SENSOR NODE LOCALIZATION USING MOBILE ACOUSTIC BEACONS

By

Manish Kushwaha

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Computer Science

August, 2005

Nashville, Tennessee

Approved:

Professor Gabor Karsai

Professor Akos Ledeczi

ACKNOWLEDGEMENT

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA), under the Networked Embedded Software Technology (NEST) program. I would like to thank my graduate advisor, Dr. Gabor Karsai, for helping me focus on my studies and research. I am also very grateful to my research advisor, Dr. Akos Ledeczi, who motivated and guided me through my work. I would also like to thank Dr. Miklos Maroti for the insightful ideas he put forward that made this work possible. I am further grateful for the cooperation of Dr. Gyula Simon, Janos Sallai, Branslav Kusy, Gyorgy Balogh, Karoly Molnar and other members of the NEST project team at ISIS. Their help, criticism and advice saw me through a number of rough spots.

I would like to thank my parents, who always believed in me and always encouraged and supported me. Thanks to my sister, Sonal, for the faith she has in me. Thanks to my little brother, Dhruv, who thinks of me as the smartest person. Finally, thanks to the human curiosity which constantly drives us to do the unthinkable, to reach the places where no human has gone before.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

The revolutionary new technology of wireless sensor networks has opened a wide area of exciting and powerful applications that will connect the cyber-world more intimately with the real world. A wireless sensor network consists of a large number of small-scale, resource-constrained computing nodes, outfitted with sensors and linked together by radios that form a perceptive network that is able to monitor an ecosystem or detect a specific phenomenon [6]. Although these smart sensors have limited power, communication and processing capabilities, an assembly of hundreds of them can spontaneously organize into an ad hoc perceptive network that is spread throughout the physical world and is able to perform tasks no ordinary computer system could.

Wireless sensor networks have a middleware system whose main purpose is to support the deployment, execution and maintenance of sensing applications. Römer [20] notes that the scope and functionality of a middleware system for WSN includes, but is not limited to, "mechanisms for formulating complex high-level sensing tasks, communicating this task to the WSN, coordination of sensor nodes to split the task and distribute it to the individual sensor nodes, data fusion for merging the sensor readings of the individual sensor nodes into a high-level result, and reporting the result back to the task issuer. Moreover, appropriate abstractions and mechanisms for dealing with the heterogeneity of sensor nodes should be provided. All mechanisms provided by a middleware system should accommodate special characteristics of WSN, which mostly boils down to energy efficiency, robustness, and scalability".

Node Localization

This thesis describes a middleware service called node localization and presents a new method for node localization. Node localization, also known as self-localization, is the problem of localizing physical sensor nodes in a given sensor network deployment. Localization is an essential tool for the deployment of low-cost sensor networks for use in location-aware applications [27, 33, 24] and ubiquitous networking [7, 19]. In a typical sensor network application each sensor node monitors and gathers local information. This local information has significance if it can be tied to the physical location it belongs to. For example in a habitat monitoring application a temperature or humidity measurement by a smart sensor doesn't provide the complete information but once combined with the physical location of the sensor, the measurements can be used to build a temperature or humidity map of the local region. In location-critical applications, such as shooter-localization, sub-meter accuracy of 3D node locations is an absolute necessity for the correct operation of the system [12].

Localization can be either range-free [34] or range-based. Range-free localization techniques provide rough estimates of node positions only. Ranging methods for range-based localization 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 relatively 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.

Sensor Network Deployment

There are many sensor network deployment scenarios suggested for practical and useful application. Simple but impractical deployment schemes include manually placing the nodes on specified locations or random deployment of nodes in a rectangular grid. We propose a new practical 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. 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 by increasing the size and power of the sound source. Furthermore, the nodes do not reveal their positions since they are only passive listeners in this scenario.

In general the node localization problem can be defined as finding physical locations of sensor nodes in a given network deployment scenario. This work redefines the self-localization problem for our network deployment scheme as finding 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. In complex environments sensor nodes might not have direct line-of-sight with mobile beacon but they receive the signal via multipath propagation. These multipaths cause ambiguity in ranging data and produce false results. To our knowledge, no solutions exist in the literature that handles multipath effects in 3D environment. For urban deployments this problem needs to be addressed as well.

The main contribution of this work is the localization algorithm based on the novel idea of a mobile beacon and its ability to handle multipath effects. The ranging method is based on the time-of-flight measurement of an acoustic signal 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 variables 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 at each iteration.

The document is organized as follows. Chapter I provides motivation, brief introduction and challenges in node localization. Chapter II introduces two interesting sensor network applications that require node localization. Related research work in self localization is summarized in chapter III. Chapter IV presents the novel acoustic ranging method used in this work. Formal problem definitions and mathematical tool definitions are presented in chapter V. In Chapter VI we present the main localization algorithm in detail. The implementation and evaluation of localization technique is provided in chapter VII. Chapter VIII concludes this document with a discussion on the significance of deployment strategy and localization algorithm presented in this thesis.

CHAPTER II

APPLICATIONS

Current and potential application areas for wireless sensor networks include habitat monitoring, geophysical monitoring, precision agriculture, military systems, surveillance systems, traffic monitoring, business processes management, smart building management, and in the future, possibly smart homes for everybody.

Two very interesting applications are described in this chapter. The first one is that of habitat monitoring of Great Duck Island off the coast of Maine designed by University of California at Berkeley [1], and the second is a countersniper system designed and developed at Vanderbilt University [12].

Habitat Monitoring

In fragile habitats, such as a small island, human intervention for observation and data gathering can adversely affect the small ecosystem. Sensor networks provide non-invasive methods of observation of such fragile ecosystems. Sensor networks also enable life science researchers to observe and monitor habitats that are not easily accessible to humans.

Sensor nodes with weather sensor boards were deployed in the observation area and their location were recorded. The weather sensor boards had temperature, light, barometric pressure, humidity and passive infrared sensors. The data recorded by sensors would be transmitted to the base-station. Base-stations would transfer the data over network to data storage or client data processing units. In this way sensor networks allow clients to remotely monitor the habitat. Figure II.1 [1] shows the system architecture for habitat monitoring.

