

Sensor Node Localization Using Mobile Acoustic Beacons

Manish Kushwaha, Károly Molnár, János Sallai, Péter Völgyesi, Miklós Maróti, Ákos Lédeczi
Institute for Software Integrated Systems, Vanderbilt University
{manish.kushwaha, janos.sallai, akos.ledeczi}@vanderbilt.edu

Abstract— We present a mobile acoustic beacon based sensor node localization method. Our technique is passive in that the sensor nodes themselves do not need to generate an acoustic signal for ranging. This saves cost, power and provides stealthy operation. Furthermore, the beacon can generate much more acoustic energy than a severely resource constrained sensor node, thereby significantly increasing the range. The acoustic ranging method uses a linear frequency modulated signal that can be accurately detected by matched filtering. This provides longer range and higher accuracy than the current state-of-the-art. The localization algorithm was especially designed to work in such acoustically reverberant environment, as urban terrain. The algorithm presented handles non-Gaussian ranging errors caused by echoes. Node locations are computed centrally by solving a global non-linear optimization problem in an iterative and incremental fashion.

I. INTRODUCTION

Localization is an essential tool for the deployment of low-cost sensor networks for use in location-aware applications [12], [13], [25] and ubiquitous networking [4], [26]. In a typical sensor network application each sensor node monitors and gathers local information. This local information has much more significance if it can be tied to the physical location it belongs to. In location-critical applications, such as shooter-localization [24], sub-meter accuracy of 3D node locations is an absolute necessity for the correct operation of the system.

Range-free localization techniques provide rough estimates of node positions only. Ranging methods fall into two main classes: acoustic and radio signal strength-based. The latter requires extensive calibration, yet it still achieves low accuracy and limited range. Acoustic ranging has relatively high accuracy, but short range. The main reasons are the limited acoustic energy a sensor node can emit and the possibly high environmental noise. Having a speaker or sounder on every node adds size and cost also. When stealthy operation is required, only ultrasound can be used. But ultrasonic ranging has even more limited range and directionality constraints.

A sensor network deployment scenario with many favorable characteristics in numerous application areas is the dispersal of sensor nodes from a low-flying unmanned aerial vehicle (UAV) platform. After deployment, an acoustic beacon mounted on the aircraft can send a radio message followed by an acoustic signal at random intervals. All the nearby sensor nodes can estimate their distance from the beacon by measuring the time-of-flight of the sound. As size and power are not as big constraints on a UAV as on a sensor node, the maximum range

can be significantly increased. Furthermore, the nodes do not reveal their positions since they are only passive listeners in this scenario.

The self-localization problem in this case is to find the sensor node locations given only the distance measurements between unknown mobile beacon transmission locations and the sensor nodes. Neither the mobile beacon positions nor the sensor nodes themselves are located necessarily on a plane. Therefore, the localization problem needs to be solved in 3D. Furthermore, to our knowledge, no solutions exist in the literature that handle multipath effects satisfactorily. For urban deployments both of these problems need to be addressed.

The main contributions of our work are (1) the acoustic ranging method providing increased range and accuracy, (2) the localization algorithm based on the novel idea of a mobile acoustic beacon and (3) the ability to handle multipath effects. The ranging method is based on the time-of-flight measurement of an acoustic signals emitted by a single beacon from multiple locations. The acoustic signal used is a linear frequency modulated (chirp) signal, that can be identified with high accuracy by matched filtering at the sensors even at low SNR. Self localization is modeled as a non-linear optimization problem where node locations are the optimization variable and distance equations involving node locations are non-linear objective functions. The localization algorithm is both iterative and incremental. At each iteration a part of the sensor network is selected, localized and evaluated. It is incremental because at each iteration the part of sensor network selected will grow around the previously localized nodes. This method is a generalization of iterative localization algorithms where node location is improved in each iteration.

The rest of the paper is organized as follows. Section II summarizes related research in self localization. Section III presents the novel acoustic ranging technique. Section IV formulates the self localization problem. The main algorithm is presented in section V while its implementation, results and conclusions are provided in sections VI and VII.

II. RELATED RESEARCH

Self localization, due to its importance in sensor network applications, has been an active research area for the past few years. An early survey of some localization systems is presented by Hightower and Boriello in [6]. Many of these systems adopt a simple connectivity based approach, while some of them further refine range estimates between node pairs

by measuring the received radio signal strength. However, RSS based ranging requires extensive calibration and still yields inaccurate range estimates [7] resulting in coarse localization.

The GPS-less system by Bulushu [1] employs a grid of reference nodes with overlapping regions. Unknown nodes localize themselves to the centroid of their proximate reference nodes. Localization accuracy is about one third of separation distance between reference nodes. Doherty [3] formulated self localization as a geometric constraint feasibility problem based on node connectivity that was solved using convex optimization. Additionally, rectangular bounds on node locations were used for tighter geometric constraints.

Other techniques that provide much better range estimates involve time-of-flight measurements, particularly when acoustic and RF signals are combined [2], [5], [20], [21], [22]. Acoustic signals, however, require an unobstructed line-of-sight. In an urban environment echoes present a significant problem, thus any localization algorithm has to consider multi-path propagation.

Savvides [22] solves for unknown node position estimates by setting up a global non-linear optimization problem and solving it using iterative least-squares. The method requires the known beacons to surround the unknown nodes, which the author calls beacon-unknown node convexity. However, this topology constraint is hard to satisfy in real world deployment scenarios.

