Subspace Techniques for Vision-Based Node Localization in Wireless Sensor Networks

Huang Lee, Laura Savidge, Hamid Aghajan
Wireless Sensor Networks Lab
Department of Electrical Engineering
Stanford University, Stanford, CA 94305

***Abstract*— We present novel techniques for localization of nodes in a wireless image sensor network. Based on visual observations of a moving object by the network nodes, the proposed techniques employ simple image processing functions to produce equations that contain the node positions and orientation angles as the unknown parameters. Observations made at the nodes relate the position of the observed object to the physical coordinates of the node via the mapped position of the object in the node's image plane. In one formulation of the problem, multiple observations by a network node from a moving beacon with known coordinates result in a system of equations with a rank-deficient matrix. Hence, the solution for the desired node coordinates lies in the null space of the data matrix. In a second formulation, a different configuration of image sensor deployment with more degrees of freedom results in a least-squares solution for the unknown parameters. In a third formulation, multiple observations are made at each node from a target which moves at a fixed velocity vector. The solution to this problem formulation is also shown to correspond to the null space of the data matrix. The proposed algorithms are based on in-node processing and hence are scalable to large networks. Simulation and experimental results are provided in the paper.**

I. Introduction

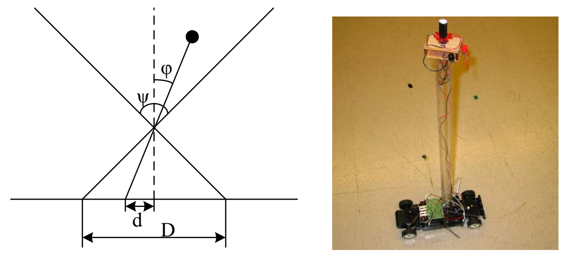
Many applications in wireless sensor networks rely on the knowledge of sensor positions. Node localization is hence a fundamental task in the deployment of sensor networks [1] [2] [3] [4]. Recently, research on image sensor networks has received large interest; however, only limited study of the localization problem has been reported for these networks [5] [6]. In the localization methods proposed in this paper, each node in the network derives equations from visual observations of a moving object in the environment. We consider two cases in which the moving object is a friendly beacon that knows and broadcasts its coordinates as it travels through the network plane. During the localization process, the radios of the nodes receive packets broadcasted by the beacon. We also consider another case based on observations of a moving target with unknown positions but assumed to have a motion model with a fixed velocity vector. In the proposed methods, network nodes estimate their coordinates and orientation angle with respect to the coordinate system defined by the moving object. This makes the proposed techniques scalable to large networks.

II. Image Plane Model

We will present 3 vision-based localization methods in this paper. These methods differ in the type of observations used and the orientation of the image planes of the nodes with respect to the 2-D coordinates plane in which the moving object travels. These methods are listed in Table I.

TABLE I
Observation types and image plane orientations.

Method	Observation Type	Image Plane Orientation
Method 1	Beacon agent	Perpendicular to motion plane
Method 2	Beacon agent	Parallel to motion plane
Method 3	Target with fixed velocity	Perpendicular to motion plane



(a) (b)

Fig. 1. (a) Pinhole camera model. (b) Moving beacon used in method 1.

We use a pinhole camera model (Fig. 1(a)) described by

$$\varphi = \tan^{-1}\left(\frac{2d}{D}\tan\left(\frac{\psi}{2}\right)\right), \tag{1}$$

where φ represents an observed object's angular displacement from the camera's orientation direction, d is the distance from the center of the image plane in pixels, D is the image plane dimension in pixels and ψ is the field-of-view angle of the image sensor. D and ψ are physical parameters specified by the image sensor manufacturers or can be measured.

III. Method 1

This method utilizes observations of a beacon agent (Fig. 1(b)) moving through the network plane. The beacon knows its own coordinates and broadcasts them at several stops along its path. Upon receiving the broadcasts, image sensors capture images from which they extract the location of the beacon if it appears in their field of view. The angular offset from the image sensor orientation is found through simple image processing and by using (1). Considering the situation shown in Fig. 2(a), with one node and two measurements, the unknown node location t and orientation θ can be related to positions of the beacon at locations s_i and s_k - in the absence of noise - via

$$t = s_i - \lambda_i e^{-j(\theta+\phi_i)} = s_k - \lambda_k e^{-j(\theta+\phi_k)} \tag{2}$$

where s_i represents the beacon's coordinates, ϕ_i is the angular offset of the beacon in the image plane, and λ_i the distance between the node and the beacon for observation i. Besides the unknown value θ, for each added measurement one new unknown (λ_i) is added. We can form $(N-1)$ independent equalities of the following form between pairs of observations:

$$s_i - \lambda_i e^{-j(\theta+\phi_i)} - s_k + \lambda_k e^{-j(\theta+\phi_k)} = 0. \tag{3}$$

