  $A \subset R^3$  contains all sensor nodes  $N$ , and that each node  $N_i$  is positioned at  $p_i(x, y, z)$ . To obtain  $p_i(x, y, z)$ , a Spotlight localization system needs to support three main functions, namely an Event Distribution Function (EDF)  $E(t)$ , an Event Detection Function  $D(e)$ , and a Localization Function  $L(T_i)$ . They are formally defined as follows:

**Definition 1:** An event  $e(t, p)$  is a detectable phenomenon that occurs at time  $t$  and at point  $p \in A$ . Examples of events are light, heat, smoke, sound, etc. Let  $T_i = \{t_{i1}, t_{i2}, \dots, t_{in}\}$  be a set of  $n$  timestamps of events detected by a node  $i$ . Let  $T' = \{t_1', t_2', \dots, t_m'\}$  be the set of  $m$  timestamps of events generated in the sensor field.

**Definition 2:** The Event Detection Function  $D(e)$  defines a binary detection algorithm. For a given event  $e$ :

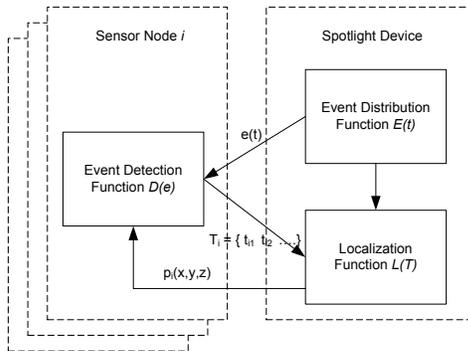
$$D(e) = \begin{cases} \text{true,} & \text{Event } e \text{ is detected} \\ \text{false,} & \text{Event } e \text{ is not detected} \end{cases} \quad (1)$$

**Definition 3:** The Event Distribution Function (EDF)  $E(t)$  defines the point distribution of events within  $A$  at time  $t$ :

$$E(t) = \{p \mid p \in A \wedge D(e(t, p)) = \text{true}\} \quad (2)$$

**Definition 4:** The Localization Function  $L(T_i)$  defines a localization algorithm with input  $T_i$ , a sequence of timestamps of events detected by the node  $i$ :

$$L(T_i) = \bigcap_{t \in T_i} E(t) \quad (3)$$



**Figure 2. Spotlight system architecture**

As shown in Figure 2, the Event Detection Function  $D(e)$  is supported by the sensor nodes. It is used to determine whether an external event happens or not. It can be implemented through either a simple threshold-based detection algorithm or other advanced digital signal processing techniques. The Event Distribution  $E(t)$  and Localization Functions  $L(T_i)$  are implemented by a Spotlight device. The Localization function is an aggregation algorithm which calculates the intersection of multiple sets of

points. The Event Distribution Function  $E(t)$  describes the distribution of events over time. It is the core of the Spotlight system and it is much more sophisticated than the other two functions. Due to the fact that  $E(t)$  is realized by the Spotlight device, the hardware requirements for the sensor nodes remain minimal.

With the support of these three functions, the localization process goes as follows:

- 1) A Spotlight device distributes events in the space  $A$  over a period of time.
- 2) During the event distribution, sensor nodes record the time sequence  $T_i = \{t_{i1}, t_{i2}, \dots, t_{in}\}$  at which they detect the events.
- 3) After the event distribution, each sensor node sends the detection time sequence back to the Spotlight device.
- 4) The Spotlight device estimates the location of a sensor node  $i$ , using the time sequence  $T_i$  and the known  $E(t)$  function.

The Event Distribution Function  $E(t)$  is the core technique used in the Spotlight system and we propose three designs for it. These designs have different tradeoffs and the cost comparison is presented in Section 3.5.

### 3.2 Point Scan Event Distribution Function

To illustrate the basic functionality of a Spotlight system, we start with a simple sensor system where a set of nodes are placed along a straight line ( $A = [0, l] \subset R$ ). The Spotlight device generates point events (e.g. light spots) along this line with constant speed  $s$ . The set of timestamps of events detected by a node  $i$  is  $T_i = \{t_{i1}\}$ . The Event Distribution Function  $E(t)$  is:

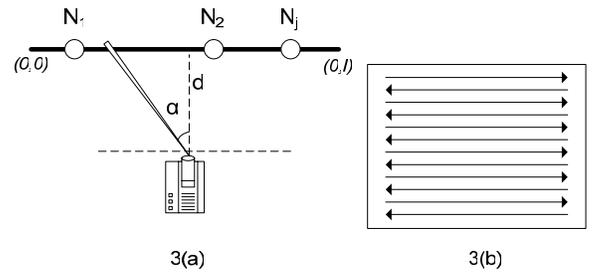
$$E(t) = \{p \mid p \in A \wedge p = t * s\} \quad (4)$$

where  $t \in [0, l/s]$ . The resulting localization function is:

$$L(T_i) = E(t_{i1}) = \{t_{i1} * s\} \quad (5)$$

where  $D(e(t_{i1}, p_i)) = \text{true}$  for node  $i$  positioned at  $p_i$ .

The implementation of the Event Distribution Function  $E(t)$  is straightforward. As shown in Figure 3(a), when a light source emits a beam of light with the angular speed given by  $S_\alpha = \frac{d\alpha}{dt} = \frac{s \cos^2(\alpha)}{d}$ , a light spot event with constant speed  $s$  is generated along the line situated at distance  $d$ .



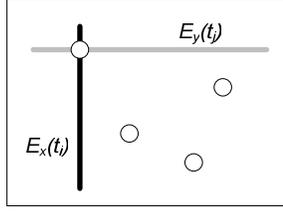
**Figure 3. The implementation of the Point Scan EDF**

The Point Scan EDF can be generalized to the case where nodes are placed in a two dimensional plane  $R^2$ . In this case, the Spotlight system progressively scans the plane to activate the sensor nodes. This scenario is depicted in Figure 3(b).

### 3.3 Line Scan Event Distribution Function

Some devices, e.g. diode lasers, can generate an entire line of events simultaneously. With these devices, we can support the Line Scan Event Distributed Function easily. We assume that the

sensor nodes are placed in a two dimensional plane ( $A=[l \times l] \subset R^2$ ) and that the scanning speed is  $s$ . The set of timestamps of events detected by a node  $i$  is  $T_i=\{t_{i1}, t_{i2}\}$ .



**Figure 4. The implementation of the Line Scan EDF**

The Line Scan EDF is defined as follows:

$$E_x(t) = \{p_k \mid k \in [0, l] \wedge p_k = (t * s, k)\}$$

for  $t \in [0, l/s]$  and:

$$E_y(t) = \{p_k \mid k \in [0, l] \wedge p_k = (k, t * s - l)\} \quad (6)$$

for  $t \in [l/s, 2l/s]$ .

$$E(t) = E_x(t) \cup E_y(t)$$

We can localize a node by calculating the intersection of the two event lines, as shown in Figure 4. More formally:

$$L(T_i) = E(t_{i1}) \cap E(t_{i2}) \quad (7)$$

where  $D(e(t_{i1}, p_i)) = true$ ,  $D(e(t_{i2}, p_i)) = true$  for node  $i$  positioned at  $p_i$ .

### 3.4 Area Cover Event Distribution Function

Other devices, such as light projectors, can generate events that cover an area. This allows the implementation of the Area Cover EDF. The idea of Area Cover EDF is to partition the space  $A$  into multiple sections and assign a unique binary identifier, called code, to each section. Let's suppose that the localization is done within a plane ( $A \subset R^2$ ). Each section  $S_k$  within  $A$  has a unique code  $k$ . The Area Cover EDF is then defined as follows:

$$BIT(k, j) = \begin{cases} true, & \text{if } j^{\text{th}} \text{ bit of } k \text{ is } 1 \\ false, & \text{if } j^{\text{th}} \text{ bit of } k \text{ is } 0 \end{cases} \quad (8)$$

$$E(t) = \{p \mid p \in S_k \wedge BIT(k, t) = true\}$$

and the corresponding localization algorithm is:

$$L(T_i) = \{p \mid p = COG(S_k) \wedge (BIT(k, t) = true \text{ if } t \in T_i) \wedge (BIT(k, t) = false \text{ if } t \in T - T_i)\} \quad (9)$$

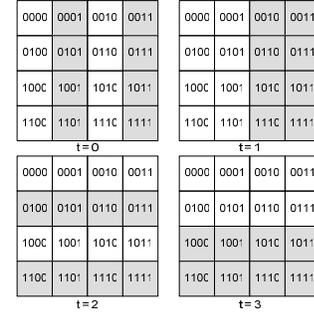
where  $COG(S_k)$  denotes the center of gravity of  $S_k$ .