Savarese [20] follows a two phase localization algorithm: start-up and refinement. The start-up phase utilizes hop-TERRAIN algorithm which is similar to DV-hop [15]. The refinement phase is an iterative algorithm that uses the location estimates from start-up phase. [20] also introduces a crude notion of confidence value, a metric for the quality of location estimate.

There are few approaches, that deal with multi-path propagation. One such approach for two dimensions is presented by Moore [14]. It identifies echoes as geometric impossibilities. The idea can be extended to three dimensions but under low connectivity or high measurement noise conditions the algorithm may be unable to localize a useful number of nodes [14]. Another case where the geometric constraint based echo identification may fail is when the distributions of nodes in the three dimensions are different. In a typical sensor network the X and Y distribution of nodes is much higher than that in Z which affects the performance of the algorithm above.

Recently some work has been done in localization using mobile beacons. Sichitiu [23] uses a mobile beacon that is aware of its location using GPS. Priyantha [17] describes mobile-assisted localization where mobile beacon movement and node localization is interlaced.

The presented localization algorithm models the problem as global non-linear optimization as in [22], however it goes one step further to deal with echoes and non-convexity of anchor-unknown node topology.

III. RANGING

The concept of acoustic ranging is based on measuring the time-of-flight of the sound signal between the source (beacon) and the acoustic sensor. The range estimate can be trivially calculated from the time measurement. However, the speed of sound is temperature dependent. This problem can be resolved by a single temperature measurement at the base station. An appealing characteristic of the proposed ranging algorithm is that this is the only calibration that is needed. That is the sensors do not need individual calibration at all.

A. Hardware

The acoustic ranging application targets the MICA2 motes developed at UC Berkeley [8]. The mote is equipped with a custom acoustic sensor board, which was developed at Vanderbilt University for a shooter localization application [24]. The heart of the sensor board is the low-power fixed point ADSP-2189 digital signal processor running at 50 MHz. The availability of the DSP enables the implementation of sophisticated digital signal processing algorithms.

There are two independent analog input channels on the board, furnished with low-cost electret microphones and 2-stage amplifiers with software programmable gain (0-54 dB). The analog channels are sampled by A/D converters at up to 100 kSPS with 12-bit resolution. The board also has an analog output channel capable of driving a 250 mW external loud-speaker. The board is connected to the mote by programmable interrupt and acknowledgment lines and a standard I2C bus.

In the current implementation the mobile beacon is based on a MICA2 mote and the same sensor board with an active loudspeaker attached to its analog output channel. The maximum output power is 105 dB measured 10 cm away from the loudspeaker.

B. Ranging algorithm

In order to calculate the range from the time-of-flight of the acoustic signal, the departure and arrival times of the signal have to be identified and measured precisely. The beginning of the transmission can be measured at the beacon, while the time of arrival is measured at the receiving sensors. The range calculation is performed on the receivers, thus the beacon has to send the starting time to the receivers in a radio message.

Employing a sophisticated time synchronization mechanism is essential to accurately measure the time-of-flight. Our approach employs the message time-stamping primitives introduced in [11]. The synchronization between the source and the sensor nodes is implemented as follows.

The source queries its local time t_0 and decides that it will emit an acoustic signal at time $t_{send} = t_0 + \delta$. The source sends the value t_{send} to all the sensors in a radio message. Therefore, the value of δ is chosen such that it is greater than the time required by the sensors to process the radio message and to prepare for receiving. The sensors schedule their acoustic board for sampling when the beacon starts the transmission of the acoustic signal.

We assume that the skew of the local clocks is negligible during the short time of the measurement, but we allow arbitrary clock offsets. Since neither the source nor the sensors have knowledge of a global time, the sensors need to convert t_{send} included in the message from the local time of the source to their own local times. This is achieved by timestamping the radio message at transmission and at reception as well. The timestamping of the radio message is done in the MAC layer just before transmission and just after reception respectively. Since the radio signal is traveling at the speed of light, the difference between the transmit time instant and the receive time instant is negligible, hence the transmit timestamp (given by the local clock of the beacon) and the receive timestamps (in the local time of the receivers) are assumed to represent the same global time instance. Thus, a sensor can use the difference of the transmit timestamp and its receive timestamp to calculate the offset of its local clock from the local clock of the beacon. This offset is added to the received t_{send} to convert it to the local time of the receiver.

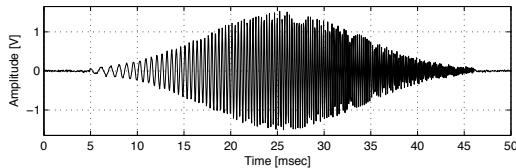


Fig. 1. The emitted acoustic signal

The sensor node also has to measure the time of arrival of the acoustic signal. The accurate detection of the signal is not trivial in a noisy environment, as it is difficult to emit sharp rising edges or pulses with general purpose loudspeakers. Additionally, the signal has to be emitted with the highest power available in order to maximize the range of the measurement. These requirements are analogous to the problems of radar signals, a well researched area [10], [18]. The problem arises as the limited bandwidth of the analog output channel restricts the emission of rising edges with arbitrarily steep slope. The contradiction is resolved by long duration signals with short duration correlation functions, so when the received signal goes through an appropriate matched filter, the output will be a sharp pulse. The emitted signal is therefore a Gaussian-windowed linear frequency modulated (chirp) signal shown in Fig.1, that is commonly used in radar applications. The windowing is needed due to the limited bandwidth of the acoustic channel.