The above equation is complex and the equality holds for both its real and imaginary parts. Separating the two gives $2(N-1)$ equations for $N+2$ unknowns. Rewriting this system of equations in matrix form yields

$$\mathbf{A}x = 0 \tag{4}$$

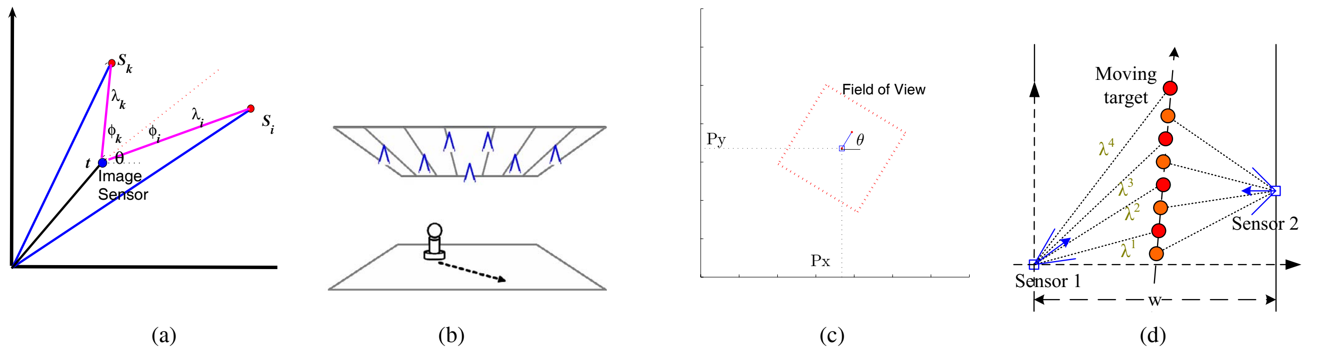


Fig. 2. Schematics of models used in the proposed schemes. (a) Vector relationship between the image sensor position and two locations of the beacon in method 1. (b) Simple diagram showing a mobile beacon assisting in localization of a network of image sensors in method 2. (c) The coordinates of an image sensor in a plane parallel to the ground (top view) used in method 2. (d) Observations made of a moving target with fixed velocity in method 3.

$$\mathbf{A} = \begin{bmatrix} \Re\{-e^{-j\phi_1}\} & \Re\{e^{-j\phi_2}\} & 0 & \cdots & 0 & 0 & \Re\{S_1 - S_2\} & -\Im\{S_1 - S_2\} \\ \Im\{-e^{-j\phi_1}\} & \Im\{-e^{-j\phi_2}\} & 0 & \cdots & 0 & 0 & \Im\{S_1 - S_2\} & \Re\{S_1 - S_2\} \\ \Re\{-e^{-j\phi_1}\} & 0 & \Re\{e^{-j\phi_3}\} & \cdots & 0 & 0 & \Re\{S_1 - S_3\} & -\Im\{S_1 - S_3\} \\ \Im\{-e^{-j\phi_1}\} & 0 & \Im\{-e^{-j\phi_3}\} & \cdots & 0 & 0 & \Im\{S_1 - S_3\} & \Re\{S_1 - S_3\} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \Re\{-e^{-j\phi_{N-1}}\} & \Re\{e^{-j\phi_N}\} & \Re\{S_{N-1} - S_N\} & -\Im\{S_{N-1} - S_N\} \\ 0 & 0 & 0 & \cdots & \Im\{-e^{-j\phi_{N-1}}\} & \Im\{-e^{-j\phi_N}\} & \Im\{S_{N-1} - S_N\} & \Re\{S_{N-1} - S_N\} \end{bmatrix} \quad (5)$$

