

# Localization in Multi-Modal Sensor Networks

Ryan Farrell <sup>1,3</sup>, Roberto Garcia <sup>2</sup>, Dennis Lucarelli <sup>3</sup>, Andreas Terzis <sup>2</sup>, I-Jeng Wang <sup>3</sup>

<sup>1</sup> *Department of Computer Science, University of Maryland  
AV Williams Bldg., College Park, MD 20742, farrell@cs.umd.edu*

<sup>2</sup> *Department of Computer Science, Johns Hopkins University  
3100 Wyman Park Drive, Baltimore, MD 21211, rgarcia,terzis@jhu.edu*

<sup>3</sup> *Applied Physics Lab, Johns Hopkins University  
11100 Johns Hopkins Road, Laurel, MD 20723, first.last@jhuapl.edu*

## Abstract

*We describe the design and implementation of solutions for localization problems in multi-modal wireless sensor networks. The problem of network self-localization, namely determining the positions of the nodes that comprise the network, is addressed optically using a set of pan-tilt-zoom (PTZ) cameras to search for a small light-source attached to each of the sensor nodes. Once the locations and headings of the network's nodes are estimated by the cameras, the network can be used to detect and estimate the location of objects traveling through it. Target localization is performed within the network, using information from magnetometers connected to the sensor nodes. We evaluate the performance of the proposed target localization algorithms through simulations and an implementation running on MicaZ motes. Simulation results show that the localization error for a 100-node network whose nodes are randomly deployed over an area of  $100 \times 100$  m, can be less than 10 cm. Moreover, our initial implementation results show that the median of the localization error for magnetic targets in a  $1\text{m} \times 1\text{m}$  field is 7.1 cm.*

## 1. INTRODUCTION

The use of networked cameras is gaining momentum for key applications such as surveillance, scientific monitoring, location-aware computing and responsive environments. The root cause is that it is easier for users to glean context and detail from imagery than other forms of measurements. Moreover, many applications from military deployments to biological field observation require an image to perform the sensing task. However, the drawbacks of ad-hoc vision networks are well known: cameras consume a lot of power and bandwidth and despite some recent hardware innovations (e.g. [12]) these limitations are still largely unresolved. Therefore cameras should be engaged only when absolutely necessary.

This paper presents a complete system that integrates vision networks with a non-imaging sensor network for the purpose of target localization and tracking. Specifically, the system initially uses a set of Pan-Tilt-Zoom (PTZ) cameras to localize the network's motes. Once localized, motes employ non-imaging sensors (e.g. magnetometers), to detect the existence and estimate the location of targets traversing the deployment area. The motes subsequently task the cameras to capture

images of those targets. We argue that the combined system offers performance and efficiencies that cannot be achieved by a single modality working in isolation. Our solution is also self contained in the sense that all operational aspects of the system are handled by the two sensing modalities and no additional elements are needed to perform target detection, acquisition and verification.

Target detection is one of the compelling applications driving much of the initial research in sensor networks. The pioneering work by Stankovic *et al.* [7], [8] and Arora *et al.* [1] among others proved that sensor networks can effectively detect targets over large geographic areas. While those networks used magnetometers, to the best of our knowledge, they were used only for target detection and not localization. On the other hand, we use information collected from the magnetometers to estimate the distance and the heading of ferrous targets as they move over the sensor field. Oh *et al.* have recently used magnetometers to localize targets in a multiple-target tracking algorithm [9]. That work assumed that mote locations are known in advance. Then, the target's location is estimated as the *centroid* of the locations of all the motes that detected the target, normalized by the signal strength measured by each of the motes. This approach has low accuracy because it assumes that the strength of the magnetic field is only a function of the distance between the target and the magnetometer, an assumption that is not valid as we explain in Section 3-A.

Given the importance of localization in sensor networks there have been many proposals for localization in the literature. Due to space constraints, we mention only those that are most closely related to our approach. Funiak *et al.* recently proposed a distributed camera localization algorithm to retrieve the locations and poses of ad-hoc placed cameras in a sensor network by tracking a moving object [5]. Along the same lines, Rahimi *et al.* used a network of cameras to track the locations of moving targets while at the same time calibrating the camera network [11]. In our setting, cameras are used only during the initial phase to estimate mote locations. Furthermore, instead of tracking the trajectory of the target using the cameras, we use non-imaging sensors to estimate the location of magnetic targets. The proposed localization algorithm is versatile enough to run at a central location or in a distributed fashion on the actual motes.