A similar solution is presented in [5], where the emitted signal is a binary phase shift keying (BPSK) spread spectrum signal. Since our method does not require to distinguish multiple sources, the use of linear frequency modulated signal is more natural.

The frequency span of this signal is spread in the whole acoustic band of the analog channels. The matched filter is realized as an FIR filter on the DSP. The matched filtering essentially means the correlation of the expected signature with the measured data, therefore the length of the FIR

filter is the same as the length of the expected signature. To avoid a high order FIR filter which would be computationally expensive, either the length of the chirp signal or the sampling rate has to be decreased. However, as the length of the chirp signal can not be arbitrarily short due to the limited bandwidth of the physical hardware, the sample rate has to be decreased. Thus, the raw data is decimated to a lower sampling frequency before the matched filtering.

In order to increase the signal-to-noise ratio (SNR), one range measurement consists of a series of time-of-arrival measurements. As the delays between the consecutive chirps are known a-priori, an accurate combined result can be calculated by averaging these measurements. In the averaged signal the chirp signature component is preserved as it is added up at the same phase, but the noise which is assumed to be independent Gaussian white noise is decreased by \sqrt{N} where N is the number of chirps added. Currently we use 8 chirps, thus the SNR of the averaged signal is 9 dB higher than the SNR of a single chirp.

Delays between consecutive chirps are varied to avoid a situation when multiple runs have the same noise pattern at the same offset, which is a common phenomenon caused by acoustic multipath effects. Hence the independent nature of the disturbances is preserved.

The decimation filtering runs online on the DSP, and the decimated signal is stored in a RAM buffer. The consecutive measurements are added together in the same buffer. After all the chirps are received, the matched filtering and the peak-detection algorithm is performed offline. The peak-detection algorithm is simply a maximum finder above a threshold level, as the output of the matched filter has distinctive peaks at chirps. The time of arrival of the chirp signal can easily be identified based on the location of the peak.

C. Results

The above algorithm was tested on a grassy field with a single beacon and multiple receivers. In Fig. 2 the ranging results are presented, and in Fig. 3 the standard deviation of the measurements is shown, after outlier rejection. Outlier

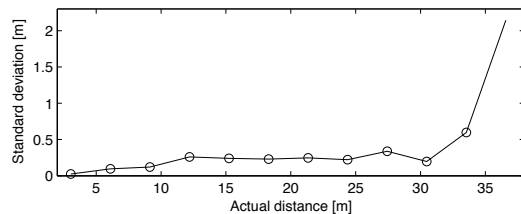
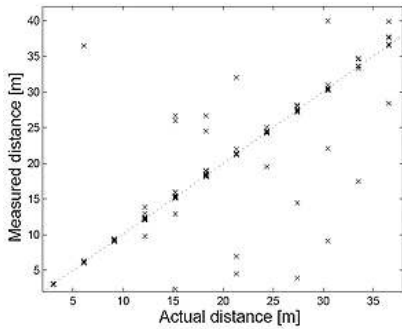


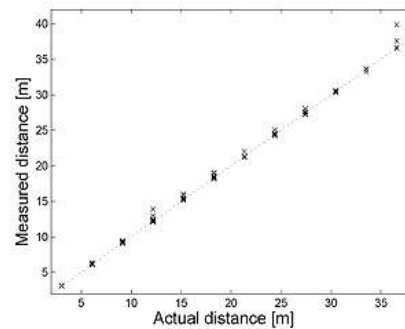
Fig. 3. Standard deviation of ranging

rejection is done by a simple median filter, where the values greatly differing from the median of the measurements are rejected. Note that since it is statistical filtering, multiple measurements are needed for each beacon position to perform the rejection algorithm.

The effective range of the presented implementation is 30 meters, as the number of outliers and the standard deviation



(a)



(b)

Fig. 2. Ranging measurement results (a) without outlier rejection, (b) with outlier rejection

of the measurements are getting significantly high above this value. Below 30 meters the standard deviation grows approximately linearly with

$$STD \cong k_1 d + k_2 \quad (1)$$

where $k_1 = 0.011$ and $k_2 = 0.024$ and d denotes the actual distance.

The effective range of the measurements are more than two times larger than in previous acoustic ranging experiments [19], [9], where the reliable range was 10 m on asphalt and 15 m on grass, respectively. The standard deviation is also significantly improved. In [19], the output power of the sounder was limited (88 dB at 10 cm from source) and the no custom DSP board was used. In [9] the power of the beacon is approximately the same as in the presented solution (105 dB at 10 cm from source), however our use of the DSP board and the linear frequency modulated signal provides better performance.

These experimental results are very promising and justify the presented approach. Moreover, the current limits on range and precision are primarily caused by issues with the current implementation. First, the power of the emitted acoustic signal is still constrained by the gain on the output channel of the board. Second, the analog input channels of the DSP board also limit the range, as they were designed for a shooter detection application [24], where even the maximum gain is relatively low.

IV. SELF LOCALIZATION

Formally, a generalized self localization problem can be defined as follows. Given node IDs and their ranges from each other conjecture the relative physical location of each node in the network. A few anchor nodes can be provided to transform relative positions to absolute locations. There are many challenges to be addressed in this problem. First let us define some terminology.

DISTANCE MATRIX D is a matrix such that d_{ij} is the range measurement between node i and node j . Distance is negative for node pairs for which range measurement is not known. Number of positive

entries in row i represents the number of neighbors of node i .