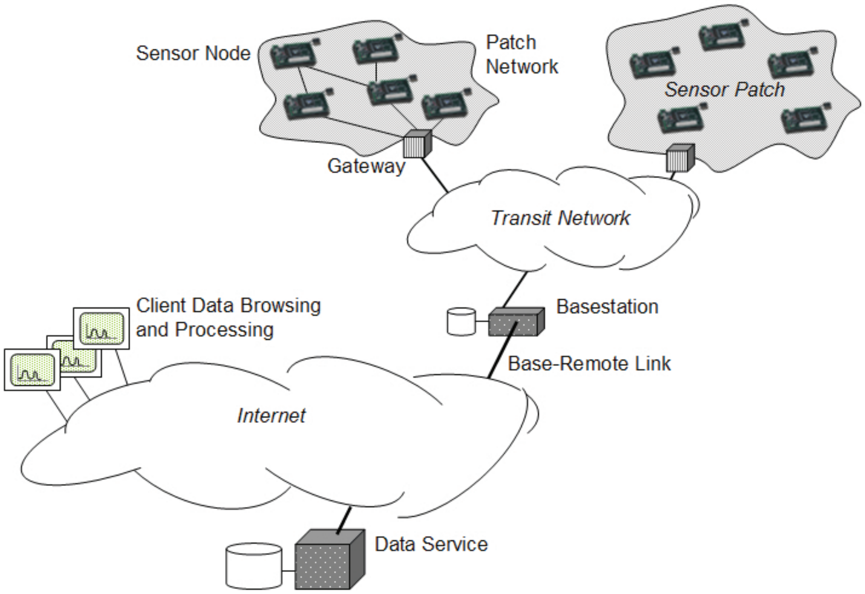


Figure II.1: System architecture for habitat monitoring

Shooter Localization

Countersniper systems are important tools for armed forces and law enforcement agencies. Many systems have been developed in the past but only a few can meet the requirements in complex environment as urban terrain. Simon [12] notes that "the main problems degrading the performance of these systems are the poor coverage due to the shading effect of the buildings and the presence of multipath effects".

Several physical phenomenon associated with shot firing can be used for shooter localization. The most obvious acoustic event due to firing of a conventional weapon is the muzzle blast. The countersniper system in [12] utilizes this muzzle blast and the shockwave generated by the supersonic bullet to localize the shooter. Figure II.2 [12] shows the simplified geometry of the bullet trajectory and the associated muzzle blast and shockwave fronts. The muzzle blast produces a spherical wave front, traveling at the speed of sound (v_s) from the muzzle (A) to the sensor (S). The shock wave is generated in every point of the trajectory of the supersonic projectile producing a cone-shaped wave front, assuming the speed of the projectile is constant

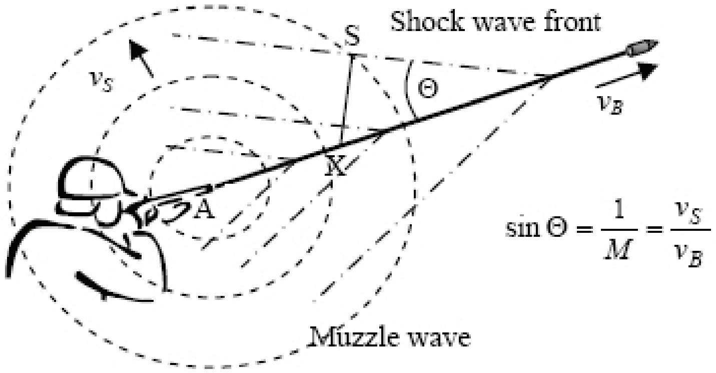


Figure II.2: Muzzle blast and shock wave

v_B. The shockwave reaching sensor S was generated in point X. The angle of the shockwave cone is determined by the speed of the projectile. The sensor nodes detect and measure the time of arrival (TOA) of shockwaves and muzzle blasts. The system then utilizes the measurements to localize the source of muzzle blast.

It is crucial to have accurate sensor node locations for the correct and accurate operation of shooter localization application described above. The shooter-localization application localizes the shooter based on the TOA of muzzle blast and shockwaves *and* the locations of the sensor nodes that detect muzzle blast. The errors in sensor node locations are propagated to the shooter localization errors. A more detailed illustration of effect of node location errors on shooter location error is given in chapter V.

CHAPTER III

EXISTING LOCALIZATION TECHNOLOGIES

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 localization systems in presented by Hightower and Boriello in [14]. Many of these systems adopt a simple connectivity based ranging 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 [22] resulting in coarse localization. A number of recent localization systems use acoustic time of flight for ranging. The calibration required for acoustic ranging is minimal. The localization system presented in this work uses acoustic ranging.

One of the earliest and most popular localization systems is GPS or Global Positioning System [13]. GPS was designed by and is controlled by the United States Department of Defense. GPS is a satellite navigation system used for global outdoor localization and providing highly accurate global time reference, which is useful, for example, in military operations. The project, started in 1978, uses a multi-million dollar infrastructure of 52 GPS satellites, the latest of which was launched in 2004. Despite the popularity and easy accessibility of GPS receivers they are not suitable for localization in wireless sensor networks. The reasons for this are the cost factor, power consumption and the localization accuracy. Even the best GPS receivers do not claim more than 2-3 meter resolution and can have up to 10-20 meter localization error. This accuracy is acceptable for applications like vehicle navigation or landmark localization but it is not sufficient for many sensor network applications.

Radio Connectivity and Hop Based Ranging

Many localization algorithms are based on simple circular radio connectivity model [8]. In this approach the radio transmitter is sitting at the center of the circular region and all nodes within this region are considered connected to the transmitter. This model is very simple and doesn't require any special ranging infrastructure. The network routing infrastructure can be used for radio connectivity based ranging. The localization systems that are based on this approach are [28, 10]. There are several problems with ranging based on radio connectivity. Asymmetries in the environment or in the antenna's orientation can affect radio connectivity. Furthermore, the radio ranges show probabilistic behavior i.e. two radio nodes are connected with some probability. This *link quality* is illustrated in Figure III.1 [8]. Apart from the problem of being *probably connected links* there are two more issues with this approach. Even if two radio nodes are connected to a transmitter, the model is insufficient to tell

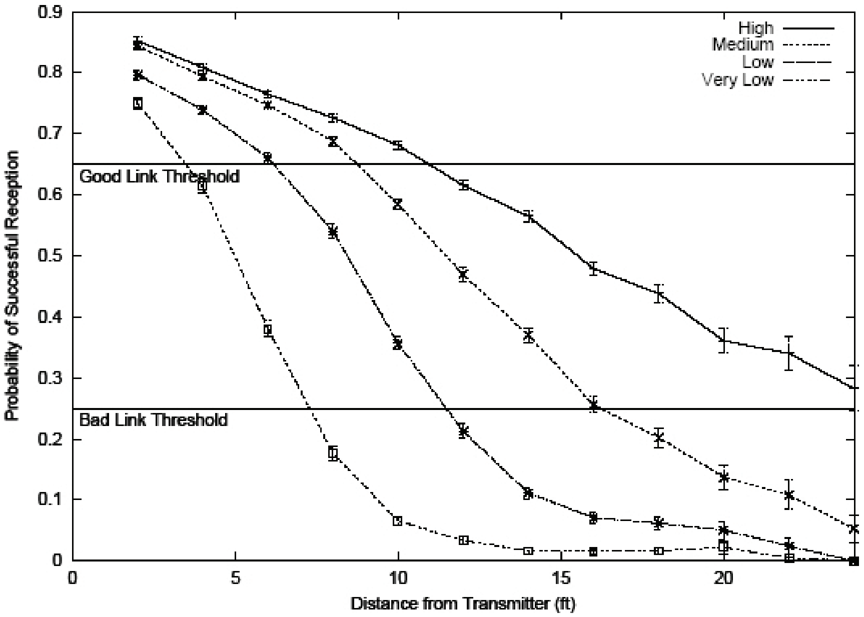


Figure III.1: Radio connectivity over distance

which node is closer. Finally, many pairs whose connectivity falls near the boundary will oscillate between being connected and disconnected.