Lymberopoulos *et al.* recently considered the use of epipolar geometry to perform localization, using LEDs to assist the visual detection of sensors [2]. We also use LEDs connected to the motes in our network to indicate their presence to the associated cameras. However, unlike the solution proposed by Lymberopoulos *et al.* we do not require any distance information from the cameras to estimate the mote locations. Even though the only information available in our case are the angular locations of the motes relative to each camera, our solution has comparable localization error.

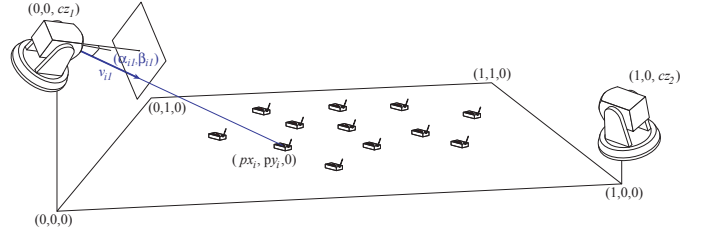
Probably the work most closely related to ours is the Spotlight system [13]. This system employs an asymmetric architecture in which motes do not require any additional hardware for localization while a single Spotlight device uses a steerable laser light source to illuminate the deployed motes. The primary difference between our approach and that work lies in our focus on truly ad-hoc deployment. We assume that the cameras will have unknown location and orientation whereas the Spotlight system relies on accurate knowledge of both camera position and orientation. Furthermore, individual motes in Spotlight must know when the laser light source is supposed to illuminate them, a design that requires time synchronization.

## 2. OPTICAL SELF-LOCALIZATION

The successful integration of the camera and sensor networks requires that the sensor field be located within the field of view of an ad-hoc deployment of cameras. One solution would be to localize the nodes through “traditional” means and solve for the transformation between the two coordinate systems. Instead, we solve for all location variables (mote and camera locations/orientations) simultaneously. This approach establishes a common frame of reference, such that the location of objects detected by a camera can be described in the spatial terms of the network and vice-versa. The use of pan-tilt cameras allows the entire sensor network space to be mapped into the camera’s pan-tilt parametrization. Arguably the greatest advantage of the system we propose is the ability to rapidly deploy a wireless sensor network in an ad-hoc fashion *without* regard to the cameras’ fields-of-view.

Let us consider a planar network of wireless sensor nodes,  $\mathbf{N} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ , where  $\mathbf{p}_i = (px_i, py_i, 0)$  denotes the location of node  $i$ . For the cameras, we must describe both position and orientation. With each camera  $j$ , we therefore associate a position  $\mathbf{c}_j = (cx_j, cy_j, cz_j)$  and a 3-DOF rotation  $\Theta_j = (\theta_{x_j}, \theta_{y_j}, \theta_{z_j})$ . As we focus on ad-hoc deployment, the values for  $\mathbf{p}_i$ ,  $\mathbf{c}_j$ , and  $\Theta_j$  are all unknown.

Without any notion of geodetic location or scale, the best that we can hope to recover is a local and relative coordinate frame. Rather than pick some arbitrary scale, we have chosen to normalize the coordinate frame based on the distance between the two cameras. The camera locations are thus  $\mathbf{c}_1 = (0, 0, cz_1)$  and  $\mathbf{c}_2 = (1, 0, cz_2)$ . The camera elevations remain unknown as do the orientation parameters. Figure 1 depicts the normalized coordinate frame (note that while the nodes



**Fig. 1: Normalized Coordinate Frame.** Illustrates the sensor network  $\mathbf{N} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ ,  $\mathbf{p}_i = (px_i, py_i, 0)$ , with two active cameras. The illustration also depicts node  $\mathbf{p}_i$  localized at pan-tilt coordinates  $(\alpha_{i1}, \beta_{i1})$  relative to the first camera.

are pictured within the unit square, they are not constrained to be positioned there).

The only values we are able to measure are the angular location of each node relative to each camera. We parameterize a camera’s pan-tilt space with coordinates  $(\alpha, \beta)$ , in radians. Camera  $j$  observes node  $i$  at location  $(\alpha_{ij}, \beta_{ij})$ , indicating that camera  $j$  would need to pan by  $\alpha_{ij}$  and tilt by  $\beta_{ij}$  in order to aim directly at node  $i$ .