NECESSARY CONDITION FOR LOCALIZATION in 3-dimensions states that a node should have distance measurements with at least four non-coplanar neighbor nodes.

In a typical urban environment many sensor nodes might not have line-of-sight with mobile beacon positions, but they can receive the acoustic signal via multipath. These multipath ranges or echoes, when used for localization, produce false or infeasible results. The amount of echoes present in range measurements heavily depends on the environment and the maximum range of the applied ranging method. In typical urban environments, low network connectivity and non-uniform node distribution in the Z-direction further deteriorate the localization accuracy, that is even more critical at boundary nodes.

A. Self Localization As Distance Optimization

The self localization problem in its most basic form can be modeled as a distance optimization problem. Here the independent optimization variables are node locations and the non-linear objective functions are the differences between distances computed from node locations and range measurements for all node pairs for which range measurements exist (Equation 2). It can be observed that the distance optimization is actually a function-fitting problem where distances are the non-linear functions of node locations. Least square optimization is known to work best for function-fitting problems [16]. The mathematical formulation of distance optimization problem is presented below.

Find \mathbf{x}^* , a global minimizer for

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^{N, \hat{d}_{ij} \geq 0} (d_{ij} - \hat{d}_{ij})^2 \quad (2)$$

where $d_{ij} = \{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2\}^{1/2}$ is the computed distance between nodes i and j , and \hat{d}_{ij} is the measured distance. $\mathbf{x} = [x_1 y_1 z_1 \dots x_n y_n z_n]^T$ is the optimization variable where $[x_i y_i z_i]$ is the 3D coordinate of node i . The non-linear objective function $F(\mathbf{x})$ is the

square sum of distance errors for all pairs (i, j) for which range measurements exist ($\hat{d}_{ij} \geq 0$). The components of the optimization variable \mathbf{x} are subjected to the boundary value constraints.

$$\begin{aligned} x_{min} &\leq x_i \leq x_{max} \\ y_{min} &\leq y_i \leq y_{max} \\ z_{min} &\leq z_i \leq z_{max} \end{aligned} \quad (3)$$

V. SELF LOCALIZATION ALGORITHM

An obvious and straightforward algorithm would be to solve for all unknown node locations simultaneously (Algorithm 1).

Algorithm 1 Self localization algorithm

- 1: Consider 3D coordinates of *all* unknown nodes in optimization variable.
 - 2: Construct and solve non-linear least-square optimization problem with objective function in eqn. (2).
-

This approach has some serious disadvantages. Convergence of the optimization problem strongly depends upon the initial guess given to the solver. A close-to-optimum initial guess would converge to global optimum relatively fast, while a bad initial guess for the same problem might lead to a local optimum. Initial estimates for node locations can be computed by using an extension of the bounding box technique described in [22]. But due to the large size of the sensor network and relatively few randomly distributed anchor nodes, it is possible that we do not have good initial estimates for the whole network, but only for the part close to the anchors.

An iterative incremental approach wherein a part of the network near anchor nodes is localized first and then the node locations are propagated further seems suitable. The idea is to iteratively select and localize a part of the network (a sub-system) for which a good initial estimate is available. At each iteration the part of the network selected for localization will grow, consisting of nodes that are already localized and few unknown neighboring nodes that have better estimates in the current iteration. In each iteration ranges that are believed to be echoes are identified and removed from computation. The algorithm is presented below (Algorithm 2). Symbol \mathbf{x} represents the 3D location vector of nodes, \mathbf{x}^{est} and \mathbf{x}^{sol} denote estimated and localized node location vectors respectively. N denotes the set of nodes in the network and η denotes the confidence value for the localization (an estimate of the accuracy of the current location described in section V-C).

There are two levels of looping in the algorithm. The outer loop starts with an estimate, \mathbf{x}^{est} for the whole network. The first run of the outer loop starts with a random (or user given) estimate. Each run afterwards starts with the final estimate of the previous run. The inner loop corresponds to the incremental selection and localization of a sub-system \tilde{N} , that we will call an iteration. At each iteration, the selected sub-system will increase in size, more nodes will be localized with

Algorithm 2 Incremental iterative self localization algorithm

- 1: $\mathbf{x}^{\text{est}} \leftarrow \mathbf{0}$, $\mathbf{x}^{\text{sol}} \leftarrow \mathbf{0}$
 - 2: **for** $run = 1$ to run_{max} **do**
 - 3: Configure parameters, read distance matrix D , set sub-system $\tilde{N} \leftarrow \emptyset$
 - 4: **repeat**
 - 5: $\tilde{N}_{old} \leftarrow \tilde{N}$
 - 6: Estimate bounding-box $B_i \forall i \in N$
 - 7: Choose $x_i^{\text{est}} \leftarrow x \in B_i \forall i \in N - \tilde{N}_{old}$ based on neighbor polling
 - 8: Select $\tilde{N} \subseteq N$ such that x_i^{est} satisfies goodness $\forall i \in \tilde{N}$
 - 9: Optimize \mathbf{x} for sub-system \tilde{N}
 - 10: $\mathbf{x}^{\text{est}} \leftarrow \mathbf{x}$
 - 11: **for all** $i \in \tilde{N}$ **do**
 - 12: Compute η_i
 - 13: $\tilde{N}_{sol} \leftarrow \emptyset$
 - 14: **if** η_i acceptable **then**
 - 15: $\mathbf{x}_i^{\text{sol}} \leftarrow \mathbf{x}_i$
 - 16: $\tilde{N}_{sol} \leftarrow \tilde{N}_{sol} \cup \{i\}$
 - 17: **end if**
 - 18: **end for**
 - 19: **until** $\tilde{N}_{sol} - \tilde{N}_{old} = \emptyset$
 - 20: **end for**
 - 21: Output \mathbf{x}^{sol}
-

higher accuracy until there are no more nodes to be localized or no more nodes can be localized (i.e. the necessary condition for localization does not hold). Later sections describe each step of the algorithm in detail.