We illustrate the Area Cover EDF with a simple example. As shown in Figure 5, the plane  $A$  is divided in 16 sections. Each section  $S_k$  has a unique code  $k$ . The Spotlight device distributes the events according to these codes: at time  $j$  a section  $S_k$  is covered by an event (lit by light), if  $j^{\text{th}}$  bit of  $k$  is 1. A node residing anywhere in the section  $S_k$  is localized at the center of gravity of that section. For example, nodes within section 1010 detect the events at time  $T = \{1, 3\}$ . At  $t = 4$  the section where each node resides can be determined

A more accurate localization requires a finer partitioning of the plane, hence the number of bits in the code will increase. Considering the noise that is present in a real, outdoor environment, it is easy to observe that a relatively small error in detecting the correct bit pattern could result in a large localization

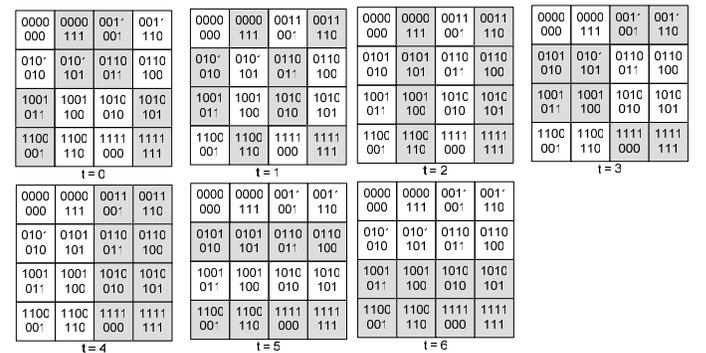
error. Returning to the example shown in Figure 5, if a sensor node is located in the section with code 0000, and due to the noise, at time  $t = 3$ , it thinks it detected an event, it will incorrectly conclude that its code is 1000, and it positions itself two squares below its correct position. The localization accuracy can deteriorate even further, if multiple errors are present in the transmission of the code.

A natural solution to this problem is to use error-correcting codes, which greatly reduce the probability of an error, without paying the price of a re-transmission, or lengthening the transmission time too much. Several error correction schemes have been proposed in the past. Two of the most notable ones are the Hamming (7, 4) code and the Golay (23, 12) code. Both are perfect linear error correcting codes. The Hamming coding scheme can detect up to 2-bit errors and correct 1-bit errors. In the Hamming (7, 4) scheme, a message having 4 bits of data (e.g. *dddd*, where  $d$  is a data bit) is transmitted as a 7-bit word by adding 3 error control bits (e.g. *dddpppp*, where  $p$  is a parity bit).



**Figure 5. The steps of Area Cover EDF. The events cover the shaded areas.**

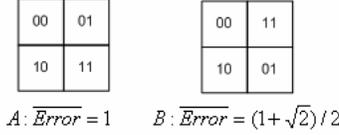
The steps of the Area Cover technique, when using Hamming (7, 4) scheme are shown in Figure 6. Golay codes can detect up to 6-bit errors and correct up to 3-bit errors. Similar to Hamming (7, 4), Golay constructs a 23-bit codeword from 12-bit data. Golay codes have been used in satellite and spacecraft data transmission and are most suitable in cases where short codeword lengths are desirable.



**Figure 6. The steps of Area Cover EDF with Hamming (7, 4) ECC. The events cover the shaded areas.**

Let's assume a 1-bit error probability of 0.01, and a 12-bit message that needs to be transmitted. The probability of a failed transmission is thus: 0.11, if no error detection and correction is used; 0.0061 for the Hamming scheme (i.e. more than 1-bit error); and 0.000076 for the Golay scheme (i.e. more than 3-bit errors). Golay is thus 80 times more robust than the Hamming scheme, which is 20 times more robust than the no error correction scheme.

Considering that a limited number of corrections is possible by any coding scheme, a natural question arises: can we minimize the localization error when there are errors that can not be corrected? This can be achieved by a clever placement of codes in the grid. As shown in Figure 7, the placement A, in the presence of a 1-bit error has a smaller average localization error when compared to the placement B. The objective of our code placement strategy is to reduce the total Euclidean distance between all pairs of codes with Hamming distances smaller than  $K$ , the largest number of expected 1-bit errors.



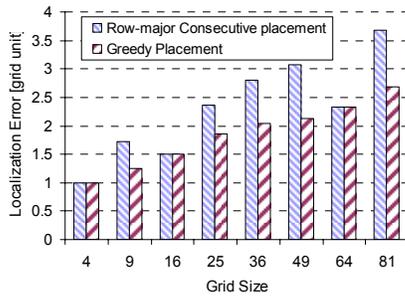
**Figure 7. Different code placement strategies**

Formally, a placement is represented by a function  $P: [0, l]^d \rightarrow C$ , which assigns a code to every coordinate in the  $d$ -dimensional cube of size  $l$  (e.g., in the planar case, we place codes in a 2-dimensional grid). We denote by  $d_E(i, j)$  the Euclidean distance between two codes  $i$  and  $j$ . In a noisy environment,  $d_H(i, j)$  determines the crossover probability between the two codes. For the case of independent detections, the higher  $d_H(i, j)$  is, the lower the crossover probability will be. The objective function is defined as follows:

$$\min \left\{ \sum_{d_H(i,j) \leq K} d_E(i,j) \right\} \text{ where } i, j \in [0, l]^d \quad (10)$$

Equation 10 is a non-linear and non-convex programming problem. In general, it is analytically hard to obtain the global minimum. To overcome this, we propose a Greedy Placement method to obtain suboptimal results. In this method we initialize the 2-dimensional grid with codes. Then we swap the codes within the grid repeatedly, to minimize the objective function. For each swap, we greedily chose a pair of codes, which can reduce the objective function (Equation 10) the most. The proposed Greedy Placement method ends when no swap of codes can further minimize the objective function.

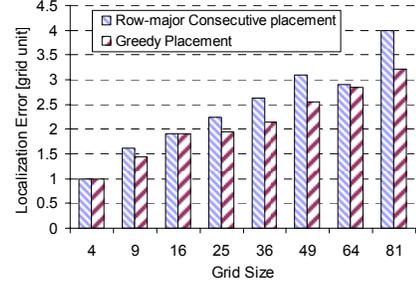
For evaluation, we compared the average localization error in the presence of  $K$ -bit error for two strategies: the proposed Greedy Placement and the Row-Major Placement (it places the codes consecutively in the array, in row-first order).



**Figure 8. Localization error with code placement and no ECC**

As Figure 8 shows, if no error detection/correction capability is present and 1-bit errors occur, then our Greedy Placement method can reduce the localization error by an average 23%, when compared to the Row-Major Placement. If error detection and correction schemes are used (e.g. Hamming (12, 8) and if 3-bit errors occur ( $K=3$ ) then the Greedy Placement method reduces

localization error by 12%, when compared to the Row-Major Placement, as shown in Figure 9. If  $K=1$ , then there is no benefit in using the Greedy Placement method, since the 1-bit error can be corrected by the Hamming scheme.



**Figure 9. Localization error with code placement and Hamming ECC**

### 3.5 Event Distribution Function Analysis

Although all three aforementioned techniques are able to localize the sensor nodes, they differ in the localization time, communication overhead and energy consumed by the Event Distribution Function (let's call it Event Overhead). Let's assume that all sensor nodes are located in a square with edge size  $D$ , and that the Spotlight device can generate  $N$  events (e.g. Point, Line and Area Cover events) every second and that the maximum tolerable localization error is  $r$ . Table 1 presents the execution cost comparison of the three different Spotlight techniques.