We now derive a relationship between these measurements and the unknown values we wish to determine. In considering the relationship between camera  $j$  and node  $i$ , we define the offset vector  $\mathbf{v}_{ij} = \mathbf{p}_i - \mathbf{c}_j$  for convenience. As a vector can also be decomposed into the product of its magnitude with a unit vector in the same direction, we can write  $\mathbf{v}_{ij} = \|\mathbf{v}_{ij}\| \cdot \hat{\mathbf{v}}_{ij}$ . The unit vector  $\hat{\mathbf{v}}_{ij}$  can be expressed in two ways. The first expression is obtained by simply dividing  $\mathbf{v}_{ij}$  by its magnitude,  $\|\mathbf{v}_{ij}\|$ , yielding

$$\hat{\mathbf{v}}_{ij} = \frac{\mathbf{v}_{ij}}{\|\mathbf{v}_{ij}\|} = \frac{\mathbf{p}_i - \mathbf{c}_j}{\|\mathbf{v}_{ij}\|} = \frac{1}{\|\mathbf{v}_{ij}\|} \begin{pmatrix} px_i - cx_j \\ py_i - 0 \\ 0 - cz_j \end{pmatrix}. \quad (1)$$

The second expression is found by rotating the optical axis of camera  $j$  such that it passes through  $\mathbf{p}_i$ . A standard camera model defines the local coordinate frame with the camera looking downward along the  $z$ -axis at an image plane parallel to the  $x$ - $y$  plane. The optical axis thus points in the  $-\hat{z}$  direction in the local coordinate frame. To transform from the camera’s local frame into world coordinates, we must apply the three-axis rotation defined by  $\Theta_j$ . In order to point the optical axis at  $\mathbf{p}_i$ , we pan and tilt the camera with the rotations corresponding to  $(\alpha_{ij}, \beta_{ij})$ . At length, we find the second expression

$$\hat{\mathbf{v}}_{ij} = \underbrace{R_{\alpha_{ij}} R_{\beta_{ij}}}_{\text{pan-tilt}} \cdot \underbrace{R_{\Theta_j}}_{\text{world}} \cdot \underbrace{(-\hat{z})}_{\text{local}}. \quad (2)$$

Now, we equate these two expressions for  $\hat{\mathbf{v}}_{ij}$ . Merging Eq.1, in terms of  $\mathbf{p}_i$  and  $\mathbf{c}_j$ , with Eq.2, in terms of  $\alpha_{ij}$ ,  $\beta_{ij}$  and  $\Theta_j$ , we derive a relationship between the observations and the unknowns we wish to solve for:

$$\frac{1}{\|\mathbf{v}_{ij}\|} \begin{pmatrix} px_i - cx_j \\ py_i - 0 \\ 0 - cz_j \end{pmatrix} = R_{\alpha_{ij}} R_{\beta_{ij}} \cdot R_{\Theta_j} \cdot \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}. \quad (3)$$

where  $\|\mathbf{v}_{ij}\| = \sqrt{(px_i - cx_j)^2 + py_i^2 + cz_j^2}$ .

This relationship embodies three constraints per node-camera pair, one each in  $x$ ,  $y$  and  $z$ . As only two measurements are used, however, the three constraints cannot be independent. So, each node-camera pair contributes only two to the rank of the system for a total rank of  $4n$ . We have  $2n + 8$  unknowns to solve for: elevation and 3-DOF orientation (4 parameters) for each camera, and the 2-D position of each of the  $n$  nodes. For the system to be sufficiently constrained, we must have  $4n \geq 2n + 8$  which yields  $n \geq 4$ . Thus, measurements from at least 4 nodes must be used or the resulting system of nonlinear equations (constraints derived as the components of Eq.3) will be under-determined. However, as we will see in Section 4, using larger numbers of nodes is preferable, as it helps mitigate the presence of noise in the measurements, greatly improving localization accuracy.

So far, we have derived a system of geometric constraints that govern the relationship between the relative configuration of nodes and cameras and the angular locations of the nodes measured by each camera. The localization problem then becomes a problem in nonlinear optimization. An analytical expression of the system's Jacobian was obtained and used to instantiate a Newton-Raphson solver. As with other techniques, Newton's method is prone to convergence at local extrema depending on the initial search location. Our implementation is therefore based on a globally convergent Newton's approach [10] which uses multiple restarts and attempts to backtrack if it reaches a non-global optimum.