A. Sub-System Selection

Each node is represented by a bounded-box with lower and upper bounds $(\mathbf{x}_{lb}, \mathbf{x}_{ub})$. The node coordinates can take any value in the closed interval $[\mathbf{x}_{lb} \mathbf{x}_{ub}]$. Since anchor nodes are known with high accuracy, their bounding-box is very small. Initially, the bounding-boxes for all unknown nodes can be set to the size of the field and can be updated using range measurements \hat{d}_{ij} between node i and its neighbors j .

$$\mathbf{x}_{lb,i} = \min_j \{(\mathbf{x}_{lb,j} - \hat{d}_{ij}), \mathbf{x}_{lb,i}\} \quad (4)$$

$$\mathbf{x}_{ub,i} = \min_j \{(\mathbf{x}_{ub,j} + \hat{d}_{ij}), \mathbf{x}_{ub,i}\} \quad (5)$$

The order in which bounding-box update should be done is also important. Considering the sensor network as a graph it turns out that a variant of the topological sort (Algorithm 3) will provide the required node ordering.

For node i that already has an estimate $\mathbf{x}_i^{\text{est}}$ and confidence value η_i , the bounds are reset as follows. Confidence values for node location estimates are computed in the sub-system evaluation section and described later.

$$\mathbf{x}_{lb,i} = \max\{(\mathbf{x}_i^{\text{est}} - \eta_i), \mathbf{x}_{lb,i}\} \quad (6)$$

$$\mathbf{x}_{ub,i} = \min\{(\mathbf{x}_i^{\text{est}} + \eta_i), \mathbf{x}_{ub,i}\} \quad (7)$$

Algorithm 3 Topological sort

- 1: Set known neighbor index, $\kappa = \infty$ for anchors and $\kappa = 0$ for all other vertices
 - 2: **while** Graph not empty **do**
 - 3: Find a vertex u with highest $\kappa[u]$
 - 4: Output u
 - 5: Delete all edges $e = (u, v)$ of u , increment $\kappa[v]$ by 1
 - 6: Delete u from graph
 - 7: **end while**
-

For all other nodes a location estimate is picked from the bounding-box. The most obvious way would be to pick the center of the box, but a heuristic method involving bounding-box partitioning is used instead. The bounding-box of a node, if larger than some critical size, is partitioned into smaller boxes and neighbors are polled for the partition in which the node is most likely to be present. The center of the winning partition is assumed to be the estimated location for that node. A polling index C_p is computed for each partition p , which is essentially a weighted sum of distance errors for all neighbors j of node i .

$$C_p = \sum_{j \in \text{Neigh}(i)} \left| \|\mathbf{x}_p - \mathbf{x}_j^{\text{est}}\| - \hat{d}_{ij} \right| \cdot \eta_j \quad (8)$$

where \mathbf{x}_p is the center point of partition p . The center point of the partition with minimum polling index is chosen as the estimated location for that node.

A part of the network is selected based the following notion of *goodness* of estimated node locations. An estimated location for node i is considered *good* if the node has at least *three* neighbors and its bounding-box satisfies two properties. First, its volume V_i is smaller than some critical volume \mathbf{V} and second, its aspect ratio α_i is greater than some critical $\bar{\alpha}_{\text{adaptive}}$. Aspect ratio α_i is a measure of *cubeness* of the bounding-box. α_i is expressed in terms of bounding-box volume V_i , space diagonal d_i and surface area A_i ,

$$\alpha_i = \frac{6\sqrt{3} \cdot V_i}{A_i \cdot d_i} \quad (9)$$

Notice that for a node with a small bounding-box an estimate is acceptable even if it has a smaller aspect ratio. For this reason the critical aspect ratio is made adaptive, quadratically depending on the bounding-box volume.

B. Sub-System Localization

The distance optimization problem for the selected sub-system is solved in multiple stages. At each stage the solution is moved closer to the optimum. First, let us define an operator *min* and two optimization problem formulations.

Operator min:

DEFINITION 1. Let f_i be a list of N function evaluations (or numbers), then $\min_p f_i$ is the list of

$\lceil pN \rceil$ -many smallest function evaluations (or numbers) where $\lceil \cdot \rceil$ is ceiling operator and $0 \leq p \leq 1$.

DEFINITION 2. Let $\sum_i^N f_i$ be a series sum of N function evaluations (or numbers), then $\sum_i^N \min_p f_i$ is the series sum of $\lceil pN \rceil$ -many smallest function evaluations where $\lceil \cdot \rceil$ is ceiling operator and $0 \leq p \leq 1$.

1) *Pruned Distance Optimization Problem.*: As mentioned in section IV we have non-Gaussian error as echoes in range measurements. In least-square optimization terminology, these echo ranges are outliers that tend to shift the least-square model from the actual model. It is desirable not to consider these outliers in optimization. The outlier rejection in section III-C is statistical and requires multiple ranging measurements. The outlier rejection in this section identifies and removes consistent echoes.