where $x = [\ \lambda_1, \lambda_2, \ldots, \lambda_N, \Re\{e^{-j\theta}\}, \Im\{e^{-j\theta}\}\ ]^T$, and $\mathbf{A}$ is defined in (5). The last two elements in x are $\cos(\theta)$ and $\sin(\theta)$, respectively. Hence, we have an additional equation relating the unknowns. This means that the system is overdetermined for $N > 3$ observations. In the presence of noise, the null space of $\mathbf{A}$ may be empty meaning that no vector x satisfies $\mathbf{A}x = 0$, and we need to find the vector x that minimizes the value of $||\mathbf{A}x||$. One way to solve for x is to determine the singular vector of $\mathbf{A}$ corresponding to its smallest singular value. This yields a normalized vector v, which represents the solution to within a scaling factor. To find the true solution, we use the additional constraint between the last two elements of the vector x. Thus we find a scaling factor a such that

$$(av_{N+1})^2 + (av_{N+2})^2 = 1 \quad (6)$$

This yields $a = 1/\sqrt{v_{N+1}^2 + v_{N+2}^2}$. We then apply the sign constraints resulting from

$$\lambda_1, \lambda_2, \ldots, \lambda_N > 0 \quad (7)$$

to obtain the final estimate

$$\hat{x} = a'v \quad (8)$$

where $a' = \pm a$ depending on the common sign of the first N elements of the vector v. The solution is equivalent to a variation of the Total Least Squares (TLS) problem [7], [8].

From $\hat{x}$, the orientation of the image sensor is known, and the location of the node is given by substituting the estimated parameters into (2) and averaging over the observations:

$$\hat{t} = \frac{1}{N}\sum_{i=1}^{N}(s_i - \hat{\lambda}_i e^{-j(\hat{\theta}+\phi_i)}). \quad (9)$$

Fig. 3(a) presents simulation results showing the effect of errors in the beacon's coordinates on the location estimates for the nodes as a function of the number of beacon observations. Fig. 3(b) illustrates the actual results of an outdoor experiment.

IV. Method 2

We now present another beacon-assisted technique in which the image sensors are assumed to have image planes parallel to the plane in which the beacon travels. An example schematic of this case is shown in Fig. 2(b). Each image plane can have an unknown rotation angle θ as in the top view illustration in Fig. 2(c).

The beacon agent (see inset in Fig. 4(a)) travels through the network and at each stop broadcasts its location relative to an initial point of reference that defines a relative coordinate system for the network. In particular, the x and y axes and the unit length of a 2-D coordinate system can be defined by the beacon. The beacon may be observed by a few image sensors at each stop, each of which then detects the beacon's location on its image plane by simple frame subtraction using an initial background frame. Fig. 4(a) shows an example of the observed beacon positions by an image sensor.

More specifically, at time step i, the beacon broadcasts its coordinates denoted by a 2×1 vector s_i. These coordinates are mapped to the coordinates of the image plane of the sensor by

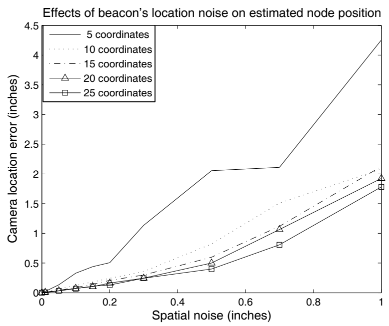
$$y_i = \alpha\mathbf{R}(s_i - p) + n_i,\ i = 1, \ldots, N \quad (10)$$

for N observations, where the vector y_i contains the observed coordinates on the image plane, the vector p contains the unknown coordinates of the sensor, the vector n_i is the noise modeled by a Gaussian random variable, α is an unknown scaling factor indicating the height of the sensor from the coordinates plane, and $\mathbf{R}$ is the rotation matrix with the unknown sensor orientation θ, i.e.

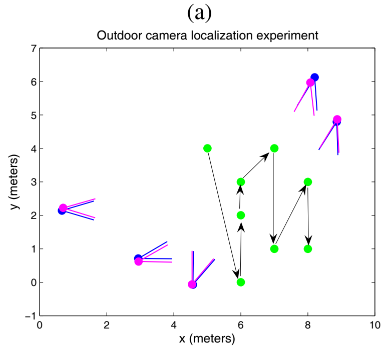
$$\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \quad (11)$$

To solve for the unknown variables α, θ, and p, we first rearrange (10) to cancel out the vector p using two observations:

$$y_{i+1} - y_i = \alpha\mathbf{R}(s_{i+1} - s_i) + (n_{i+1} - n_i) \quad (12)$$



(a)



(b)

Fig. 3. (a) Simulated effect of spatial noise on estimated node position for different number of observations. (b) Results of outdoor experiments for method 1.