A simple connectivity-based localization system by Bulushu [28] employs a grid of reference nodes with overlapping regions. Unknown nodes localize themselves to the centroid of their proximate reference nodes. The accuracy of localization is then dependent on the separation distance between two adjacent reference nodes and the transmission range of these reference nodes. Experiments show localization accuracy of about one third of separation distance between reference nodes.

Niculescu and Nath [10] proposed a distributed hop by hop localization algorithm called APS (Ad hoc Positioning System). The idea here was to propagate the anchor node location information in the network. Nodes with unknown location note the shortest hop count to each of the anchor nodes and multiply this with an average hop distance to get an approximate distance to each of the anchor nodes. Nodes can then perform triangulation to get an estimate of their locations. They describe three different propagation methods, *DV-Hop, DV-distance* and *Euclidean* propagation methods, each providing different tradeoff between accuracy, signaling complexity, coverage and isotropy of the network. Typical location errors were of the order of $20 - 150\%$ of radio range.

Doherty [21] formulated self localization as a geometric constraint feasibility problem based on node connectivity. The problem was formulated as, given anchor node locations find a possible position for each unknown node subjected to the proximity constraints between node pairs imposed by known connections. The problem was solved using convex optimization. Additionally, rectangular bounds on node positions were used for tighter geometric constraints. Doherty noted that provided tight enough geometric constraints, simulations show that the node estimates become close to actual node locations. For sensor network of 200 nodes in a $10{\times}10$ R region, where

R was the maximum range. The simulations show that the mean location errors were $1 - 9\ R$ for different anchor nodes ratios.

Radio Signal Strength Based Ranging

Several localization systems use received radio signal strength to estimate ranges between transmitter and receiver. Recent research projects in ad-hoc localization using signal strength are SpotON [15] and AHLoS [4] among others. For a symmetric transmitting antenna in a near-ideal environment signal strength at a receiver with distance r from a transmitter can be described by RADAR equation,

$$P_r = \frac{P_t G_t A_r}{4\pi R^2} \tag{III.1}$$

where P_r is the received power, P_t is the transmit power, G_t is antenna gain, A_r is the effective receiver area and R is the distance between transmitter and receiver. Since receiver area A_r is constant, signal strength is effectively inversely proportional to square of the distance. This simple radio signal strength model is unreliable in complex indoor or urban environments due to obstacles and reflections. Figure III.2 [11] shows the observed mean and standard deviation of signal strength with distance in a *near-ideal* environment i.e. outdoor field.

Savarese [5] follows two phase localization algorithm,, containing a start-up and a refinement phase. The start-up phase utilizes the hop-TERRAIN algorithm which is similar to DV-hop [10]. The hop-TERRAIN algorithm finds the number of hops from a node to each anchor nodes and then multiplies this hop count by an average hop distance to estimate the range from anchors. The nodes with distances from anchors and known anchor location triangulate their estimated position. The second stage called the refinement is an iterative algorithmic step that uses the ranges and the location estimates from hop-TERRAIN phase. Refinement updates the node positions

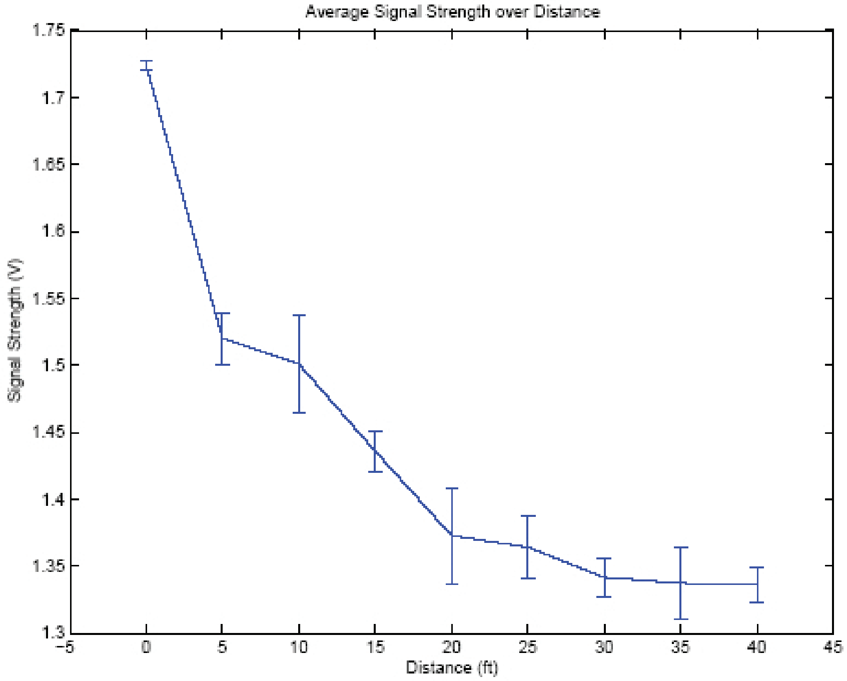


Figure III.2: Radio signal strength

in a number of steps. Savarese [5] also introduces a crude notion of confidence value, a metric for the quality of location estimate.

Savvides [3] solves for unknown node position estimates by setting up a global nonlinear 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 with airborne deployment of sensor nodes in hostile urban areas.

Acoustic Time of Flight Based Ranging

The ranging techniques based on time of flight (TOF) measurements of signals provide better results that are sufficient for fine-grained localization [18]. RF time of flight based techniques, like GPS, have limited applicability in sensor networks. Much better ranging results can be achieved when acoustic and RF signals are combined [22, 2]. Recently, several localization methods have been proposed that utilize distance estimates using time-of-flight measurements [32, 2, 30]. Acoustic signals, however, are temperature dependent and require an unobstructed line of sight. In a typical indoor or urban environment non-line of sight range measurements, or multi-paths produce false ranges that are difficult to separate from good ranges. Also, the ratio of multi-paths to line-of-sight range measurements is considerable, thus any algorithm localizing nodes in such scenarios has to consider multi-path propagation.

Cricket [30, 31] is an indoor location support system for pervasive and sensor-based computing applications. Cricket is intended for use indoors or in urban areas where outdoor systems like the Global Positioning System (GPS) don't work well. Cricket uses a combination of RF and ultrasound to provide a location-support service to users and applications. Wall and ceiling-mounted beacons are spread through the building, publishing location information on an RF signal. With each RF advertisement, the beacon transmits a concurrent ultrasonic pulse. The listeners receive these RF and ultrasonic signals, correlate them to each other, estimate distances to the different beacons using the difference in RF and ultrasonic signal propagation times, and therefore infer the space they are currently in.

We use acoustic time of flight based ranging developed at Vanderbilt University [18] in our localization system. It is described in detail in section IV.