Find \mathbf{x}^* , a global minimizer for

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^{N, \hat{d}_{ij} \geq 0} \min_p \left(d_{ij} - \hat{d}_{ij} \right)^2 \quad (10)$$

where \hat{d}_{ij} and d_{ij} are the range measurement and distance computed from localized nodes i and j respectively and optimization variable $\mathbf{x} = [x_1 y_1 z_1 \dots x_n y_n z_n]^T$.

If the optimizer \mathbf{x} is close to global optimizer \mathbf{x}^* then all function evaluations but those corresponding to echoes will be close to zero. We can say that near the global optimizer large function evaluations correspond to echoes. Least-square optimization works best if the errors have Gaussian distribution. When we discard the top few function evaluations using the *min* operator, we are discarding the most significant outliers in the distribution and hence obtaining an approximate Gaussian distribution.

2) *Distance Penalty Optimization Problem.*: The optimization solver used in this work are for unconstrained optimization. The bounded-value constraints on the optimization variables are incorporated by modeling them as penalty functions in the objective function. Penalty functions incorporate a penalty value if variables go out of bound.

The most intuitive form of a penalty function is a rectangular penalty wherein a constant high penalty is incorporated if the variable goes out of bounds. For optimization purposes rectangular penalty does not provide motivation (descent direction) for the variable to fall within bounds. Another forms of penalty functions are linear or quadratic growing linearly or quadratically with the offset from the bounds. Logarithmic penalty functions are most suitable for bounded-value constraints because of their sudden descent near boundary values.

Find \mathbf{x}^* , a global minimizer for

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^N \{ \kappa \cdot \ln(1 + \Delta x_{\text{off},i}) \}^2 \quad (11)$$

where κ is penalty constant and $\Delta x_{\text{off},i}$ is the offset from

feasible boundary,

$$\Delta x_{\text{off},i} = \begin{cases} |x_i - x_{\min}| & \text{if } x_i < x_{\min} \\ 0 & \text{if } x_{\min} \leq x_i \leq x_{\max} \\ |x_i - x_{\max}| & \text{if } x_i > x_{\max} \end{cases} \quad (12)$$

and optimization variable $\mathbf{x} = [x_1 y_1 z_1 \dots x_n y_n z_n]^T$.

3) *Composition Of Least-Square Optimization Problems.*: Two or more least-square optimization problems can be composed as follows. Consider two least-square optimization problems P_1 and P_2 on optimization variable \mathbf{x} and objective functions $\sum_i^N f_i(\mathbf{x})$ and $\sum_j^M g_j(\mathbf{x})$, then the combined least-square optimization problem P on variable \mathbf{x} have the objection function

$$F_P(\mathbf{x}) = \sum_i^N f_i(\mathbf{x}) + \sum_j^M g_j(\mathbf{x}) \quad (13)$$

Now we describe the stages of optimization. We solve problem V-B.1 or the combination of problems V-B.1 and V-B.2 at each stage. The solution from the previous stage is used as a starting point for the current stage. At the end of each stage some range measurements that are believed to have non-Gaussian errors (echoes) are identified and removed from the distance matrix.

- STAGE I. At this stage echo ranges are identified and discarded based on the evaluation of the objective function in Equation 10 at the current optimizer \mathbf{x}^{est} .
- STAGE II. At this stage the optimization problem V-B.1 is optimized in a fixed number of iterations. The solver is stopped even if the optimizer has not converged. Lets visualize this stage as a 3D earth terrain optimization problem where x and y directions are optimization variables and altitude from sea-level, i.e. z , is the optimization function. The global optimization in this problem is looking for the deepest trench on earth. Optimizing for fixed number of iterations can be visualized as going downwards a local trench but not going all the way down because that may take unbounded time.
- STAGE III. At the previous stage we did not consider bounded-value constraints on the optimization variable. The variable might go out of the feasible region as guided by the objective function. In this stage the combination of the optimization problems V-B.1 and V-B.2 are optimized in a fixed number of iterations. The objective function in Equation 11 ensures that the variable will fall within the feasible region. The reason for having stage II separate from stage III is that sometimes the path to the global optimizer goes through a region that might not be part of the feasible region.
- STAGE IV. This final stage is similar to stage III except parameter p in Equation 10 is set to 1.0, i.e. no pruning of the distance matrix is done. It is expected that by the end of stage III we would have discarded most significant echo measurements.

C. Sub-System Evaluation

The quality of computed locations produced by the solver is evaluated using a measure called confidence value. Confidence value is an indicator of uncertainty in node location around the current location estimate.

The algorithm to compute the confidence value is following. Compute the ranges between node locations and the deviation of these computed ranges from measured ranges. Now for each node i we have a deviation vector Δ_i whose elements are the deviations of computed ranges from measured ranges for all its neighbors. A large value in Δ_i indicate that either (1) the node location is incorrect or (2) the corresponding range measurement is incorrect. If the node location is incorrect then most of the elements of Δ_i should be large. If only a few range measurements are incorrect then the mean and the variance of Δ_i should be small except for those incorrect range measurements. Practically, all node locations are categorized based on mean μ_i and standard deviation σ_i in Δ_i . Confidence value η_i is equal to $|\mu_i| + \sigma_i$.