**Table 1. Execution Cost Comparison**

Criterion	Point Scan	Line Scan	Area Cover
Localization Time	$(D^2/r^2)/N$	$(2D/r)/N$	$\log_r D/N$
# Detections	1	2	$\log_r D$
# Time Stamps	1	2	$\log_r D$
Event Overhead	$D^2$	$2D^2$	$D^2 \log_r D/2$

Table 1 indicates that the Event Overhead for the Point Scan method is the smallest - it requires a one-time coverage of the area, hence the  $D^2$ . However the Point Scan takes a much longer time than the Area Cover technique, which finishes in  $\log_r D$  seconds. The Line Scan method trades the Event Overhead well with the localization time. By doubling the Event Overhead, the Line Scan method takes only  $r/2D$  percentage of time to complete, when compared with the Point Scan method. From Table 1, it can be observed that the execution costs do not depend on the number of sensor nodes to be localized.

It is important to remark the ratio Event Overhead per unit time, which is indicative of the power requirement for the Spotlight device. This ratio is constant for the Point Scan ( $r^2*N$ ) while it grows linearly with area, for the Area Cover ( $D^2*N/2$ ). If the deployment area is very large, the use of the Area Cover EDF is prohibitively expensive, if not impossible. For practical purposes, the Area Cover is a viable solution for small to medium size networks, while the Line Scan works well for large networks. We discuss the implications of the power requirement for the Spotlight device, and offer a hybrid solution in Section 6.

### 3.6 Localization Error Analysis

The accuracy of localization with the Spotlight technique depends on many aspects. The major factors that were considered during the implementation of the system are discussed below:

- Time Synchronization: the Spotlight system exchanges time stamps between sensor nodes and the Spotlight device. It is necessary for the system to reach consensus on global time through synchronization. Due to the uncertainty in hardware processing and wireless communication, we can only confine such errors within certain bounds (e.g. one jiffy). An imprecise input to the Localization Function  $L(T)$  leads to an error in node localization.
- Uncertainty in Detection: the sampling rate of the sensor nodes is finite, consequently, there will be an unpredictable delay between the time when an event is truly present and when the sensor node detects it. Lower sampling rates will generate larger localizations errors.
- Size of the Event: the events distributed by the Spotlight device can not be infinitely small. If a node detects one event, it is hard for it to estimate the exact location of itself within the event.
- Realization of Event Distribution Function: EDF defines locations of events at time  $t$ . Due to the limited accuracy (e.g. mechanical imprecision), a Spotlight device might generate events which locate differently from where these events are supposed to be.

It is important to remark that the localization error is independent of the number of sensor nodes in the network. This independence, as well as the aforementioned independence of the execution cost, indicate the very good scalability properties (with the number of sensor nodes, but not with the area of deployment) that the Spotlight system possesses.

#### 4. SYSTEM IMPLEMENTATION

For our performance evaluation we implemented two Spotlight systems. Using these two implementations we were able to investigate the full spectrum of Event Distribution techniques, proposed in Section 3, at a reduced “one time” cost (less than \$1,000).

The first implementation, called  $\mu$ Spotlight, had a short range (10-20 meters), however its capability of generating the entire spectrum of EDFs made it very useful. We used this implementation mainly to investigate the capabilities of the Spotlight system and tune its performance. It was not intended to represent the full solution, but only a scaled down version of the system.

The second implementation, the Spotlight system, had a much longer range (as far as 6500m), but it was limited in the types of EDFs that it can generate. The goal of this implementation was to show how the Spotlight system works in a real, outdoor environment, and show correlations with the experimental results obtained from the  $\mu$ Spotlight system implementation.

In the remaining part of this section, we describe how we implemented the three components (Event Distribution, Event Detection and Localization functions) of the Spotlight architecture, and the time synchronization protocol, a key component of our system.

##### 4.1 $\mu$ Spotlight System

The first system we built, called  $\mu$ Spotlight, used as the Spotlight device, an Infocus LD530 projector connected to an IBM Thinkpad laptop. The system is shown in Figure 10.

The Event Distribution Function was implemented as a Java GUI. Due to the stringent timing requirements and the delay caused by the buffering in the windowing system of a PC, we used the Full-Screen Exclusive Mode API provided by Java2. This allowed us to bypass the windowing system and more precisely

estimate the time when an event is displayed by the projector, hence a higher accuracy of timestamps of events. Because of the 50Hz refresh rate of our projector, there was still an uncertainty in the time stamping of the events of 20msec. We explored the possibility of using and modifying the Linux kernel to expose the vertical synch (VSYNCH) interrupt, generated by the displaying device after each screen refresh, out of the kernel mode. The performance evaluation results showed, however, that this level of accuracy was not needed.

The sensor nodes that we used were Berkeley Mica2 motes equipped with MTS310 multi-sensor boards from Crossbow. This sensor board contains a CdSe photo sensor which can detect the light from the projector.

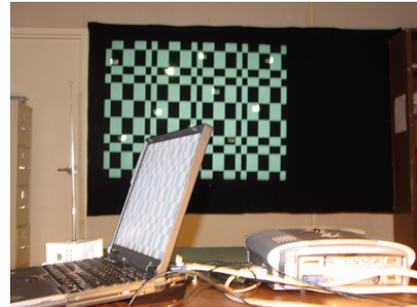


Figure 10.  $\mu$ Spotlight system implementation

With this implementation of the Spotlight system, we were able to generate Point, Line and Area Scan events.

##### 4.2 Spotlight System

The second Spotlight system we built used, as the Spotlight device, diode lasers, a computerized telescope mount (Celestron CG-5GT, shown in Figure 11), and an IBM Thinkpad laptop. The laptop was connected, through RS232 interfaces, to the telescope mount and to one XSM600CA [7] mote, acting as a base station.

The diode lasers we used ranged in power from 7mW to 35mW. They emitted at 650nm, close to the point of highest sensitivity for CdSe photosensor. The diode lasers were equipped with lenses that allowed us to control the divergence of the beam.



Figure 11. Spotlight system implementation

The telescope mount has worm gears for a smooth motion and high precision angular measurements. The two angular measures that we used were the, so called, Alt (from Altitude) and Az (from Azimuth). In astronomy, the Altitude of a celestial object is its angular distance above or below the celestial horizon, and the Azimuth is the angular distance of an object eastwards of the meridian, along the horizon.

The laptop computer, through a Java GUI, controls the motion of the telescope mount, orienting it such that a full Point Scan of an area is performed, similar to the one described in Figure 3(b). For each turning point  $i$ , the 3-tuple (Alt $_i$  and Az $_i$  angles and the timestamp  $t_i$ ) is recorded. The Spotlight system uses the timestamp received from a sensor node  $j$ , to obtain the angular measures Alt $_j$  and Az $_j$  for its location.

For the sensor nodes, we used XSM motes, mainly because of their longer communication range. The XSM mote has the photo sensor embedded in its main board. We had to make minor adjustments to the plastic housing, in order to expose the photo sensor to the outside. The same mote code, written in nesC, for TinyOS, was used for both  $\mu$ Spotlight and Spotlight system implementations.

### 4.3 Event Detection Function $D(t)$

The Event Detection Function aims to detect the beginning of an event and record the time when the event was observed. We implemented a very simple detection function based on the observed maximum value. An event  $i$  will be time stamped with time  $t_i$ , if the reading from the photo sensor  $d_{it}$ , fulfills the condition:

$$d_{max} + \Delta < d_{t_i}$$

where  $d_{max}$  is the maximum value reported by the photo sensor before  $t_i$  and  $\Delta$  is a constant which ensures that the first large detection gives the timestamp of the event (i.e. small variations around the first large signal are not considered). Hence  $\Delta$  guarantees that only sharp changes in the detected value generate an observed event.

### 4.4 Localization Function $L(T)$

The Localization Function is implemented in the Java GUI. It matches the timestamps created by the Event Distribution Function with those reported by the sensor nodes.