There are few approaches that deal with multi-path propagation. One such approach for two dimensions is presented by Moore [9]. It identifies multi-paths as geometric impossibilities. Moore et. al. formulated node localization as a two-dimensional graph realization problem. They identified that due to insufficient or noisy data there could be ambiguity or uncertainty in node positions. The two types of ambiguities in a graph that prevent unique realization are flip and flex ambiguities as shown in Figure III.3[9]. As defined in [9], "*Flip ambiguities* (Figure III.3a) occur for a graph in a d-dimensional space when the positions of all neighbors of some vertex span a $(d-1)$-dimensional subspace. In this case, the neighbors create a mirror

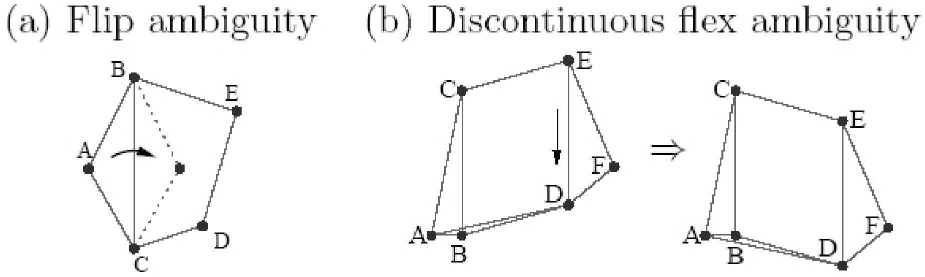


Figure III.3: (a) Flip ambiguity. Vertex A can be reflected across the line connecting B and C with no change in the distance constraints. (b) Discontinuous flex ambiguity. If edge AD is removed, then reinserted, the graph can flex in the direction of the arrow, taking on a different configuration but exactly preserving all distance constraints.

through which the vertex can be reflected. *Discontinuous flex ambiguities* (Figure III.3b) occur when the removal of one edge will allow part of the graph to be flexed to a different configuration and the removed edge reinserted with the same length".

They introduce the concept of robust quadrilaterals to overcome these ambiguities. In terms of graph theory a robust quadrilateral is a set of four fully-connected nodes that are unique up to a global rotation, translation and reflection, i.e. it is globally rigid. The idea can be extended to three dimensions but as [9] noted, under low connectivity or high measurement noise conditions the algorithm may be unable to localize a useful number of nodes. Another case where the geometric constraint based echo identification fails is when there is disproportionate geometry thickness in one direction. In a typical sensor network the X and Y distribution of nodes is much higher than that in Z which can affect the performance of the algorithm.

The localization algorithm presented in this thesis models the problem as global non-linear optimization problem as in [3], however it goes one step further to deal with echoes and non-convexity of anchor-unknown node topology.

CHAPTER IV

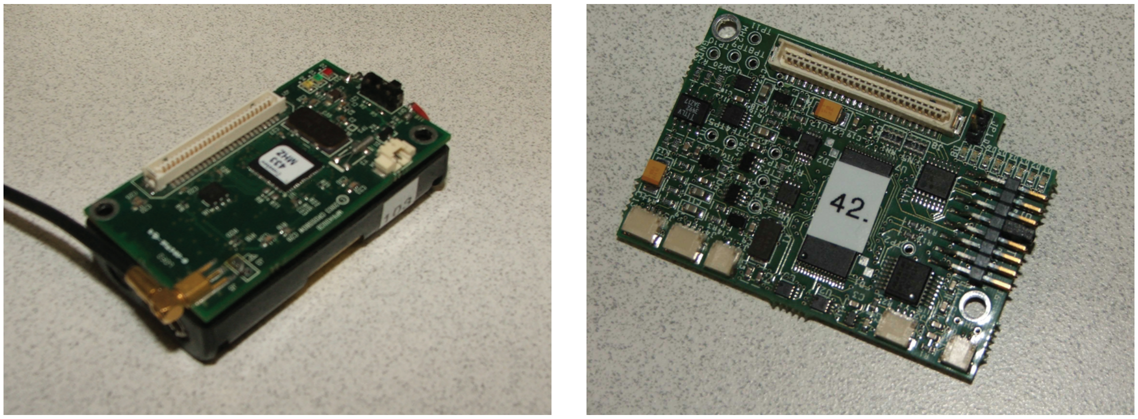
APPLIED ACOUSTIC RANGING

The localization system described in this work uses acoustic time-of-flight based ranging developed at Vanderbilt University [18]. The concept of acoustic ranging is based on measuring the time of flight of the sound signal between the signal 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. Assuming insignificant spatial temperature distribution in the sensor field, this problem can be solved 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.

Hardware

The acoustic ranging application targets the MICA2 motes developed at UC Berkeley [16]. The mote is equipped with a custom acoustic sensor board, which was developed at the Vanderbilt University for a shooter localization application [12]. The heart of the sensor board is a low-power fixed point ADSP-2189 digital signal processor running at 50 MHz (see Figure IV.1). 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 (0-54dB). The analog channels are sampled by A/D converters at up to 100kSPS with 12-bit resolution. The board also has an analog output channel capable of driving a



(a) (b)

Figure IV.1: Ranging hardware (a) MICA2 motes, (b) acoustic sensor board

250 mW external loudspeaker. 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.

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 [26]. The synchronization between the source and 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 reception. 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. 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.

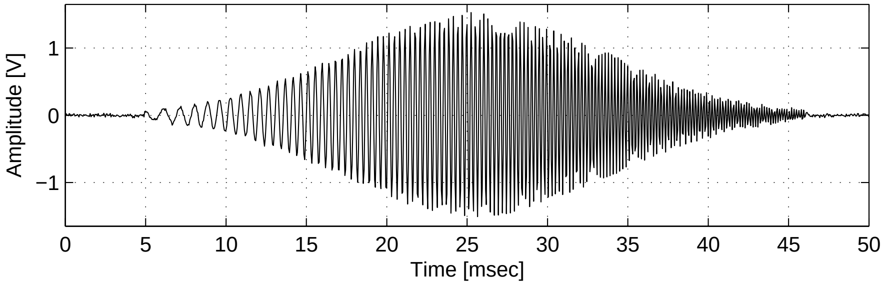
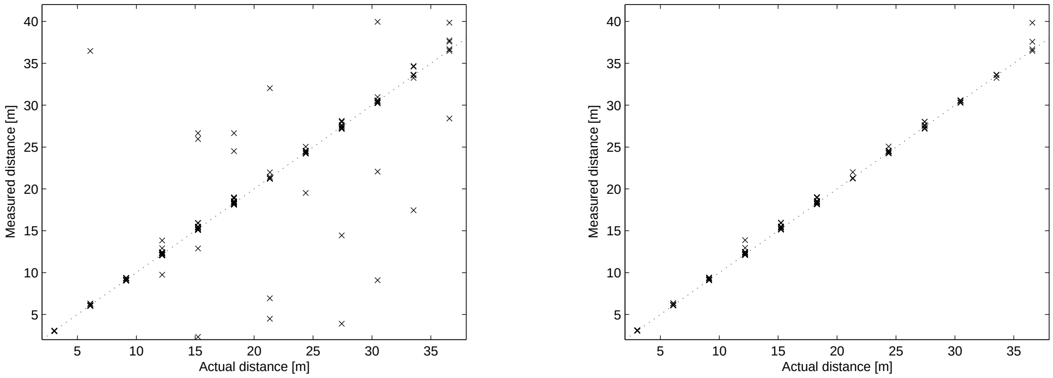


Figure IV.2: The emitted acoustic signal



(a) (b)

Figure IV.3: Ranging measurement results (a) without outlier rejection, (b) with outlier rejection

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 [29, 23]. 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.IV.2, 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 [22], 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 multi-path effects. Hence the independent nature of the disturbances is preserved.