- 1) If both μ_i and σ_i are close to zero then the node location is correct.
- 2) If μ_i is close to zero but σ_i is large then either the range deviations are spread around zero or few large deviations caused σ_i to be large. We say that the node location may be affected by echo. In this case we strike out a few large deviations and re-categorize the location based on a recomputed mean and standard deviation.
- 3) If $|\mu_i|$ is large but σ_i is small then all elements of Δ_i are large i.e. the node location is definitely incorrect.
- 4) If both $|\mu_i|$ and σ_i are large then again location might be affected by echo and we follow the same procedure as in case 2 above.
- 5) If $|\mu_i|$ and σ_i are neither large nor small then location correctness is undecided. We follow the same procedure here as in cases 2 and 4.

Node locations categorized as incorrect or potentially echo-affected are considered not localized.

VI. IMPLEMENTATION AND RESULTS

We have implemented the proposed localization algorithm in MATLAB and ran it on simulated sensor network topologies and ranging data. The Levenberg-Marquardt solver was used for optimization.

A topology of 50 sensor node locations was generated randomly in a $100 \times 100 \times 20$ m field with at least half of the nodes on ground level. 80 sound sources were generated on random paths such that the separation between successive sound sources was bounded (0–8 m). Also, the Z variation of the sources was limited to 2 m to simulate a mobile beacon, which is moving on the ground in the sensor field. Ranging data was generated with 30 m maximum range. Gaussian noise with zero mean and range dependent standard deviation (Equation 1 in section III) was added to the ranging data. This matches the results from our ranging experiments. Echoes were also introduced to ranging data based on our previous

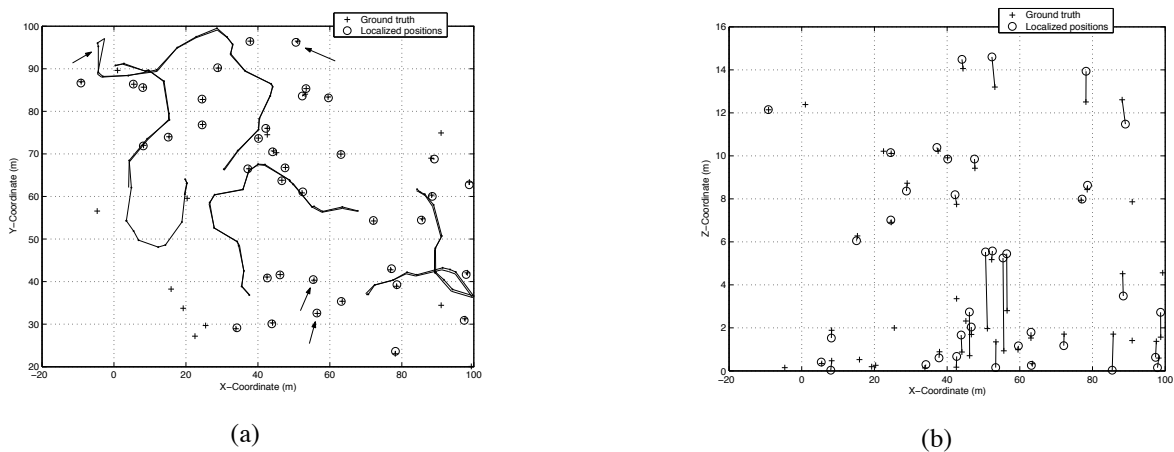


Fig. 4. Comparison of computed node locations to their true values in (a) XY and (b) XZ plane for ranging data w/ echoes. ‘+’s without a ‘o’ indicate unlocalized nodes.

ranging measurements in urban environments. Approximately 1000 range estimates were gathered using the 80 beacon positions. 10% of these had added non Gaussian error (echo). Five sensor nodes were assumed to be known anchor locations. Two different ranging data sets, one with echoes and another one without echoes, were generated for the topology.

In the presence of ground truth, the performance of the algorithm can be evaluated by the localization error which is the difference between computed locations and the ground truth. Localization error for node i is,

$$\sigma_{p,i}^2 = (x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2 + (z_i - \tilde{z}_i)^2 \quad (14)$$

where x_i , y_i and z_i are the computed coordinates of node i , and \tilde{x}_i , \tilde{y}_i and \tilde{z}_i are the true location coordinates of the same node.

Figure 4 compares the computed node locations to their true values in XY and XZ views for ranging data with echoes. Solid lines show the paths of the mobile beacon. Solid arrows in Figure 4(a) indicate the sensor nodes that has the highest localization errors. Notice that all such nodes are very far from their nearest sound source.

Figures 5(a) and 5(b) show the histograms of localization error with and without simulated echoes. Table I summarizes the localization results.

	Ranges echoes	w/o	Ranges echoes	w/
Unlocalized sensors	7		9	
Mean error [m]	0.8962		1.0664	
Max error [m]	4.3252		4.5119	

TABLE I
LOCALIZATION RESULTS

Notice that the distribution of localization error is very steep in case (a) while its more flat in case (b). More nodes were localized with better accuracy when we did not have echoes in ranging data as expected.

From Figure 5 we can see that the computed locations of sound sources are more accurate than that for sensor nodes.

This high accuracy can be attributed to the topological fact that sensor nodes are uniformly distributed *around* the sound sources. For node localization application we are actually not concerned about the computed beacon locations. However, it is an important observation that if we distribute the sound sources uniformly around sensor nodes, then we can get higher localization accuracy for the sensors.