The Localization Function for the Point Scan EDF has as input a time sequence  $T_i = \{t_i\}$ , as reported by node  $i$ . The function performs a simple search for the event with a timestamp closest to  $t_i$ . If  $t_i$  is constrained by:

$$t_{e_n} < t_i < t_{e_{n+1}}$$

where  $e_n$  and  $e_{n+1}$  are two consecutive events, then the obtained location for node  $i$  is:

$$x = x_{e_{n+1}}, y = y_{e_{n+1}}$$

The case for the Line Scan is treated similarly. The input to the Localization Function is the time sequence  $T_i = \{t_1, t_2\}$  as reported by node  $i$ . If the reported timestamps are constrained by:

$$t_{e_n} < t_1 < t_{e_{n+1}}, \text{ and } t_{e_m} < t_2 < t_{e_{m+1}}$$

where  $e_n$  and  $e_{n+1}$  are two consecutive events on the horizontal scan and  $e_m$  and  $e_{m+1}$  are two consecutive events on vertical scan, then the inferred location for node  $i$  is:

$$x = x_{e_{n+1}}, y = y_{e_{m+1}}$$

The Localization Function for the Area Cover EDF has as input a timestamp set  $T_i = \{t_{i1}, t_{i2}, \dots, t_{in}\}$  of the  $n$  events, detected by node  $i$ . We recall the notation for the set of  $m$  timestamps of events generated by the Spotlight device,  $T' = \{t_1', t_2', \dots, t_m'\}$ . A code  $d_i = d_{i1}d_{i2} \dots d_{im}$  is then constructed for each node  $i$ , such that  $d_{ij} = 1$  if

$t_j' \in T_i$  and  $d_{ij} = 0$  if  $t_j' \notin T_i$ . The function performs a search for an event with an identical code. If the following condition is true:

$$d_i = d_{e_n}$$

where  $e_n$  is an event with code  $d_{en}$ , then the inferred location for node  $i$  is:

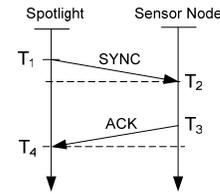
$$x = x_{e_n}, y = y_{e_n}$$

## 4.5 Time Synchronization

The time synchronization in the Spotlight system consists of two parts:

- Synchronization between sensor nodes: This is achieved through the Flooding Time Synchronization Protocol [18]. In this protocol, synchronized nodes (the root node is the only synchronized node at the beginning) send time synchronization message to unsynchronized nodes. The sender puts the time stamp into the synchronization message right before the bytes containing the time stamp are transmitted. Once a receiver gets the message, it follows the sender's time and performs the necessary calculations to compensate for the clock drift.

- Synchronization between the sensor nodes and the Spotlight device: We implemented this part through a two-way handshaking between the Spotlight device and one node, used as the base station. The sensor node is attached to the Spotlight device through a serial interface.



**Figure 12. Two-way synchronization**

As shown in Figure 12, let's assume that the Spotlight device sends a synchronization message (SYNC) at local time  $T_1$ , the sensor node receives it at its local time  $T_2$  and acknowledges it at local time  $T_3$  (both  $T_2$  and  $T_3$  are sent back through ACK). After the Spotlight device receives the ACK, at its local time  $T_4$ , the time synchronization can be achieved as follows:

$$Offset = \frac{(T_2 - T_1) + (T_3 - T_4)}{2} \quad (11)$$

$$T_{global} = T_{node} = T_{spotlight} + Offset$$

We note that Equation 11 assumes that the one trip delays are the same in both directions. In practice this does not hold well enough. To improve the performance, we separate the handshaking process from the timestamp exchanges. The handshaking is done fast, through a 2 byte exchange between the Spotlight device and the sensor node (the timestamps are still recorded, but not sent). After this fast handshaking, the recorded time stamps are exchanged. The result indicates that this approach can significantly improve the accuracy of time synchronization.

## 5. PERFORMANCE EVALUATION

In this section we present the performance evaluation of the Spotlight systems when using the three event distribution functions, i.e. Point Scan, Line Scan and Area Cover, described in Section 3.

For the  $\mu$ Spotlight system we used 10 Mica2 nodes. The sensor nodes were attached to a vertically positioned Veltex board. By projecting the light to the sensor nodes, we are able to generate well controlled Point, Line and Area events. The Spotlight device was able to generate events, i.e. project light patterns, covering an area of approximate size  $180 \times 140 \text{cm}^2$ . The screen resolution for the projector was  $1024 \times 768$ , and the movement of the Point Scan and Line Scan techniques was done through increments (in the appropriate direction) of 10 pixels between events.

Each experimental point was obtained from 10 successive runs of the localization procedure. Each set of 10 runs was preceded by a calibration phase, aimed at estimating the total delays (between the Spotlight device and each sensor node) in detecting an event. During the calibration, we created an event covering the entire sensor field (illuminated the entire area). The timestamp reported by each sensor node, in conjunction with the timestamp created by the Spotlight device were used to obtain the time offset, for each sensor node. More sophisticated calibration procedures have been reported previously [35]. In addition to the time offset, we added a manually configurable parameter, called bias. It was used to best estimate the center of an event.



**Figure 13. Deployment site for the Spotlight system**

For the Spotlight system evaluation, we deployed 10 XSM nodes in a football field. The site is shown in Figure 13 (laser beams are depicted with red arrows and sensor nodes with white dots). Two sets of experiments were run, with the Spotlight device positioned at 46m and at 170m from the sensor field. The sensor nodes were aligned and the Spotlight device executed a Point Scan. The localization system computed the coordinates of the sensor nodes, and the Spotlight device was oriented, through a GoTo command sent to the telescope mount, towards the computed location. In the initial stages of the experiments, we manually measured the localization error.

For our experimental evaluation, the metrics of interest were as follows:

- Localization error, defined as the distance, between the real location and the one obtained from the Spotlight system.
- Localization duration, defined as the time span between the first and last event.
- Localization range, defined as the maximum distance between the Spotlight device and the sensor nodes.
- A Localization Cost function  $Cost: \{\{\text{localization accuracy}\}, \{\text{localization duration}\}\} \rightarrow [0,1]$  quantifies the trade-off between the accuracy in localization and the localization duration. The objective is to minimize the Localization Cost function. By

denoting with  $e_i$  the localization error for the  $i^{\text{th}}$  scenario, with  $d_i$  the localization duration for the  $i^{\text{th}}$  scenario, with  $\max(e)$  the maximum localization error, with  $\max(d)$  the maximum localization duration, and with  $\alpha$  the importance factor, the Localization Cost function is formally defined as:

$$Cost(e_i, d_i) = \alpha * \frac{e_i}{\max(e)} + (1 - \alpha) * \frac{d_i}{\max(d)}$$

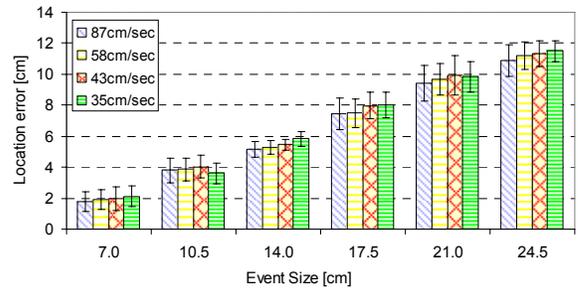
- Localization Bias. This metric is used to investigate the effectiveness of the calibration procedure. If, for example, all computed locations have a bias in the west direction, a calibration factor can be used to compensate for the difference.

The parameters that we varied during the performance evaluation of our system were: the type of scanning (Point, Line and Area), the size of the event, the duration of the event (for Area Cover), the scanning speed, the power of the laser and the distance between the Spotlight device and sensor field, to estimate the range of the system.

### 5.1 Point Scan - $\mu$ Spotlight system

In this experiment, we investigated how the size of the event and the scanning speed affect the localization error. Figure 14 shows the mean localization errors with their standard deviations.

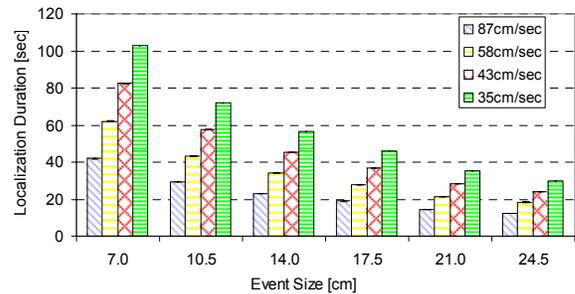
It can be observed, that while the scanning speed, varying between 35cm/sec and 87cm/sec has a minor influence on the localization accuracy, the size of the event has a dramatic effect.