### 3. TARGET LOCALIZATION

After the network has self-localized, it can estimate the location of targets moving through the sensor field using its non-imaging sensors. We chose to use magnetometers as our non-imaging sensors. Magnetometers detect the presence of ferrous targets through the changes they create to the earth's magnetic field. This choice was based on a number of factors. First, it has been experimentally proven that magnetometers can detect the presence of metal objects at distances up to tens of feet [1], [8]. Second, magnetometers can inherently focus only on 'interesting' objects moving through the sensor field (*e.g.* persons carrying large metallic objects). Finally, magnetometers can be used to classify targets based on their magnetic signatures.

#### A. Magnetic Signal Strength Model

Given that we use magnetometers to detect and localize targets, we model these targets as moving magnets. At a high level, the magnetometers measure the strength and the direction of the magnetic field generated by the target and translate those measurements to a heading and distance. This translation requires a model that describes the magnetic field at an arbitrary location in space around the magnet. For this purpose, we adopted the model described in [6] (p.246), which describes the strength of the magnetic field created by a magnetic dipole  $m$  at distance  $r$  and angles  $\theta, \phi$  from the location of the dipole, as:

$$B(\vec{r}) = \frac{\mu_0 m}{4\pi r^3} (2 \cos \theta \hat{r} + \sin \theta \hat{\theta}) = B_r + B_\theta \quad (4)$$

where  $\hat{r}, \hat{\theta}$  are unit size vectors in the direction of the radius  $r$  and angle  $\theta$  respectively. The parameter  $\mu_0$  is the permeability of free space, while  $m$  describes the dipole's magnetic moment, which indicates the dipole's relative strength.

#### B. Model Validation

We validated the applicability of Eq.4 by comparing its predictions with measurements collected by a MicaZ mote equipped with the MTS310 sensor board from Crossbow.

The MTS310 sensor board includes an HMC1002 2-axis magnetometer manufactured by Honeywell. Each of the two axes of this sensor is a Wheatstone bridge, whose resistance changes according to the magnetic field applied to it. This change in resistance causes a change in the voltage differential of the bridge which can be measured by the mote's Analog to Digital Converter (ADC) circuit. According to the manufacturer's specification, the recorded voltage differential  $\Delta V$  generated by the magnetometer is equal to  $\Delta V = s \cdot B + o$ , where  $s$  is the magnetometer's sensitivity (in mV/Gauss),  $B$  is the strength of the applied magnetic field (in Gauss) and  $o$  is the bridge offset (in mV).

While we can measure  $\Delta V$  directly, we are interested in the magnetic field strength  $B$ . However to derive  $B$  we need to estimate the values of  $s$  and  $o$  which are device-specific. To do so, we placed the magnetometer inside a Helmholtz coil, which is a device that can generate magnetic fields of known  $B$ . We were able in this way to experimentally validate the accuracy of the above formula and estimate the  $s$  and  $o$  parameters.

We performed two additional experiments to test the validity of Eq.4. First, we varied the distance  $r$  between the mote magnet, while keeping  $\theta$  constant. Next, we kept  $r$  constant while rotating the magnet around the magnetometer. The results of these experiments, included in the associated technical report [4], showed that there is a good match between the model and the experimental values.

#### C. Simplified Localization Algorithm

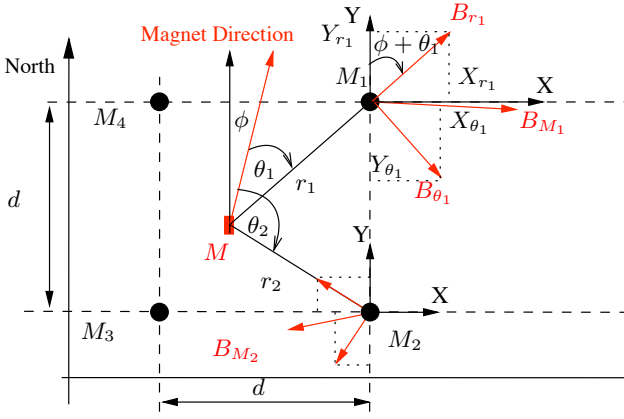
We use Eq.4 to estimate the location of a magnetic target located inside a lattice of motes  $M_i$ ,  $d$  distance units apart from each other. Figure 2 illustrates such a lattice with four magnetometers and one magnetic target. For now, we assume that all motes are arranged in such a way that their  $Y$ -axes are all aligned and point North. The exact direction is not important, but rather all magnetometers' directions must be aligned. We will remove this restriction in Section 3-D.