The decimation filtering is running online on the DSP, and the decimated signal is stored in RAM buffer. The consecutive measurements are added together in the same buffer and 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.

Results

The above algorithm was tested on a grassy field with a single beacon, and multiple receivers. In Fig. IV.3 the ranging results are presented, and in Fig. IV.4 the standard deviation of the measurements is shown, after outlier rejection. Outlier rejection is done by a simple median filter, where the values greatly differing from the mean of the measurements are rejected. Note that multiple measurements are needed for each beacon position to perform the rejection algorithm.

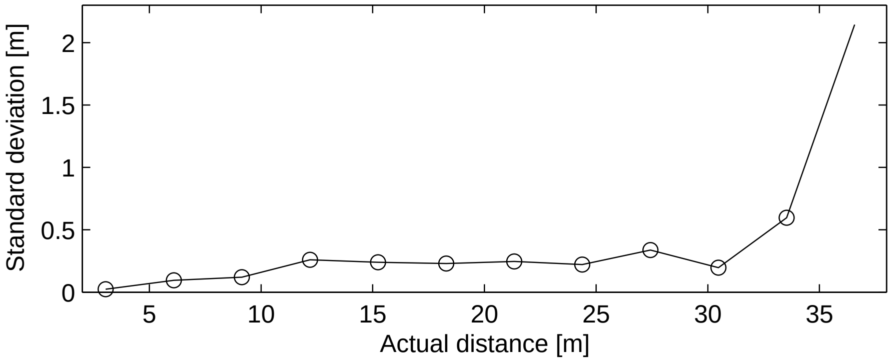


Figure IV.4: Standard deviation of ranging

The effective range of the presented implementation is 30 meters, as the number of outliers and the standard deviation of the measurements are getting significantly high above this value. Below 30 meters the standard deviation grows approximately linearly, with

$$STD \cong k_1 d + k_2 \tag{IV.1}$$

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

The effective range of the measurements is more than two times larger than in previous acoustic ranging experiments [18, 25], where the reliable range was 10 m on asphalt and 15 m on grass, respectively. The standard deviation is also significantly improved. In [18], the output power of the sounder was limited (88 dB at 10 cm from source) and no custom DSP board was used. In [25] 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.

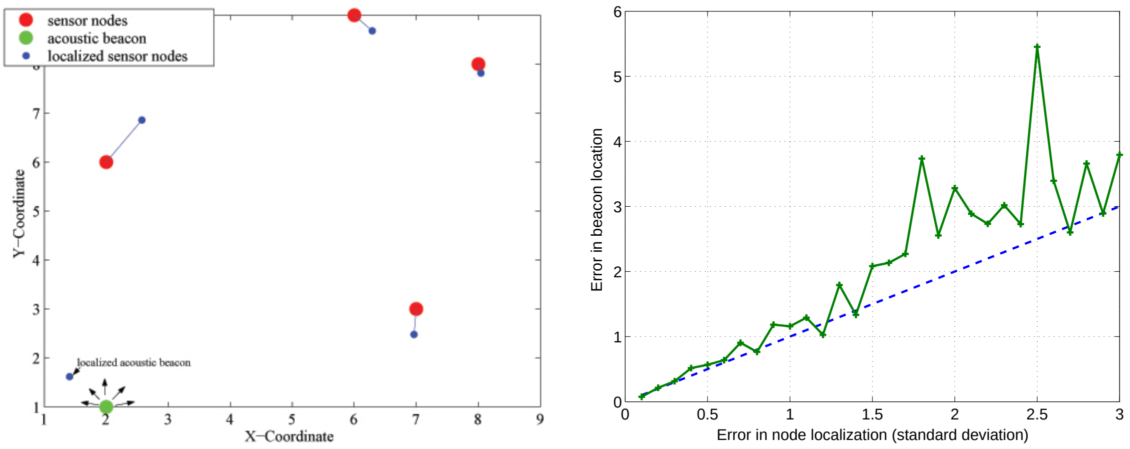
These experimental results are very promising and justify the presented approach. Moreover, the current limits on range and precision are primarily caused by issues with the current implementation. First, the power of the emitted acoustic signal is still constrained by the gain on the output channel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

CHAPTER V

SELF LOCALIZATION

As described in chapter I, high node localization accuracy is an absolute necessity for the correct operation of location-critical applications, such as shooter-localization. The application uses TOA of acoustic signals and locations of the nodes that detect the muzzle blast of shooter. If the node locations are erroneous, the error will propagate to the shooter localization error.

An illustration to show the effect of node localization accuracy on acoustic-beacon localization is presented below. The simulation setup consists of four sensor nodes and an acoustic beacon in 2D plane. The goal of the simulation is to localize the beacon, given the TOA of acoustic signal from beacon to each of the sensor nodes. Sensor node locations are given with zero mean and given standard deviation Gaussian error. The simulation uses least-squares optimization for beacon localization. Figure V.1(a)



(a) (b)

Figure V.1: (a) Simulation setup. Big dots denote actual node/beacon location while the small dots indicate computed locations. (b) shows the behavior of beacon localization error w.r.t. the standard deviation in node location error.

shows the 2D simulation setup while Figure V.1(b) shows the beacon localization error against standard deviation in sensor node location error. Beacon localization error increases at least linearly with sensor node location error. Hence, it is very important to have accurate sensor node locations for accurate operation of location-critical applications.

Formalization

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. Few anchor nodes can be provided to transform relative node locations to absolute positions.

There are many challenges to be addressed in this problem. First we define some terms that we will use in the rest of the document.

Distance Matrix

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.

$$D = \begin{bmatrix} 0 & d_{12} & \dots & d_{1n} \\ d_{21} & 0 & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & 0 \end{bmatrix} \tag{V.1}$$

Necessary Condition for Localization

Necessary Condition for Localization in 3 dimensions states that a node should have distance measurements with at least four non-coplanar neighbor nodes.

Following figure illustrates the necessity of four non-coplanar neighbor nodes. Figure V.2 shows the locus of node location with number of neighbors. In case of one neighbor the locus is a sphere; for two neighbors the locus is a closed curve that is intersection of two spheres (a circle). For three neighbors the locus is reduced to two points. We need fourth neighbor to get rid of this mirror ambiguity.