**Figure 14. Localization Error vs. Event Size for the Point Scan EDF**

The obtained localization error varied from as little as 2cm to over 11cm for the largest event. This dependence can be explained by our Event Detection algorithm: the first detection above a threshold gave the timestamp for the event.

The duration of the localization scheme is shown in Figure 15. The dependency of the localization duration on the size of the event and scanning speed is natural. A bigger event allows a reduction in the total duration of up to 70%. The localization duration is directly proportional to the scanning speed, as expected, and depicted in Figure 15.



**Figure 15. Localization Duration vs. Event Size for the Point Scan EDF**

An interesting trade-off is between the localization accuracy (usually the most important factor), and the localization time (important in environments where stealthiness is paramount). Figure 16 shows the Localization Cost function, for  $\alpha = 0.5$  (accuracy and duration are equally important).

As shown in Figure 16, it can be observed that an event size of approximately 10-15cm (depending on the scanning speed) minimizes our Cost function. For  $\alpha = 1$ , the same graph would be a monotonically increasing function, while for  $\alpha = 0$ , it would be monotonically decreasing function.

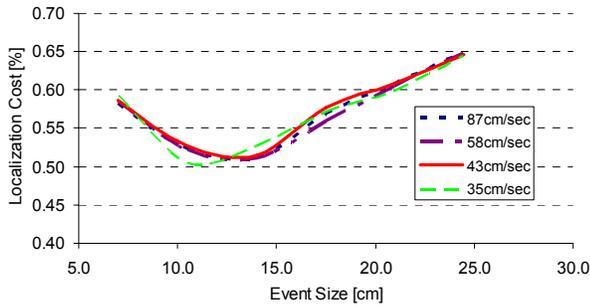


Figure 16. Localization Cost vs. Event Size for the Point Scan EDF

## 5.2 Line Scan - $\mu$ Spotlight system

In a similar manner to the Point Scan EDF, for the Line Scan EDF we were interested in the dependency of the localization error and duration on the size of the event and scanning speed.

We represent in Figure 17 the localization error for different event sizes. It is interesting to observe the dependency (concave shape) of the localization error vs. the event size. Moreover, a question that should arise is why the same dependency was not observed in the case of Point Scan EDF.

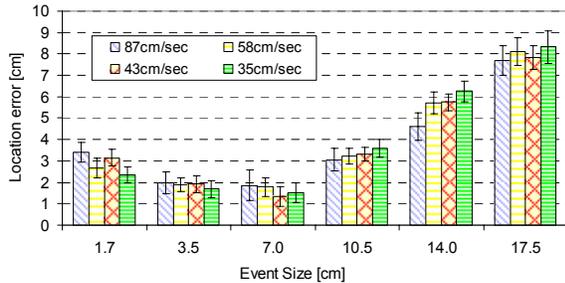


Figure 17. Localization Error vs. Event Size for the Line Scan EDF

The explanation for this concave dependency is the existence of a bias in location estimation. As a reminder, a bias factor was introduced in order to best estimate the central point of events that have a large size. What Figure 17 shows is the fact that the bias factor was optimal for an event size of approximately 7cm. For events smaller and larger than this, the bias factor was too large, and too small, respectively. Thus, it introduced biased errors in the position estimation.

The reason why we did not observe the same dependency in the case of the Point Scan EDF was that we did not experiment with event sizes below 7cm, due to the long time it would have taken to scan the entire field with events as small as 1.7cm.

The results for the localization duration as a function of the size of the event are shown in Figure 18. As shown, the localization

duration is directly proportional to the scanning speed. The size of the event has a smaller influence on the localization duration. One can remark the average localization duration of about 10sec, much shorter than the duration obtained in the Point Scan experiment.

The Localization Cost function dependency on the event size and scanning speed, for  $\alpha=0.5$ , is shown in Figure 19. The dependency on the scanning speed is very small (the Cost Function achieves a minimum in the same 4-6cm range). It is interesting to note that this 4-6cm optimal event size is smaller than the one observed in the case of Point Scan EDF. The explanation for this is that the smaller localization duration observed in the Line Scan EDF, allowed a shift (towards smaller event sizes) in the total Localization Cost Function.

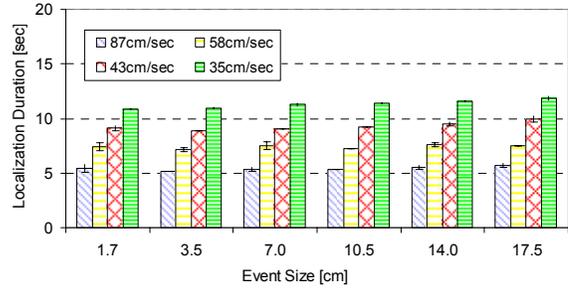


Figure 18. Localization Duration vs. Event Size for the Line Scan EDF

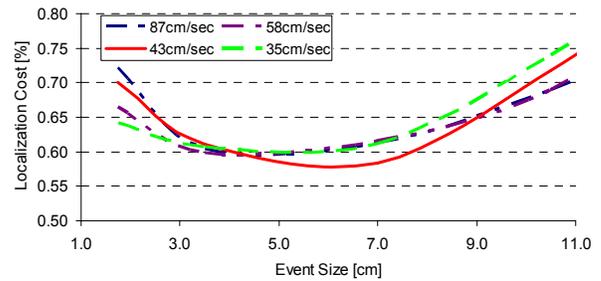


Figure 19. Cost Function vs. Event Size for the Line Scan EDF

During our experiments with the Line Scan EDF, we observed evidence of a bias in location estimation. The estimated locations for all sensor nodes exhibited different biases, for different event sizes. For example, for an event size of 17.5cm, the estimated location for sensor nodes was to the upper-left size of the actual location. This was equivalent to an “early” detection, since our scanning was done from left to right and from top to bottom. The scanning speed did not influence the bias.

In order to better understand the observed phenomena, we analyzed our data. Figure 20 shows the bias in the horizontal direction, for different event sizes (the vertical bias was almost identical, and we omit it, due to space constraints).

From Figure 20, one can observe that the smallest observed bias, and hence the most accurate positioning, was for an event of size 7cm. These results are consistent with the observed localization error, shown in Figure 17.

We also adjusted the measured localization error (shown in Figure 17) for the observed bias (shown in Figure 20). The results of an ideal case of Spotlight Localization system with Line Scan EDF are shown in Figure 21. The errors are remarkably small, varying between 0.1cm and 0.8cm, with a general trend of higher localization errors for larger event sizes.

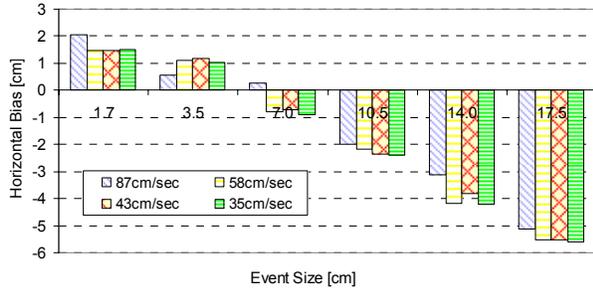


Figure 20. Position Estimation Bias for the Line Scan EDF

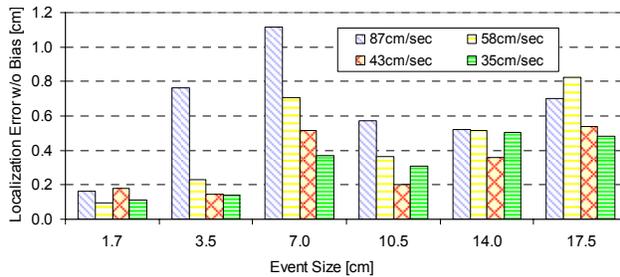


Figure 21. Position Estimation w/o Bias (ideal), for the Line Scan EDF

### 5.3 Area Cover - $\mu$ Spotlight system

In this experiment, we investigated how the number of bits used to quantify the entire sensor field, affected the localization accuracy. In our first experiment we did not use error correcting codes. The results are shown in Figure 22.