The strength of the magnetic field recorded at mote  $M_i$  has two components recorded by the magnetometer's  $X$ - and  $Y$ -axes, respectively. From Figure 2 one can see that  $X_i$  is the sum of  $X_{r_i}$  and  $X_{\theta_i}$  which are the projections of  $B_{r_i}$  and  $B_{\theta_i}$  on the  $X$ -axis.

$$X_i = X_{r_i} + X_{\theta_i} \quad (5)$$

$$X_i = B_{r_i} \sin(\phi + \theta_i) + B_{\theta_i} \cos(\phi + \theta_i) \quad (6)$$

$$X_i = \frac{C_i}{2} (3 \sin(\phi + 2\theta_i) + \sin \phi) \quad (7)$$



**Fig. 2:** Magnetic target and magnetic field strength measured at magnetometers  $M_1$  and  $M_2$ .

where  $C_i = \frac{\mu_0 m}{4\pi r_i^3}$ . Likewise,  $Y_i = \frac{C_i}{2} (3 \cos(\phi + 2\theta_i) + \cos \phi)$ .

Since each mote measures its own  $X_i, Y_i$ , these equations offer four nonlinear constraints for each mote pair. We have however five unknowns:  $r_1, r_2, \phi, \theta_1, \theta_2$ . Fortunately, there is another relationship that we can utilize:  $d^2 = r_1^2 + r_2^2 - 2r_1r_2 \cos(\theta_2 - \theta_1)$ , which is none other than the generalized Pythagorean theorem for triangle  $MM_1M_2$ .

Analogous sets of equations can be written for all other mote pairs that detect the magnetic target  $M$ . We found it helpful in practice to add explicit constraints to ensure that the location corresponding to  $M_1 + \vec{r}_1$  is the same as that of  $M_2 + \vec{r}_2$ . To derive these constraints, we simply equate the  $X$ - and  $Y$ - components as follows:

$$r_1 \sin(\phi + \theta_1) = r_2 \sin(\phi + \theta_2) \quad (8)$$

$$d - r_1 \cos(\phi + \theta_1) = -r_2 \cos(\phi + \theta_2) \quad (9)$$

For  $k$  magnetometers detecting the target, we thus derive  $2k + 3\binom{k}{2}$  nonlinear equations: two equations for the  $X$  and  $Y$  magnetometer readings of each node and three equations (the generalized Pythagorean equation and the two “consistency” constraints) for each magnetometer pair.

#### D. Heading Calibration

The algorithm presented in the previous section assumes that all the motes are aligned. However, we can remove this limitation and support arbitrary mote headings  $\phi_i$  as long as these headings are known. Moreover, we can estimate the mote’s orientation by taking advantage of the earth’s magnetic field and treating the magnetometer as an electronic compass. Assuming that the earth’s field points north (with some offset depending on the location where the system is deployed), one can derive the heading of the mote  $\phi_i$  by calculating the arctan of the  $B_x/B_y$  ratio.

#### E. Generalized Localization Problem

Given mote headings  $\phi_i$  estimated by the mechanism described above, the generalized version of the localization problem can be solved.

Denote the target’s coordinates by  $(x_m, y_m)$  and its orientation as before by  $\phi_m$ . Each magnetometer  $M_i$  is located at  $(x_i, y_i)$  and has orientation  $\phi_i$ . The angle from the target to each magnetometer relative to the target’s axis is denoted by  $\theta_i$ . Instead of adding the magnetometer orientation as a third unknown in addition to  $r_i$  and  $\theta_i$ , we now express the quantities  $r_i$  and  $\theta_i$  in terms of the known position  $p_i = (x_i, y_i)$  and the unknown target parameters  $p_m = (x_m, y_m)$  and  $\phi_m$ , as  $r_i = \|p_i - p_m\|$ ,  $\theta_i = \arcsin\left(\frac{x_i - x_m}{r_i}\right) - \phi_m$ .

Thus, instead of using three unknowns for each magnetometer, we have only one unknown,  $\phi_i$ . The equations relating the magnetometer readings are relatively unchanged, differing only by the incorporation of  $\phi_i$ :

$$B_{x_i} = \frac{C_i}{2} (3 \sin(2\theta_i + \phi - \phi_i) + \sin(\phi - \phi_i)) \quad (10)$$

$$B_{y_i} = \frac{C_i}{2} (3 \cos(2\theta_i + \phi - \phi_i) + \cos(\phi - \phi_i)) \quad (11)$$

where  $C_i$  is defined as before.