for $i = 1, \ldots, N-1$. We define 2×1 vectors $\widetilde{s}_i$, $\widetilde{y}_i$, and $\widetilde{n}_i$ by $\widetilde{s}_i = s_{i+1} - s_i$, $\widetilde{y}_i = y_{i+1} - y_i$, and $\widetilde{n}_i = n_{i+1} - n_i$. Then, (12) can be rewritten in the matrix form as

$$\begin{bmatrix} \widetilde{y}_1(1) \\ \widetilde{y}_1(2) \\ \vdots \\ \widetilde{y}_{N-1}(1) \\ \widetilde{y}_{N-1}(2) \end{bmatrix} = \begin{bmatrix} \widetilde{s}_1(1) & \widetilde{s}_1(2) \\ \widetilde{s}_1(2) & -\widetilde{s}_1(1) \\ \vdots & \vdots \\ \widetilde{s}_{N-1}(1) & \widetilde{s}_{N-1}(2) \\ \widetilde{s}_{N-1}(2) & -\widetilde{s}_{N-1}(1) \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} + \widetilde{n} \quad (13)$$

where $\widetilde{n} = [\widetilde{n}_1(1), \widetilde{n}_1(2), \ldots, \widetilde{n}_{N-1}(1), \widetilde{n}_{N-1}(2)]^T$, and $v = \alpha \cos\theta$ and $w = \alpha \sin\theta$ are the unknown variables. We can solve for v and w in (13) by a standard least-square technique, which leads to a closed-form solution since only a 2×2 matrix inversion is needed in our model. After obtaining the estimates $\widehat{v}$ and $\widehat{w}$, we can easily find $\widehat{\alpha}$, $\widehat{\theta}$ and $\widehat{\mathbf{R}}$. The sensor coordinates p then can be derived by

$$\widehat{p} = \frac{1}{N} \sum_{i=1}^{N} (s_i - \frac{1}{\widehat{\alpha}} \widehat{\mathbf{R}}^{-1} y_i). \quad (14)$$

Knowing estimates for the image sensor coordinates, orientations, and heights relative to the coordinte system defined by the beacon, we can create a visual coverage map for the network. Fig. 4(b) shows a simulated network of 4 image sensors with random image plane orientations. The path and the locations that the beacon makes stops at to broadcast its position are also shown. Each image sensor needs to observe the beacon on at least two stop points to be able to solve for its own location parameters. The estimates for each image sensor's coordinates and orientation are superimposed on the actual positions, and each sensor's field-of-view is also drawn based on its estimated orientation and height. Fig. 4(c) shows the effect of image plane object detection error as pixel shift on the node location estimates.

V. Method 3

In this approach, we consider the use of a motion model for a moving target to localize the nodes. The technique is applicable to various target tracking applications, but we use the context of a roadway traffic monitoring application to derive the algorithm. The sensor nodes work in pairs to first estimate their relative orientations and positions by making observations of a moving target and exchanging information. This can be done based on the mere assumption that the target moves with a fixed (but unknown) velocity. For example, in the case of traffic monitoring, the movement of a vehicle moving with fixed speed along a highway lane can be used to localize the network of image sensor nodes. We will use the terms target and vehicle interchangeably in this section to refer to the moving object. Once the sensor pair is localized, it can estimate the target's observed positions and velocity by triangulation. The image sensors are assumed to be deployed as shown in Fig. 2(d). We can choose sensor 1 as the origin, and define a coordinate system as shown in the figure. When the sensors are deployed, we do not know their positions and orientations and the second sensor may not be on the x axis.

Given sampling time $\triangle t$, we can use three equations to describe the relationship of four observations at sensor 1:

$$\lambda_1^n e^{j\phi_1^n} e^{j\theta_1} + v\triangle t = \lambda_1^{n+1} e^{j\phi_1^{n+1}} e^{j\theta_1}, \quad n = 1, 2, 3 \quad (15)$$

where v is the target's speed, θ_1 is the orientation of the sensor with respect to the x axis, λ_1^n is the distance from the sensor to the object at the nth observation, and ϕ_1^n is the nth observed angle. By rearranging the equations, we can remove the unknown term $v\triangle t e^{-j\theta_1}$. Thus the unknown distances can be described by

$$\begin{bmatrix} \cos\phi_1^1 & -2\cos\phi_1^2 & \cos\phi_1^3 & 0 \\ 0 & \cos\phi_1^2 & -2\cos\phi_1^3 & \cos\phi_1^4 \\ \sin\phi_1^1 & -2\sin\phi_1^2 & \sin\phi_1^3 & 0 \\ 0 & \sin\phi_1^2 & -2\sin\phi_1^3 & \sin\phi_1^4 \end{bmatrix} \begin{bmatrix} \lambda_1^1 \\ \lambda_1^2 \\ \lambda_1^3 \\ \lambda_1^4 \end{bmatrix} = 0. \quad (16)$$