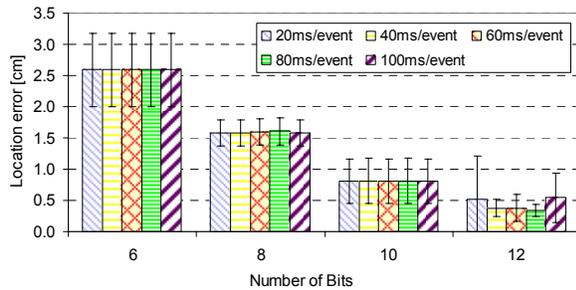


Figure 22. Localization Error vs. Event Size for the Area Cover EDF

One can observe a remarkable accuracy, with localization error on the order of 0.3-0.6cm. What is important to observe is the variance in the localization error. In the scenario where 12 bits were used, while the average error was very small, there were a couple of cases, where an incorrect event detection generated a larger than expected error. An example of how this error can occur was described in Section 3.4. The experimental results, presented in Figure 22, emphasize the need for error correction of the bit patterns observed and reported by the sensor nodes.

The localization duration results are shown in Figure 23. It can be observed that the duration is directly proportional with the number of bits used, with total durations ranging from 3sec, for the least accurate method, to 6-7sec for the most accurate. The duration of an event had a small influence on the total localization time, when considering the same scenario (same number of bits for the code).

The Cost Function dependency on the number of bits in the code, for  $\alpha=0.5$ , is shown in Figure 24. Generally, since the localization duration for the Area Scan can be extremely small, a higher accuracy in the localization is desired. While the Cost function achieves a minimum when 10 bits are used, we attribute the slight increase observed when 12 bits were used to the two 12-bit scenarios where larger than the expected errors were observed, namely 6-7mm (as shown in Figure 22).

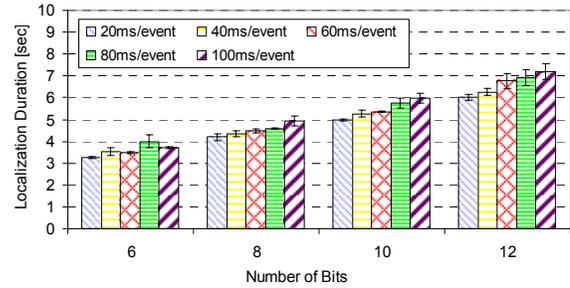


Figure 23. Localization Duration vs. Event Size for the Area Cover EDF

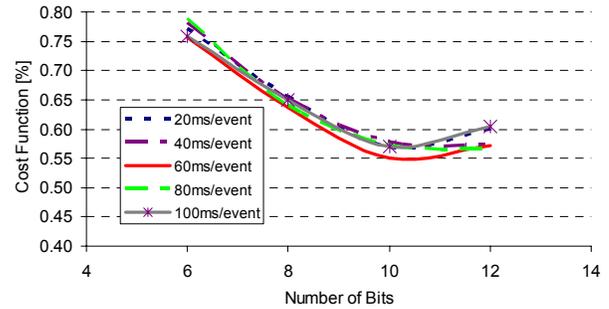


Figure 24. Cost Function vs. Event Size for the Area Cover EDF

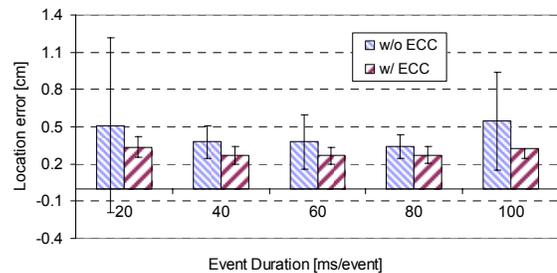


Figure 25. Localization Error w/ and w/o Error Correction

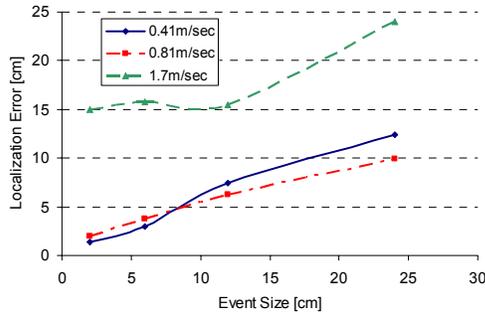
The two problematic scenarios (shown in Figure 22, where for 12-bit codes we observed errors larger than the event size, due to errors in detection) were further explored by using error correction codes. As described in Section 3.3, we implemented an extended Golay (24, 12) error correction mechanism in our location estimation algorithm.

The experimental results are depicted in Figure 25, and show a consistent accuracy. The scenario without error correction codes, is simply the same 12-bit code scenario, shown in Figure 22. We only investigated the 12-bit scenario, due to its match with the 12-bit data required by the Golay encoding scheme (extended Golay producing 24-bit codewords).

## 5.4 Point Scan - Spotlight system

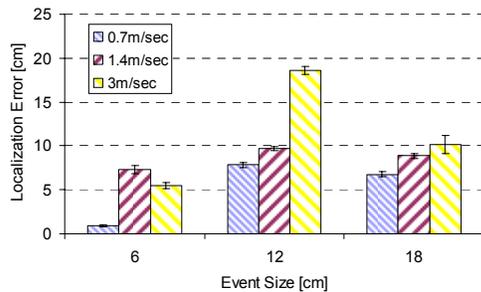
In this section we describe the experiments performed at a football stadium, using our Spotlight system. The hardware that we had available allowed us to evaluate the Point Scan technique of the Spotlight system. In our evaluation, we were interested to see the performance of the system at different ranges. Figures 26 and 27 show the localization error versus the event size at two different ranges: 46m and 170m.

Figure 26 shows a remarkable accuracy in localization. The errors are in the centimeter range. Our initial, manual measurements of the localization error were most of the time difficult to make, since the spot of the laser was almost perfectly covering the XSM mote. We are able to achieve localization errors of a few centimeters, which only range-based localization schemes are able to achieve [25]. The observed dependency on the size of the event is similar to the one observed in the  $\mu$ Spotlight system evaluation, and shown in Figure 14. This proved that the  $\mu$ Spotlight system is a viable alternative for investigating complex EDFs, without incurring the costs for the necessary hardware.



**Figure 26. Localization Error vs. Event Size for Spotlight system at 46m**

In the experiments performed over a much longer distance between the Spotlight device and sensor network, the average localization error remains very small. Localization errors of 5-10cm were measured, as Figure 27 shows. We were simply amazed by the accuracy that the system is capable of, when considering that the Spotlight system operated over the length of a football stadium. Throughout our experimentation with the Spotlight system, we have observed localization errors that were simply offsets of real locations. Since the same phenomenon was observed when experimenting with the  $\mu$ Spotlight system, we believe that with auto-calibration, the localization error can be further reduced.

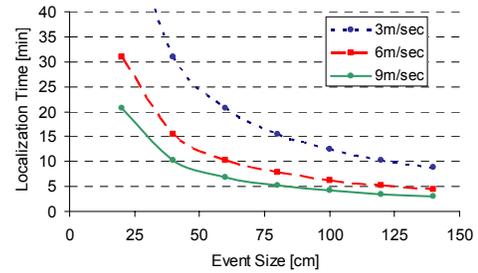


**Figure 27. Localization Error vs. Event Size for Spotlight system at 170m**

The time required for localization using the Spotlight system with a Point Scan EDF, is given by:  $t = (L * l) / (s * E_s)$ , where  $L$  and  $l$  are the dimensions of the sensor network field,  $s$  is the scanning

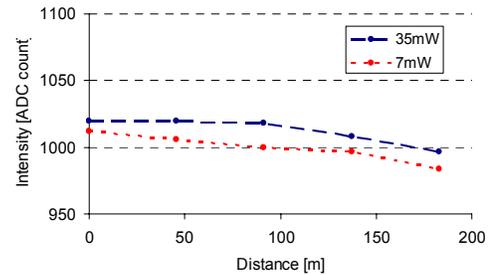
speed, and  $E_s$  is the size of the event. Figure 28 shows the time for localizing a sensor network deployed in an area of size of a football field using the Spotlight system. Here we ignore the message propagation time, from the sensor nodes to the Spotlight device.