Interestingly, the generalized Pythagorean equation and the two “consistency” constraints are no longer useful as they contribute no rank to the Jacobian matrix used to solve the system of non-linear equations.

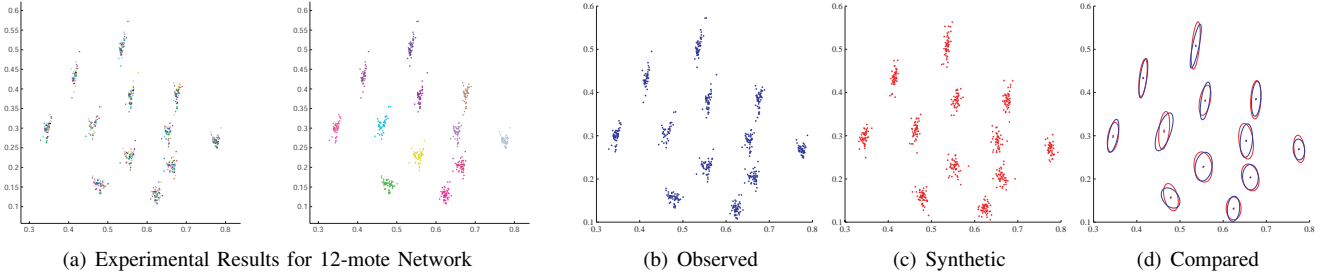
### 4. SELF-LOCALIZATION RESULTS

The primary objectives in evaluating the proposed self-localization technique are to measure localization accuracy and assess its scalability. We first employed a network of 12 MicaZ motes to examine the localization accuracy of our approach under realistic conditions. To overcome the limitations of this small network, we used a simulated network of 100 motes and found that larger networks offer improved accuracy despite their modest increase in computation overhead.

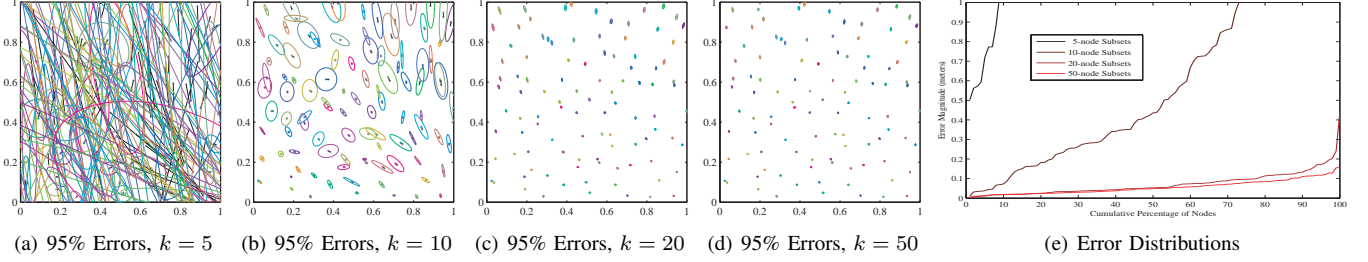
#### A. Experimental Deployment

Our experimental platform consists of a network of 12 MicaZ motes equipped with omni-directional LEDs and two PTZ cameras: a Sony EVI-D100 and a Sony SNC-RZ25N. Our application software, written primarily in Java, runs on a standard laptop, using JNI to integrate native code for image processing and for controlling the EVI-D100. Communication with the sensor network is facilitated by another mote wired directly to the laptop.

To self-localize a network of  $n$  nodes, we select a subset of  $k$  nodes, namely  $S_k = \{\mathbf{p}_{i_1}, \dots, \mathbf{p}_{i_k}\}$ . The nodes in  $S_k$  are sequentially interrogated, each ascending an omni-directional LED which the cameras use to pinpoint the node’s location in their individual pan-tilt parameterization. After finding the pan-tilt location for each node  $\mathbf{p}$ , the corresponding nonlinear system of equations is solved numerically, yielding location estimates for  $\forall \mathbf{p} \in S_k$ . Many such subsets,  $S_k^{(1)}, S_k^{(2)}, \dots, S_k^{(T)}$ , are randomly chosen and location estimates for each subset  $S_k^{(t)}$  is determined as above. When this process completes, we have several location estimates for each node ( $\frac{Tk}{n}$  expected) and average them to derive the final location estimate for each given node.