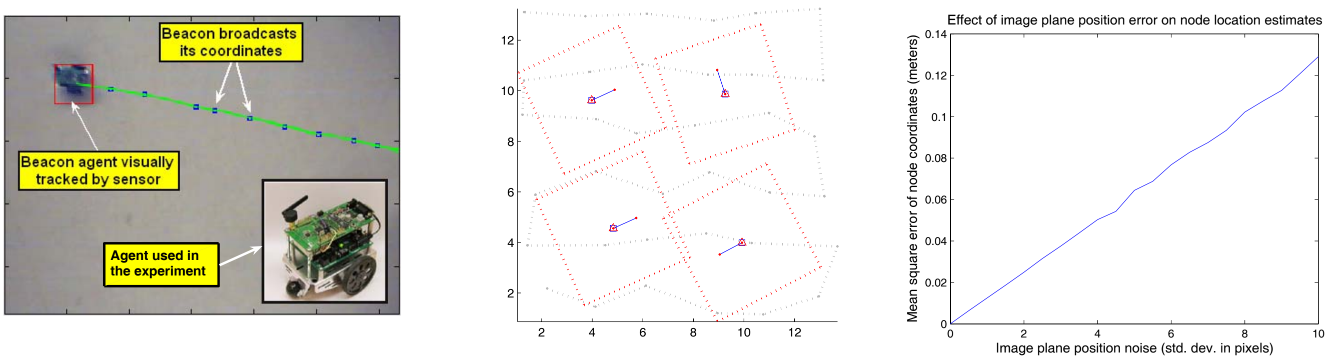
The vector of unknowns hence lies in the null space of the data matrix. We can obtain the values $\widetilde{\lambda}_1^n$ which are the normalized non-zero solutions of (16) as the elements of the singular vector corresponding to the smallest singular value of the data matrix. These values are the normalized values of $\widehat{\lambda}_1^n = c_1 \widetilde{\lambda}_1^n$, where $\widehat{\lambda}_1^n$ represents the estimate of λ_1^n and c_1 is the normalization factor, which is discussed later.

We now derive a relation for the sensor orientation. In the context of traffic monitoring, if we assume that the nodes observe several vehicles each of which generally (but not necessarily exactly) moving parallel to the roadway lane, we can model the movement direction $\measuredangle v$ as a random variable with a mean $\frac{\pi}{2}$, which indicates a velocity parallel to the roadway lane in our model. The orientation of each sensor can then be estimated by averaging the values obtained from observations made of different vehicles:

$$\widehat{\theta}_k = \text{average}\left[-\tan^{-1}\left(\frac{\widetilde{\lambda}_k^{n+1} \sin\phi_k^{n+1} - \widetilde{\lambda}_k^n \sin\phi_k^n}{\widetilde{\lambda}_k^{n+1} \cos\phi_k^{n+1} - \widetilde{\lambda}_k^n \cos\phi_k^n} \right) \right] \quad (17)$$

where k indicates the kth sensor.

Up to this point each sensor would perform its own calculations and can estimate its orientation angle with respect to the direction of the object's motion path. To find the distance between a pair of sensor nodes, the two nodes need to exchange information. In order to define a measure of length, we can use the distance w, which is the normal distance between the two lines passing through the sensor locations and parallel to the direction of the object's motion (see



(a) (b) (c)

Fig. 4. (a) The path taken by the beacon and the observation points within a node's field-of-view in method 2. (b) Simulated results of method 2 for a network of 4 image sensors. (c) Simulation results of average node position estimation error for image plane noise.

Fig. 2(d)). Using this measure of distance, we can find the factors $c_k, k = 1, 2$ that appear in the normalized solutions $\widehat{\lambda}_k^n = c_k \widetilde{\lambda}_k^n$ for the two sensors. Once we have these factors, we can derive the position of the target relative to sensors 1 and 2 as $p_1^n = \widehat{\lambda}_1^n e^{j\phi_1^n} e^{j\widehat{\theta}_1}$, and $p_2^n = \widehat{\lambda}_2^n e^{j\phi_2^n} e^{j\widehat{\theta}_2}, \quad n = 1, 2, 3, 4$.