From Figure 28 it can be observed that the very small localization errors are prohibitively expensive in the case of the Point Scan. When localization errors of up to 1m are tolerable, localization duration can be as low as 4 minutes. Localization durations of 5-10 minutes, and localization errors of 1m are currently state of art in the realm of range-free localization schemes. And these results are achieved by using the Point Scan scheme, which required the highest Localization Time, as it was shown in Table 1.



**Figure 28. Localization Time vs. Event Size for Spotlight system**

One important characteristic of the Spotlight system is its range. The two most important factors are the sensitivity of the photosensor and the power of the Spotlight source. We were interested in measuring the range of our Spotlight system, considering our capabilities (MTS310 sensor board and inexpensive, \$12-\$85, diode laser). As a result, we measured the intensity of the laser beam, having the same focus, at different distances. The results are shown in Figure 29.



**Figure 29. Localization Range for the Spotlight system**

From Figure 29, it can be observed that only a minor decrease in the intensity occurs, due to absorption and possibly our imperfect focusing of the laser beam. A linear fit of the experimental data shows that distances of up to 6500m can be achieved. While we do not expect atmospheric conditions, over large distances, to be similar to our 200m evaluation, there is strong evidence that distances (i.e. altitude) of 1000-2000m can easily be achieved. The angle between the laser beam and the vertical should be minimized (less than 45°), as it reduces the difference between the beam cross-section (event size) and the actual projection of the beam on the ground.

In a similar manner, we were interested in finding out the maximum size of an event, that can be generated by a COTS laser and that is detectable by the existing photosensor. For this, we

varied the divergence of the laser beam and measured the light intensity, as given by the ADC count. The results are shown in Figure 30. It can be observed that for the less powerful laser, an event size of 1.5m is the limit. For the more powerful laser, the event size can be as high as 4m.

Through our extensive performance evaluation results, we have shown that the Spotlight system is a feasible, highly accurate, low cost solution for localization of wireless sensor networks. From our experience with sources of laser radiation, we believe that for small and medium size sensor network deployments, in areas of less than 20,000m<sup>2</sup>, the Area Cover scheme is a viable solution. For large size sensor network deployments, the Line Scan, or an incremental use of the Area Cover are very good options.

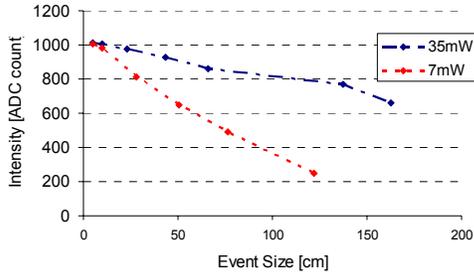


Figure 30. Detectable Event Sizes that can be generated by COTS lasers

## 6. OPTIMIZATIONS/LESSONS LEARNED

### 6.1 Distributed Spotlight System

The proposed design and the implementation of the Spotlight system can be considered centralized, due to the gathering of the sensor data and the execution of the Localization Function  $L(t)$  by the Spotlight device. We show that this design can easily be transformed into a distributed one, by offering two solutions.

One idea is to disseminate in the network, information about the path of events, generated by the EDF (similar to an equation, describing a path), and let the sensor nodes execute the Localization Function. For example, in the Line Scan scenario, if the starting and ending points for the horizontal and vertical scans, and the times they were reached, are propagated in the network, then any sensor in the network can obtain its location (assuming a constant scanning speed).

A second solution is to use anchor nodes which know their positions. In the case of Line Scan, if three anchors are present, after detecting the presence of the two events, the anchors flood the network with their locations and times of detection. Using the same simple formulas as in the previous scheme, all sensor nodes can infer their positions.

### 6.2 Localization Overhead Reduction

Another requirement imposed by the Spotlight system design, is the use of a time synchronization protocol between the Spotlight device and the sensor network. Relaxing this requirement and imposing only a time synchronization protocol among sensor nodes is a very desirable objective. The idea is to use the knowledge that the Spotlight device has about the speed with which the scanning of the sensor field takes place. If the scanning speed is constant (let's call it  $s$ ), then the time difference (let's call it  $\Delta t$ ) between the event detections of two sensor nodes is, in fact, an accurate measure of the range between them:  $d = s * \Delta t$ . Hence, the Spotlight system can be used for accurate ranging of the distance between any pair of sensor nodes. An important

observation is that this ranging technique does not suffer from limitations of others: small range and directionality for ultrasound, or irregularity, fading and multipath for Received Signal Strength Indicator (RSSI). After the ranges between nodes have been determined (either in a centralized or distributed manner) graph embedding algorithms can be used for a realization of a rigid graph, describing the sensor network topology.

### 6.3 Dynamic Event Distribution Function $E(t)$

Another system optimization is for environments where the sensor node density is not uniform. One disadvantage of the Line Scan technique, when compared to the Area Cover, is the localization time.

An idea is to use two scans: one which uses a large event size (hence larger localization errors), followed by a second scan in which the event size changes dynamically. The first scan is used for identifying the areas with a higher density of sensor nodes. The second scan uses a larger event in areas where the sensor node density is low and a smaller event in areas with a higher sensor node density.

A dynamic EDF can also be used when it is very difficult to meet the power requirements for the Spotlight device (imposed by the use of the Area Cover scheme in a very large area). In this scenario, a hybrid scheme can be used: the first scan (Point Scan) is performed quickly, with a very large event size, and it is meant to identify, roughly, the location of the sensor network. Subsequent Area Cover scans will be executed on smaller portions of the network, until the entire deployment area is localized.

### 6.4 Stealthiness

Our implementation of the Spotlight system used visible light for creating events. Using the system during the daylight or in a room well lit, poses challenges due to the solar or fluorescent lamp radiation, which generate a strong background noise. The alternative, which we used in our performance evaluations, was to use the system in a dark room ( $\mu$ Spotlight system) or during the night (Spotlight system). While using the Spotlight system during the night is a good solution for environments where stealthiness is not important (e.g. environmental sciences) for others (e.g. military applications), divulging the presence and location of a sensor field, could seriously compromise the efficacy of the system.

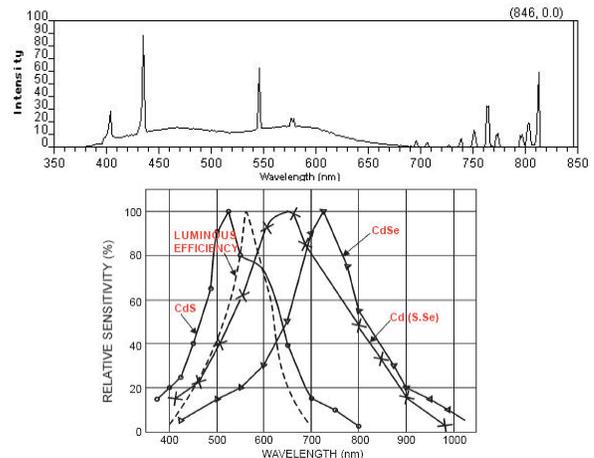


Figure 31. Fluorescent Light Spectra (top), Spectral Response for CdSe cells (bottom)

A solution to this problem, which we experimented with in the  $\mu$ Spotlight system, was to use an optical filter on top of the light

sensor. The spectral response of a CdSe photo sensor spans almost the entire visible domain [37], with a peak at about 700nm (Figure 31-bottom). As shown in Figure 31-top, the fluorescent light has no significant components above 700nm. Hence, a simple red filter (Schott RG-630), which transmits all light with wavelength approximately above 630nm, coupled with an Event Distribution Function that generates events with wavelengths above the same threshold, would allow the use of the system when a fluorescent light is present.

A solution for the Spotlight system to be stealthy at night, is to use a source of infra-red radiation (i.e. laser) emitting in the range [750, 1000]nm. For a daylight use of the Spotlight system, the challenge is to overcome the strong background of the natural light. A solution we are considering is the use of a narrow-band optical filter, centered at the wavelength of the laser radiation. The feasibility and the cost-effectiveness of this solution remain to be proven.