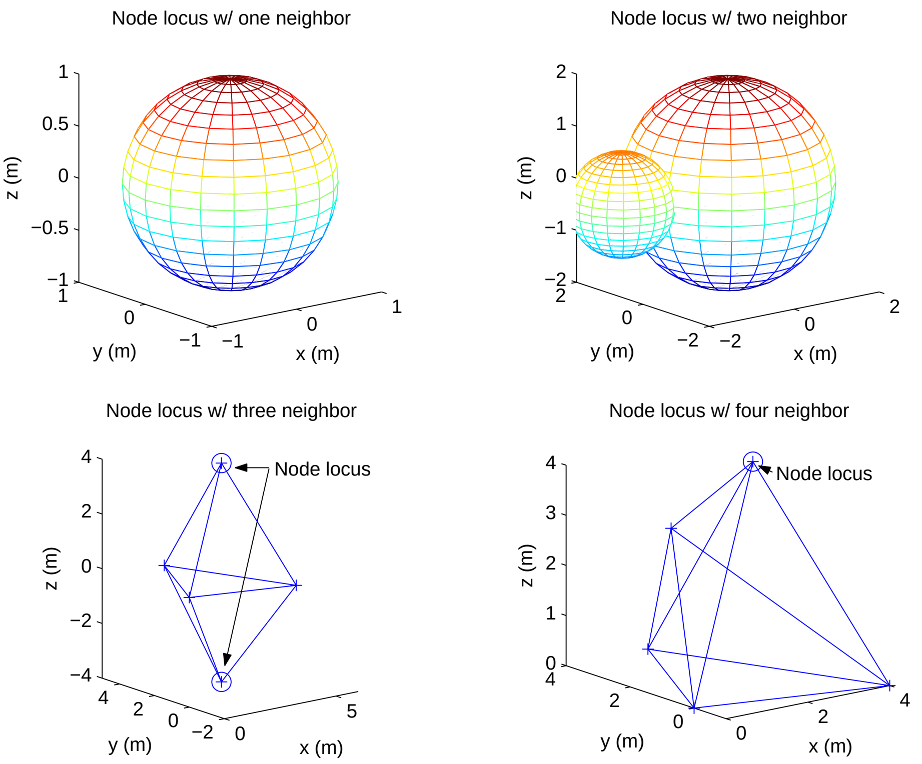


Figure V.2: Node Locus

Multi-path

In a typical urban environment many sensor nodes might not have line-of-sight with the mobile beacon but they can receive the acoustic signal via multipath. These multipath ranges or echoes when used for localization tend to produce false or infeasible results. Consider the following 2-dimensional example (Figure V.3) for illustration. Sound sources S_1 and S_2 have line-of-sight with sensor node U but the direct path to

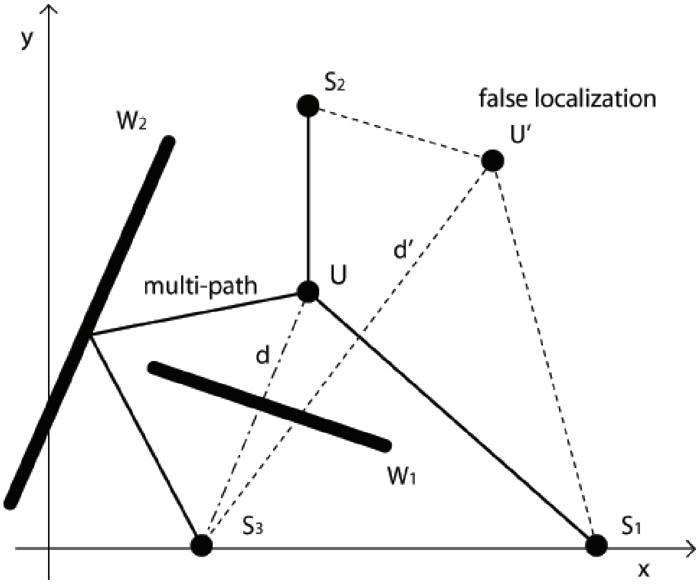


Figure V.3: Illustration of effect of echo.

sound source S_3 is obstructed by wall W_1, thus U receives the acoustic signal from S_3 via multipath through wall W_2. Due to this echo the range measurement between S_3 and U is d' instead of the actual value d. Based on these range measurements, node U' is localized away from its actual location U. If the wall W_2 were moved little farther the echo distance d' would increase and at some point no feasible location would exist for node U. Similar illustration can be drawn for a 3-dimensional case. It is clear from the above illustration that range measurements with echoes would not only *prevent* some nodes to be localized but also—even worse—cause *mis*-localization.

Notice also that the amount of echoes present in the range measurements depends on the topology. We found no way of estimating the extent of echoes in the measurement data. In typical urban environments, low network connectivity and non-uniform node distribution in the vertical (Z-)direction further deteriorate the localization accuracy. These effects can cause even larger localization errors at boundary nodes.

Error Model

For most measuring instruments, the actual physical measurement is affected by some sources of errors that we call noise. The most common form of noise is the white Gaussian noise, which exhibits a Gaussian distribution with zero mean and an instrument or measurement process dependent standard deviation.

Ranging measurements also show white Gaussian noise for a "near ideal" environment, like an outdoor field where all sensor nodes have direct line of sight without any obstructions. Multipath propagation in urban or complex environments causes non-Gaussian errors in ranging measurements.

Most of the localization systems do not consider non-Gaussian errors in ranging error distribution. Considering the non-Gaussian nature of the ranging error is essential to solve the localization problem for satisfactory results. There are many ways ranging error distribution can be modeled. It can be considered as non-parametric distribution or as combined distribution of Gaussian and chi-squared distribution.

This work doesn't directly address the error modeling problem and doesn't explore the non-Gaussian nature of ranging errors. This work does acknowledge that ranging are non-Gaussian, and deals with them by using a technique called pruned least-squared optimization. Pruned least-squared optimization is described later in this chapter.

Distance Optimization

The self localization problem in its most basic form can be modeled as a distance optimization problem. In the distance optimization problem 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. It can be observed that the distance optimization problem is actually a function-fitting problem where distances are the non-linear functions of node locations. Least squares optimization is known to work best for function-fitting problems [17]. The mathematical formulation of distance optimization problem is presented below.

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

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

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. The optimization variable is $\mathbf{x} = [x_1y_1z_1x_2y_2z_2\ldots x_ny_nz_n]^{\mathrm{T}}$ where $[x_iy_iz_i]$ 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 measurement exists ($\hat{d}_{ij} \geq 0$). The components of the optimization variable $\mathbf{x}$ are subjected to the boundary value constraints.

$$\begin{aligned} x_{min} &\leq x_i \leq x_{max} \\ y_{min} &\leq y_i \leq y_{max} \\ z_{min} &\leq z_i \leq z_{max} \end{aligned} \tag{V.3}$$

Pruned Distance Optimization

As mentioned in section V we have non-Gaussian error in the form of 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 logical not to consider these outliers in optimization. For this reason the optimization problem described in previous section is modified to account for probable outliers. The definition of the new optimization problem requires an operator *min* which is defined below.

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 $\lceil \; \rceil$ 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 $\lceil \; \rceil$ is ceiling operator and $0 \leq p \leq 1$.

The mathematical formulation of the new distance optimization problem is presented below.

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

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N, \hat{d}_{ij} \geq 0} \min_p \left(d_{ij} - \hat{d}_{ij} \right)^2 \tag{V.4}$$

where $\hat{d}_{ij}$ and d_{ij} are the range measurement and distance computed from localized nodes i and j and optimization variable is $\mathbf{x} = [x_1 y_1 z_1 x_2 y_2 z_2 \ldots x_n y_n z_n]^{\mathrm{T}}$.

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. In other words, near the global minimizer all consistent range measurements will have function evaluations with values close to zero while echoes, which are non-consistent range measurements, will produce function evaluations with larger values.