**Fig. 3: Experimental 12-mote Results and Empirically-validated Noise Model.** (a) depicts simultaneous plots of the localization results from 100 randomly selected 6-mote subsets, on the left colored by subset ID, on the right by mote ID. (b) plots the solutions for 100 subsets using actual measurements obtained from the experimental system (same data displayed in Figure (a)). (c) plots the solutions for 100 subsets using simulated measurements obtained using the synthetic noise model. (d) shows a comparison between the the observed and the synthetic data. The mean location for each mote along with the error ellipse indicating 95%-confidence are depicted. As in (c) and (d), blue is used for the observed data, red for the synthetic data.



**Fig. 4: Simulated Large-scale (100 Node) Network Results.** (a)-(d) present the localization estimates (mean of the solutions obtained) for each node, together with the ellipse of 95% confidence. Each node is plotted in a unique color and a black error line, most easily discerned in (a), is drawn between each node's ground truth location and the estimated location. A comparison of the error distributions is shown in (e), highlighting the improvement in accuracy realized by using larger subsets.

Sample results are presented in Figure 3(a) for our network of 12 MicaZ motes. Each dot in these graphs shows an estimated mote location over the unit square  $[0, 1]^2$ , as calculated by the system. In Figure 3(a), the localization results for subsets of size  $k = 6$  are shown, on the left colored by subset ID and on the right colored by mote ID. While the absence of ground truth localization prevents us from determining precise localization errors, we can see how the solutions for random subsets collectively suggest an accurate location estimate.

Though unable to quantify the localization accuracy of our experiments due to the lack of ground truth locations (and camera orientations), we nonetheless model the observed noise in order to produce similar measurements synthetically. Instead of generating the exact pan-tilt measurements that would result from the ground truth locations, we apply an angular perturbation  $\sim N(0, \zeta)$  to each measurement, akin to the model in [3], using  $\zeta = 0.1^\circ$ . We generate synthetic measurements using the mean locations derived from the experimental network using subsets of size  $k = 6$ . A comparison between the real and synthetic data is presented in Figures 3(b)-3(d). The synthetic noise model closely reflects the error characteristics of the observed data.

### B. Simulation

To demonstrate the improved localization accuracy obtained with larger subsets, we use a larger simulated network. Simulations provide us with ground truth for mote locations, essential to properly addressing the question of localization error. As noted before, localization accuracy is associated not with the

**TABLE 1: MEAN ERRORS FOR  $100m \times 100m$  100-NODE NETWORK.**

Subset Size	50% Confidence	95% Confidence
5 nodes	6.70 m	96.25 m
10 nodes	43.57 cm	2.14 m
20 nodes	5.26 cm	17.19 cm
50 nodes	4.85 cm	11.73 cm

size of the network, but with the subset size  $k$ . So, whether the network has tens, hundreds or thousands of nodes, the localization accuracy will be same for a given subset size  $k$  assuming the number of estimates per node is fixed. For a fixed  $k$ , the required computation time grows linearly with the size of the network. The reason is that as  $n$  grows, the number of random subsets processed,  $T$ , must increase proportionally.

### C. Empirically Verified Noise Model

Our simulated network distributed 100 nodes uniformly at random within a unit square  $[0, 1]^2$ . The cameras were placed at the same normalized heights as in our experimental setup, the first above the origin  $(0, 0)$ , the second above  $(0, 1)$ . The noise model used is the same described in Section 4-C. Figure 4 presents results for this 100-node network obtained using  $k = 5, 10, 20, 50$ . To articulate these results in real-world terms, if the deployment area were  $100m \times 100m$ , then these errors would be as depicted in Figure 4(e) and Table 1.

## 5. TARGET LOCALIZATION RESULTS

### A. Simulation

Prior to conducting any experiments with actual magnets, we evaluate the localization accuracy and orientation estimates

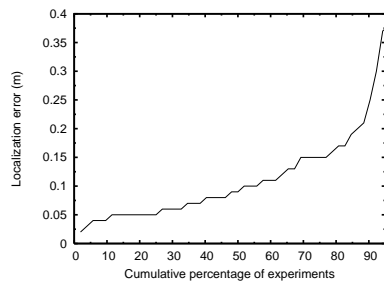


Fig. 5: Target localization error distribution.

through simulation. To do so, we generate noisy magnetometer readings by adding white noise to the magnetic field strength provided by Eq. 4. Certain intrinsic properties of the system emerged not through theoretical analysis, but rather through simulation. One such property is the inherent ambiguity in orientation. Magnetometer readings will be identical if both the target and network nodes are rotated by 180 degrees. Due to space constraints we can only summarize these simulations by stating that when six or more magnetometers in general position collaborate to localize the target, over %90 of all simulations result in errors less than 0.1 for targets placed in the unit square. These results are comparable to the localization accuracy of three magnetometers aligned on a grid. The interested reader can find details regarding these results in [4].