## 6.5 Network Deployed in Unknown Terrain

A further generalization is when the map of the terrain where the sensor network was deployed is unknown. While this is highly unlikely for many civil applications of wireless sensor network technologies, it is not difficult to imagine military applications where the sensor network is deployed in a hostile and unknown terrain. A solution to this problem is a system that uses two Spotlight devices, or equivalently, the use of the same device from two distinct positions, executing, from each of them, a complete localization procedure. In this scheme, the position of the sensor node is uniquely determined by the intersection of the two location directions obtained by the system. The relative localization (for each pair of Spotlight devices) will require an accurate knowledge of the 3 translation and 3 rigid-body rotation parameters for Spotlight's position and orientation (as mentioned in Section 3).

This generalization is also applicable to scenarios where, due to terrain variations, there is no single aerial point with a direct line of sight to all sensor nodes, e.g. hilly terrain. By executing the localization procedure from different aerial points, the probability of establishing a line of sight with all the nodes, increases. For some military scenarios [1] [12], where open terrain is prevalent, the existence of a line of sight is not a limiting factor. In light of this, the Spotlight system can not be used in forests or indoor environments.

## 7. CONCLUSIONS AND FUTURE WORK

In this paper we presented the design, implementation and evaluation of a localization system for wireless sensor networks, called Spotlight. Our localization solution does not require any additional hardware for the sensor nodes, other than what already exists. All the complexity of the system is encapsulated into a single Spotlight device. Our localization system is reusable, i.e. the costs can be amortized through several deployments, and its performance is not affected by the number of sensor nodes in the network. Our experimental results, obtained from a real system deployed outdoors, show that the localization error is less than 20cm. This error is currently state of art, even for range-based localization systems and it is 75% smaller than the error obtained when using GPS devices or when the manual deployment of sensor nodes is a feasible option [31].

As future work, we would like to explore the self-calibration and self-tuning of the Spotlight system. The accuracy of the system can be further improved if the distribution of the event, instead of a single timestamp, is reported. A generalization could

be obtained by reformulating the problem as an angular estimation problem that provides the building blocks for more general localization techniques.

## 8. ACKNOWLEDGEMENTS

This work was supported by the DARPA IXO office, under the NEST project (grant number F336616-01-C-1905) and by the NSF grant CCR-0098269. We would like to thank S. Cornwell for allowing us to run experiments in the stadium, M. Klopff for his assistance with optics, and anonymous reviewers and our shepherd, Koen Langendoen, for their valuable feedback.

## 9. REFERENCES

- [1] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulharni, U. Arumugam, M. Nesterenko, A. Vora, M. Miyashita, "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification and Tracking", in *Computer Networks* 46(5), 2004.
- [2] P. Bahl, V.N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System", in *Proceedings of Infocom*, 2000
- [3] M. Broxton, J. Lifton, J. Paradiso, "Localizing a Sensor Network via Collaborative Processing of Global Stimuli", in *Proceedings of EWSN*, 2005.
- [4] N. Bulusu, J. Heidemann, D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices", in *IEEE Personal Communications Magazine*, 2000.
- [5] P. Corke, R. Peterson, D. Rus, "Networked Robots: Flying Robot Navigation Using a Sensor Net", in *ISSR*, 2003.
- [6] L. Doherty, L. E. Ghaoui, K. Pister, "Convex Position Estimation in Wireless Sensor Networks", in *Proceedings of Infocom*, 2001
- [7] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, D. Culler, "Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events", in *Proceedings of IPSN*, 2005.
- [8] E. Elnahrawy, X. Li, R. Martin, "The Limits of Localization using RSSI", in *Proceedings of SECON*, 2004.
- [9] D. Fox, W. Burgard, S. Thrun, "Markov Localization for Mobile Robots in Dynamic Environments", in *Journal of Artificial Intelligence Research*, 1999.
- [10] D. Fox, W. Burgard, F. Dellaert, S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots", in *Conference on Artificial Intelligence*, 2000.
- [11] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, S. Wicker, "Complex Behaviour at Scale: An Experimental Study of Low Power Wireless Sensor Networks", in *Technical Report, UCLA-TR 01-0013*, 2001.
- [12] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, B. Krogh, "An Energy-Efficient Surveillance System Using Wireless Sensor Networks", in *Proceedings of Mobisys*, 2004.
- [13] T. He, C. Huang, B. Blum, J. A. Stankovic, T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks" in *Proceedings of Mobicom*, 2003.
- [14] L. Hu, D. Evans, "Localization for Mobile Sensor Networks", in *Proceedings of Mobicom*, 2004.
- [15] Y. Kwon, K. Mechtov, S. Sundresh, W. Kim, G. Agha, "Resilient Localization for Sensor Networks in Outdoor Environments", *UIUC Technical Report*, 2004.

- [16] K. Langendoen, N. Reijers, "Distributed Localization in Wireless Sensor Networks, A Comparative Study", in *Computer Networks* vol. 43, 2003.
- [17] K. Lorincz, M. Welsh, "MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking", in *Proceedings of Intl. Workshop on Location and Context-Awareness*, 2005.
- [18] M. Maroti, B. Kusy, G. Simon, A. Ledeczi, "The Flooding Time Synchronization Protocol", in *Proceedings of Sensys*, 2004.
- [19] D. Moore, J. Leonard, D. Rus, S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements" in *Proceedings of Sensys*, 2004.
- [20] R. Nagpal, H. Shrobe, J. Bachrach, "Organizing a Global Coordinate System for Local Information on an Adhoc Sensor Network", in *A.I Memo 1666. MIT A.I. Laboratory*, 1999.
- [21] D. Niculescu, B. Nath, "DV-based Positioning in Adhoc Networks" in *Telecommunication Systems*, vol. 22, 2003.
- [22] E. Osterweil, T. Schoellhammer, M. Rahimi, M. Wimbrow, T. Stathopoulos, L. Girod, M. Mysore, A. Wu, D. Estrin, "The Extensible Sensing System", *CENS-UCLA poster*, 2004.
- [23] B.W. Parkinson, J. Spilker, "Global Positioning System: theory and applications", in *Progress in Aeronautics and Astronautics*, vol. 163, 1996.
- [24] P.N. Pathirana, N. Bulusu, A. Savkin, S. Jha, "Node Localization Using Mobile Robots in Delay-Tolerant Sensor Networks", in *Transactions on Mobile Computing*, 2004.
- [25] N. Priyantha, A. Chakaborty, H. Balakrishnan, "The Cricket Location-support System", in *Proceedings of MobiCom*, 2000.
- [26] N. Priyantha, H. Balakrishnan, E. Demaine, S. Teller, "Mobile-Assisted Topology Generation for Auto-Localization in Sensor Networks", in *Proceedings of Infocom*, 2005.
- [27] A. Savvides, C. Han, M. Srivastava, "Dynamic Fine-grained localization in Adhoc Networks of Sensors", in *Proceedings of MobiCom*, 2001.
- [28] Y. Shang, W. Ruml, "Improved MDS-Based Localization", in *Proceedings of Infocom*, 2004.
- [29] M. Sichitiu, V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon", in *Proceedings of MASS*, 2004.
- [30] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, "Sensor Network-Base Countersniper System", in *Proceedings of Sensys*, 2004.
- [31] R. Stoleru, T. He, J.A. Stankovic, "Walking GPS: A Practical Solution for Localization in Manually Deployed Wireless Sensor Networks", in *Proceedings of EmNetS*, 2004.
- [32] R. Stoleru, J.A. Stankovic, "Probability Grid: A Location Estimation Scheme for Wireless Sensor Networks", in *Proceedings of SECON*, 2004.
- [33] R. Szcwcyk, A. Mainwaring, J. Polastre, J. Anderson, D. Culler, "An Analysis of a Large Scale Habitat Monitoring Application", in *Proceedings of Sensys*, 2004.
- [34] K. Whitehouse, A. Woo, C. Karlof, F. Jiang, D. Culler, "The Effects of Ranging Noise on Multi-hop Localization: An Empirical Study", in *Proceedings of IPSN*, 2005.
- [35] K. Whitehouse, D. Culler, "Calibration as Parameter Estimation in Sensor Networks", in *Proceedings of WSNA*, 2002.
- [36] P. Zhang, C. Sadler, S. A. Lyon, M. Martonosi, "Hardware Design Experiences in ZebraNet", in *Proceedings of Sensys*, 2004.
- [37] Selco Products Co. "Construction and Characteristics of CdS Cells", *product datasheet*, 2004