Least-square optimization works best if the errors have Gaussian distribution. When we discard few largest function evaluations using the *min* operator, we are discarding the most significant outliers in the distribution and hence obtaining an approximate Gaussian distribution. Figure V.4 illustrates pruned optimization approach for a line fitting problem. In Figure V.4(a) all the data points are considered in optimization. The resulting curve is highly offset from the actual curve due to the non-Gaussian errors. In Figure V.4(b) data points with large deviations from optimized solution are removed from computation and curve is re-optimized for remaining data points. Figure V.4(c) and (d) show the subsequent steps; the final solution is very close to the actual curve.

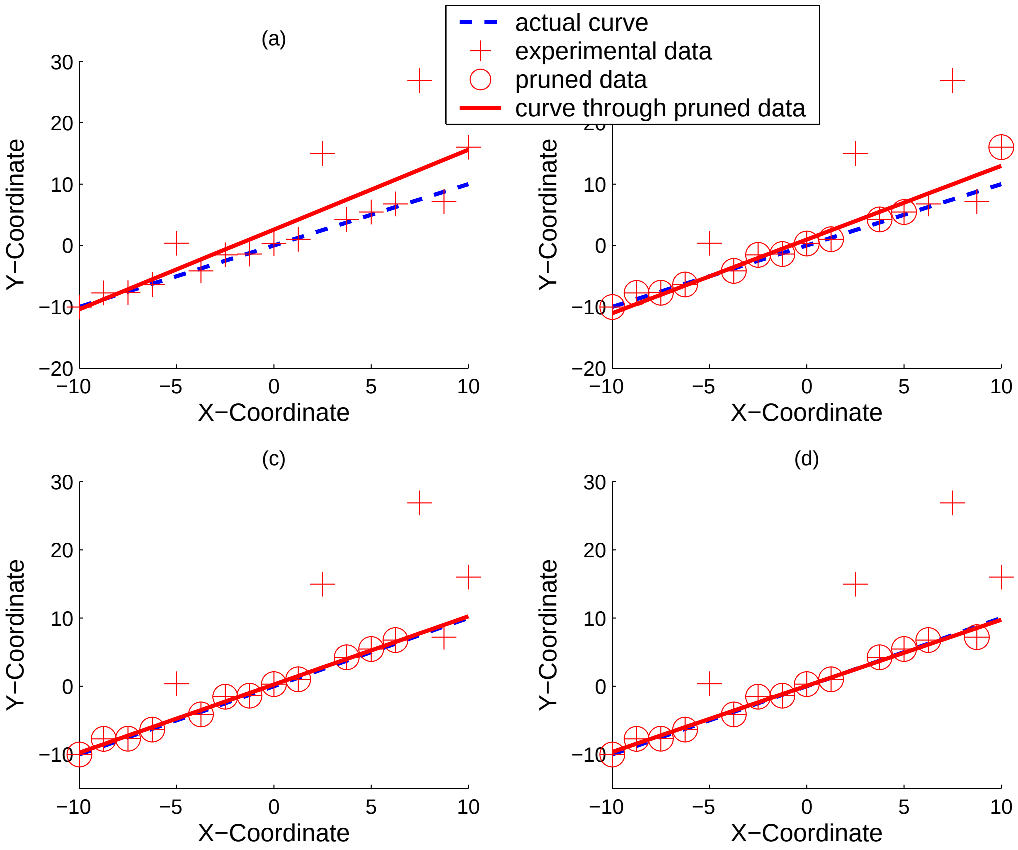


Figure V.4: Illustration of pruned least square optimization

Penalty Functions

The standard optimization solvers used in this work were designed for unconstrained optimization. The bounded-value constraints on the optimization variables are incorporated separately 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. Other forms of penalty functions are linear or quadratic penalties which grow 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. Figure V.5 shows the comparison of different penalty functions. The least-square optimization problem for penalty functions is defined below.

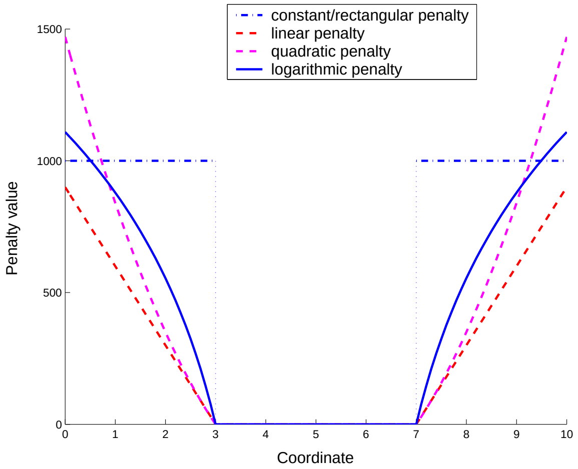


Figure V.5: Penalty functions

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

$$F(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{N} \{\kappa \cdot \ln(1 + \Delta x_{\text{off},i})\}^2 \tag{V.5}$$

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

$$\Delta x_{\text{off},i} = \begin{cases} |x_i - x_{min}| & \text{if } x_i < x_{min} \\ 0 & \text{if } x_{min} \leq x_i \leq x_{max} \\ |x_i - x_{max}| & \text{if } x_i > x_{max} \end{cases} \tag{V.6}$$

and optimization variable $\mathbf{x} = [x_1 y_1 z_1 x_2 y_2 z_2 \ldots x_n y_n z_n]^{\mathrm{T}}$.

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 objective function

$$F_P(\mathbf{x}) = \sum_i^N f_i(\mathbf{x}) + \sum_j^M g_j(\mathbf{x}) \tag{V.7}$$

CHAPTER VI

LOCALIZATION ALGORITHM

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

Algorithm 1 Self localization algorithm

```
Consider 3D coordinates of all unknown nodes in optimization
  variable.
Construct and solve non-linear least-square optimization problem
  with objective function in equation (V.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 in considerably less time while a bad initial guess for the same problem might end up in local optima. Initial estimates for nodes can be computed by using an extension of the bounding box technique described in [3]. 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.

Iterative Incremental Localization Algorithm

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 here is to iteratively select and localize a part of 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 $\mathbf{x^{sol}}$ denotes estimated and localized node location vectors respectively. N denotes the set of nodes in the network and η (described in section sub-system evaluation later in this chapter) denotes confidence value for the localization.

Algorithm 2 Incremental iterative self localization algorithm