### B. Experimental results

We designed an experiment to test mote localization in the following way. First, we selected random locations on a level surface to place a set of six pre-calibrated motes. We then selected a set of fifty target locations by placing a magnet of known strength at each of these locations. The resulting Euclidean distance between the targets and the network's motes was 40 cm on the average.

Figure 5 presents the distribution of localization errors across all experiments. The majority of cases result in low localization error (the median and average localization errors are 13.8cm and 9 cm respectively). There are however a few cases with high localization error, indicating that some positions did not localize well. These positions are at the outer fringes of the magnetometers' sensing range where the field strength measurements are inaccurate.

## 6. SUMMARY

We present a comprehensive solution to the problem of localization in multi-modal sensor networks. First, two PTZ cameras are used to estimate the locations of the network's motes. Unlike previous solutions, only angular information from the cameras is used for self-localization. Results from an experimental implementation, coupled with results from larger simulated networks indicate that the accuracy of the proposed localization algorithm increases with the subset size  $k$ , the number of the nodes localized together. Techniques to complete the scalability assessment of the approach are

left for future work. One approach would be to use several pairs of cameras to localize overlapping subsets of sensors and compute the coordinate transformation between solutions. These considerations, coupled with the fact that localization accuracy obtained by one pair of cameras does not degrade as the size of the network increases, make the proposed approach very attractive for large-scale WSN deployments.

Once the network has self-localized, it can detect the presence and estimate the location of ferrous target moving over the sensor field, with the help of magnetometers connected to the network's motes. We present two variations of the target localization algorithm, one which assumes that motes are deployed on a regular lattice and a generalized version that works with arbitrary (but known) mote headings  $\phi_i$ . We have implemented this second version of the localization algorithm on the MicaZ motes of our testbed and evaluated its accuracy through experiments with actual magnets as well as large scale simulations. We find that as the number of motes detecting the target increases, the generalized version performs almost as well as the version requiring regular mote formations. Furthermore, early results from our experimental evaluation indicate that it is possible to locate magnetic targets with average localization error in the order of a few cm in a 1m x 1m field.

## REFERENCES

- [1] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y-R. Choi, T. Herman, S. S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks*, pages 605–634, 2004.
- [2] A. Barton-Sweeney, D. Lymberopoulos, and A. Savvides. Sensor Localization and Camera Calibration in Distributed Camera Sensor Networks. In *Proceedings of IEEE BaseNets*, Oct 2006.
- [3] A.O. Ercan, D.B. Yang, A. El Gamal, and L. Guibas. Optimal placement and selection of camera network nodes for target localization. In *DCOSS*, pages 389–404, 2006.
- [4] Ryan Farrell, Roberto Garcia, Dennis Lucarelli, Andreas Terzis, and I-Jeng Wang. Localization in multi-modal sensor networks. Technical Report HINRG-01-06-07, Hopkins Internetworking Research Group, June 2007.
- [5] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar. Distributed localization of networked cameras. In *IPSN*, pages 34–42, 2006.
- [6] D. J. Griffiths. *Introduction to Electrodynamics*. Prentice Hall, 1999.
- [7] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Qi. Cao, T. He, J. A. Stankovic, T. Abdelzaher, and B. Krogh. Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments. In *Sensys 2005*, November 2005.
- [8] T. He, S. Krishnamurthy, J. A. Stankovic, T. F. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. An Energy-Efficient Surveillance System Using Wireless Sensor Network. In *MobiSys*, 2004.
- [9] S. Oh, L. Schenato, P. Chen, and S. Sastry. A Scalable Real-Time Multiple-Target Tracking Algorithm for Sensor Networks. Technical Report UCB/ERL M05/9, University of California, Berkeley, 2005.
- [10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.
- [11] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Proceedings of CVPR*, 2004.
- [12] M. Rahimi, R. Baer, O. Iroez, J. Garcia, D. Estrin J. Warrior, and M. Srivastava. Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks. In *SenSys*, 2005.
- [13] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke. A high-accuracy, low-cost localization system for wireless sensor networks. In *SenSys*, pages 13–26, 2005.