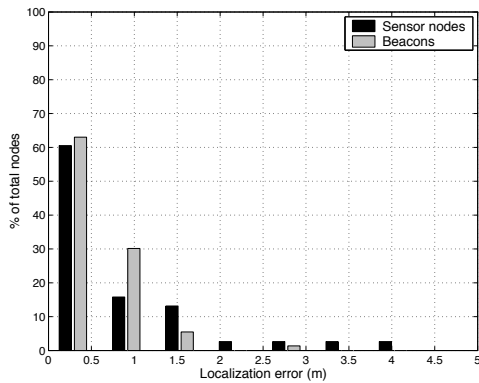
VII. CONCLUSIONS

The presented sensor node localization technique has several contributions. The method is passive since only the mobile beacon needs to emit acoustic signals. This saves energy, size and cost on the sensor nodes and provides stealthy operation. Furthermore, the mobile beacon can emit much higher-energy sound than the sensor nodes thereby increasing the effective range. To the best of our knowledge our acoustic ranging method has the longest range even when normalized by the emitted sound energy. This is due to the signal processing algorithms implemented on the sensor board.

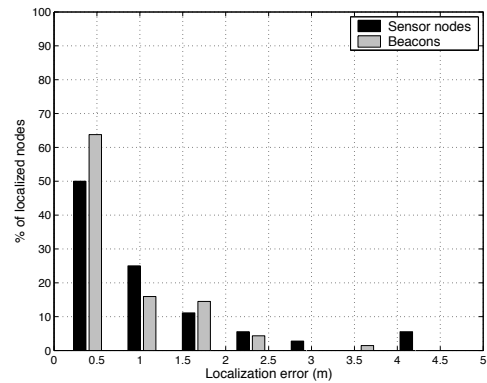
The iterative and incremental non-linear optimization technique provides an effective way to deal with acoustic multipath effects and works well for 3D localization. There is little work in the wireless sensor network literature that addresses these problems.

We put special emphasis on making our experiments/simulation as realistic as possible. Our setup strongly resemble a feasible real world deployment. Node density was relatively low. The technique needed to deal with both echoes and 3D locations. There were a relatively low number of beacon positions. Beacon positions varied very little in the Z dimension. We had only a few anchor nodes. Therefore, we believe that the results are realistic.

Approximately half of the nodes were localized with sub-meter accuracy. That is very good when compared to the current state-of-the-art, but unfortunately still not good enough for such location-critical applications as shooter localization. However, many other application domains have much less strict requirements. Finally, to put the results into perspective,



(a)



(b)

Fig. 5. Histograms of localization error for all sensor nodes and sound sources without (a) and with (b) echoes in ranging data.

(non-differential) GPS-based localization would have much less accuracy than these results.

VIII. ACKNOWLEDGMENT

The DARPA IXO NEST program has supported the research described in this paper.

REFERENCES

- [1] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low cost outdoor localization for very small devices. Technical report, Computer Science Department, University of Southern California, 2000.
- [2] S. Capkun, M. Hamdi, and J.-P. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proc. of HICSS*, 2001.
- [3] L. Doherty, K. S. J. Pister, and L. E. Ghaoui. Convex optimization methods for sensor node position estimation. In *Proc. of INFOCOM*, 2001.
- [4] D. Estrin, R. Govindan, and J. Heidemann. Embedding the internet: introduction. *Commun. ACM*, 43(5), 2000.
- [5] L. Girod and D. Estrin. Robust range estimation using acoustic and multimodal sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.
- [6] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*.
- [7] J. Hightower, R. Want, and G. Borriello. SpotON: An indoor 3D location sensing technology based on RF signal strength. Technical report, University of Washington, Department of Computer Science and Engineering.
- [8] J. Hill. and D. Culler. MICA: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6), 2002.
- [9] Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha. Resilient localization for sensor networks in outdoor environments. Technical Report UIUCDCS-R-2004-2449, Dept. of Computer Science, UIUC, 2004.
- [10] N. Levanon and E. Mozeson. *Radar Signals*. Hoboken, NJ John Wiley & Sons, Inc., 2004.
- [11] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. The flooding time synchronization protocol. In *SenSys 04*, 2004.
- [12] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks, 2001.
- [13] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in wireless ad-hoc sensor networks. In *MOBICOM*, 2001.
- [14] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *SenSys '04*, 2004.
- [15] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In *Proc. of INFOCOM*, 2001.
- [16] J. Nodedal and S. J. Wright. *Numerical Optimization*, pages 250–275. Springer-Verlag New York, Inc., 1999.
- [17] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *IEEE InfoCom 2005*, 2005.

- [18] L. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing*, pages 715–716. Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [19] J. Sallai, G. Balogh, M. Maroti, and A. Ledeczi. Acoustic ranging in resource-constrained sensor networks. In *International Conference on Wireless Networks, Las Vegas, Nevada*, 2004.
- [20] C. Savarese, J. Rabay, and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX Technical Annual Conference*, 2002.
- [21] A. Savvides, C.-C. Han, and M. B. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. *Mobile Computing and Networking*, 2001.
- [22] A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *WSNA*, 2002.
- [23] M. L. Sichitiu and V. Ramadurai. Localization of wireless sensor networks with a mobile beacon. In *MASS 2004*, 2004.
- [24] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *SenSys 04*, 2004.
- [25] M. B. Srivastava, R. R. Muntz, and M. Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments. In *MOBICOM*, 2001.
- [26] S. J. Teller, J. Chen, and H. Balakrishnan. IEEE computer graphics & applications: Projects in VR pervasive pose-aware applications and infrastructure. *IEEE Distributed Systems Online*, 4(8), 2003.