Although p_1^n can be used to estimate the vehicle's velocity, four observations of the vehicle would be required. We can alternatively take two observations of the vehicle to estimate its velocity by triangulation if two sensors collaborate. If the two sensors are reasonably synchronized, they can take images simultaneously, and we will have $p_1^n = p_2^n + s_2$, where s_2 is the polar coordinates of sensor 2 and is unknown. With this information, the second sensor's position is derived by

$$\widehat{s}_2 = \text{average} \left[\widehat{\lambda}_1^n e^{j\phi_1^n} e^{j\widehat{\theta}_1} - \widehat{\lambda}_2^n e^{j\phi_2^n} e^{j\widehat{\theta}_2} \right], \tag{18}$$

in which averaging is performed over all the observations. Based on the above results, the vehicle's position at observation index n is given by

$$p^n = \frac{\sin\left(\phi_2^n + \widehat{\theta}_2\right) \mathrm{Re}\left(\widehat{s}_2\right) - \cos\left(\phi_2^n + \widehat{\theta}_2\right) \mathrm{Im}\left(\widehat{s}_2\right)}{\sin\left(\phi_2^n - \phi_1^n + \widehat{\theta}_2 - \widehat{\theta}_1\right)} e^{j\left(\phi_1^n + \widehat{\theta}_1\right)} \tag{19}$$

and the vehicle's velocity between the two observation instances $n-1$ and n is obtained by $\widehat{v}^n = (p^n - p^{n-1})/\triangle t$.

Fig. 5 shows the behavior of the estimation error in the node position as the number of observations increases. A demonstration video from a roadway traffic monitoring scenario that we developed based on this method can be downloaded from [9]. In this simulation, the sensors first estimate their orientations by observing the first few passing cars, and then estimate the velocities of new vehicles.

VI. Conclusion

This work addressed the localization problem in a novel way, in that visual observations made by image sensors from a beacon agent or a moving target were used to estimate network node coordinates. Mathematical formulation of the problem resulted in the desired parameter vectors to be in the null subspace of the data matrix or be obtained via simple least-squares methods. Application areas in which the image sensor node location and orientation information can be used include target detection and tracking, robot tracking and control, estimation of traffic speed in roadways, and implementing geographic routing schemes for wireless sensor networks.

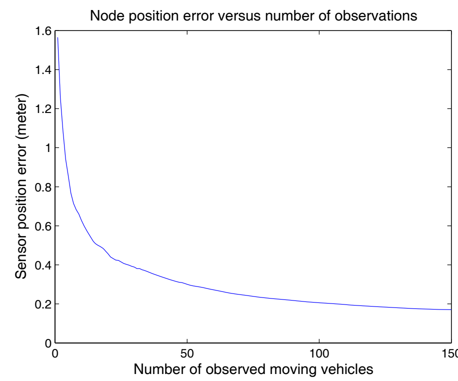


Fig. 5. Simulation results for method 3 show that the average position estimation error decreases when more moving objects are observed.

VII. Acknowledgments

The authors would like to credit Agilent Technologies and Chipcon for providing hardware samples and support.

References

[1] J. C. Chen, K. Yao, and R. E. Hudson, "Source localization and beamforming," *IEEE Signal Proc. Mag.*, vol. 19, no. 2, pp. 30–39, Mar. 2002.

[2] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. of the ACM MobiCom'04*, Sept. 2004.

[3] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed online localization in sensor networks using a moving target," in *in Proc. of the Third International Symposium on Information Processing in Sensor Networks*, Apr. 2004, pp. 61–70.

[4] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in distributed adhoc wireless sensor networks," in *Proc. ICASSP*, 2001, pp. 2037–2040.

[5] W. Mantzel, H. Choi, and R. G. Baraniuk, "Distributed alternating localization-estimation of camera networks," in *Proceedings of the 38th Asilomar Conference on Signals, Systems, and Computers*, Nov. 2004.

[6] D. Agathangelou, B. P. Lo, J. L. Wang, and G.-Z. Yang, "Self-configuring video-sensor networks," in *Proceedings of the 3rd International Conference on Pervasive Computing (PERVASIVE 2005)*, May 2005.

[7] S. V. Huffel and J. Vandewalle, *The Total Least Square Problem: Computational Aspects and Analysis*. Frontiers in Appl. Math., SIAM, 1991.

[8] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[9] [Online]. Available: Roadway Traffic Monitoring Simulated Demo, Wireless Sensor Networks Lab, Stanford University, http://wsnl.stanford.edu/demos/trafficsim.avi.