```
$\mathbf{x^{est}} \leftarrow \mathbf{0}$, $\mathbf{x^{sol}} \leftarrow \mathbf{0}$
for $run = 1$ to $run_{max}$ do
  Configure parameters, read distance matrix $d$, set sub-system $\tilde{N} \leftarrow \emptyset$
  repeat
    $\tilde{N}_{old} \leftarrow \tilde{N}$
    Estimate bounding-box $B_i$ $\forall i \in N$
    Choose $x_i^{\mathbf{est}} \leftarrow x \in B_i$ $\forall i \in N - \tilde{N}_{old}$ based on neighbor polling
    Select $\tilde{N} \subseteq N$ such that $x_i^{\mathbf{est}}$ satisfies goodness $\forall i \in \tilde{N}$
    Optimize x for sub-system $\tilde{N}$
    $\mathbf{x^{est}} \leftarrow \mathbf{x}$
    for all $i \in \tilde{N}$ do
      Compute $\eta_i$
      $\tilde{N}_{sol} \leftarrow \emptyset$
      if $\eta_i$ acceptable then
        $\mathbf{x_i^{sol}} \leftarrow \mathbf{x_i}$
        $\tilde{N}_{sol} \leftarrow \tilde{N}_{sol} \cup \{i\}$
      end if
    end for
  until $\tilde{N}_{sol} - \tilde{N}_{old} = \emptyset$
end for
Output $\mathbf{x^{sol}}$
```

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 afterward 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 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.

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.

$$\mathbf{x}_{lb,i} = \min_j \{(\mathbf{x}_{lb,j} - \hat{d}_{ij} \cdot \mathbf{1}), \mathbf{x}_{lb,i}\} \tag{VI.1}$$

$$\mathbf{x}_{ub,i} = \min_j \{(\mathbf{x}_{ub,j} + \hat{d}_{ij} \cdot \mathbf{1}), \mathbf{x}_{ub,i}\} \tag{VI.2}$$

The order in which bounding-box updates should be done is also important. Considering the sensor network as a graph, it turns out that a variant of the topological sorting (Algorithm 3) on the network graph will provide the required node ordering. The main idea in this algorithm is that unknown nodes that are closer to known

Algorithm 3 Topological sort

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

nodes get higher precedence.

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

$$\mathbf{x}_{lb,i} = max\{(\mathbf{x}_i^{\mathbf{est}} - \eta_i \cdot \mathbf{1}), \mathbf{x}_{lb,i}\} \tag{VI.3}$$

$$\mathbf{x}_{ub,i} = min\{(\mathbf{x}_i^{\mathbf{est}} + \eta_i \cdot \mathbf{1}), \mathbf{x}_{ub,i}\} \tag{VI.4}$$

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 more heuristic method involving bounding-box partitioning is used instead. The bounding-box of a node, if larger than a 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 that 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.

$$C_p = \sum_{j \in Neigh(i)} \left| \left\| \mathbf{x}_p - \mathbf{x}_j^{\mathbf{est}} \right\| - \hat{d}_{ij} \right| \cdot \eta_j \tag{VI.5}$$

where $\mathbf{x}_p$ is the center point of partition p. The center point of the partition with minimum polling index is chosen as 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 it's bounding-box satisfies two properties. First, its volume V_i is smaller than a critical volume $\mathbf{V}$ and second, its aspect ratio α_i is greater than a critical $\bar{\alpha}_{adaptive}$. 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 as,

$$\alpha_i = \frac{6\sqrt{3} \cdot V_i}{A_i \cdot d_i} \tag{VI.6}$$

Notice that for nodes with small bounding-boxes an estimate is acceptable even if it has smaller aspect ratio. For this reason the critical aspect ratio is made adaptive, quadratically depending on the bounding-box volume. In the equation below $\bar{\alpha}_{min}$ and $\bar{\alpha}_{max}$ are constants.

$$\bar{\alpha}_{adaptive} = \bar{\alpha}_{max} - \left(\frac{V_i}{\mathbf{V}} - 1\right)^2 \cdot (\bar{\alpha}_{max} - \bar{\alpha}_{min}) \tag{VI.7}$$

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 optima.

We solve the objective function in Equation (V.4) or the combination of objective function in Equations (V.4) and (V.5) 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. The different stages of sub-system localization are described below.

- Stage I. At this stage echo ranges are identified and discarded based on the evaluation of the objective function in Equation (V.4) at the current optimizer $\mathbf{x^{est}}$.

- Stage II. At this stage of the optimization problem, Equation (V.4) is optimized in a fixed number of iterations. The solver is stopped even if the optimizer has not converged. Let us 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 terrain. Optimizing for a fixed number of iterations can be visualized as going downwards a local trench but not going all the way down because that would take unlimited amount of 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, Equations (V.4) and (V.5) are optimized in a fixed number of iterations. The objective function in Equation (V.5) ensures that the variable will fall within the feasible region. The reason for having stage II separate from stage III is that there exist a possibility that the path to the global optimizer goes through a region that might not be part of the feasible region. Figure VI.1 shows a case when such a path exists. The optimization variable from initial

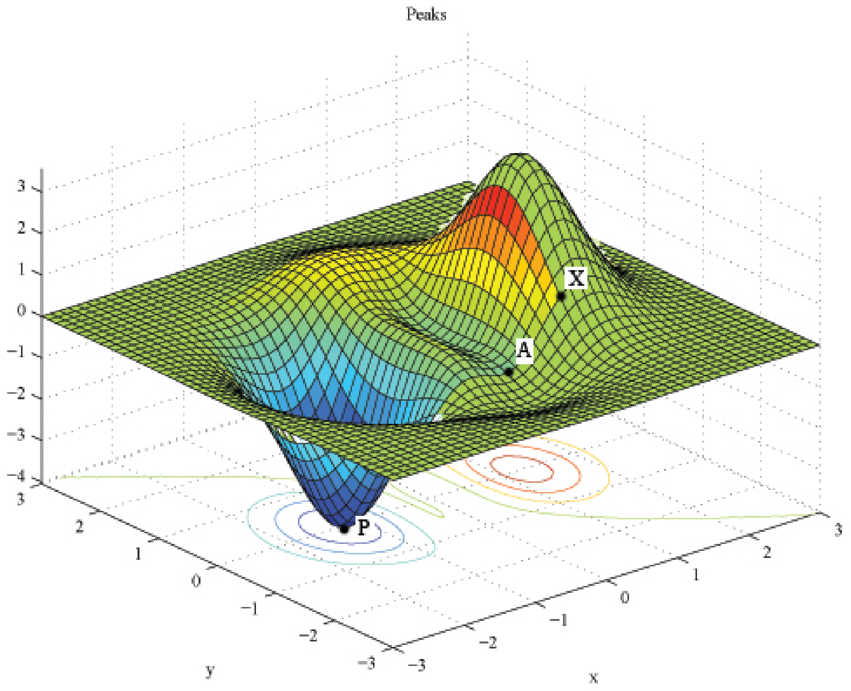


Figure VI.1: An example optimization landscape. $-2 \leq x \leq 1$ and $-1 \leq y \leq 2$ is the feasible region.

guess X will optimize to A while the global optimum is at P. But if we use two stage optimization the optimizer will converge to P.

- Stage IV. This final stage is similar to stage III except parameter p in Equation (V.4) 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.

Sub-System Evaluation

An important measure for any algorithm is its performance metric. In case of self localization the metric can be defined as the closeness of the computed locations to ground truth (actual node locations); we call this the localization error. Since in practical scenarios the ground truth is not known, it would be very helpful if there was an indirect performance metric that correlates with the localization error.

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

The algorithm to compute confidence value is the following. Compute the ranges between computed node locations for all node pairs for which measured range exist. Next compute 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. The categorization of node locations is described below. Confidence value η_i is computed as $\sigma_i \exp(|\mu_i|)$.

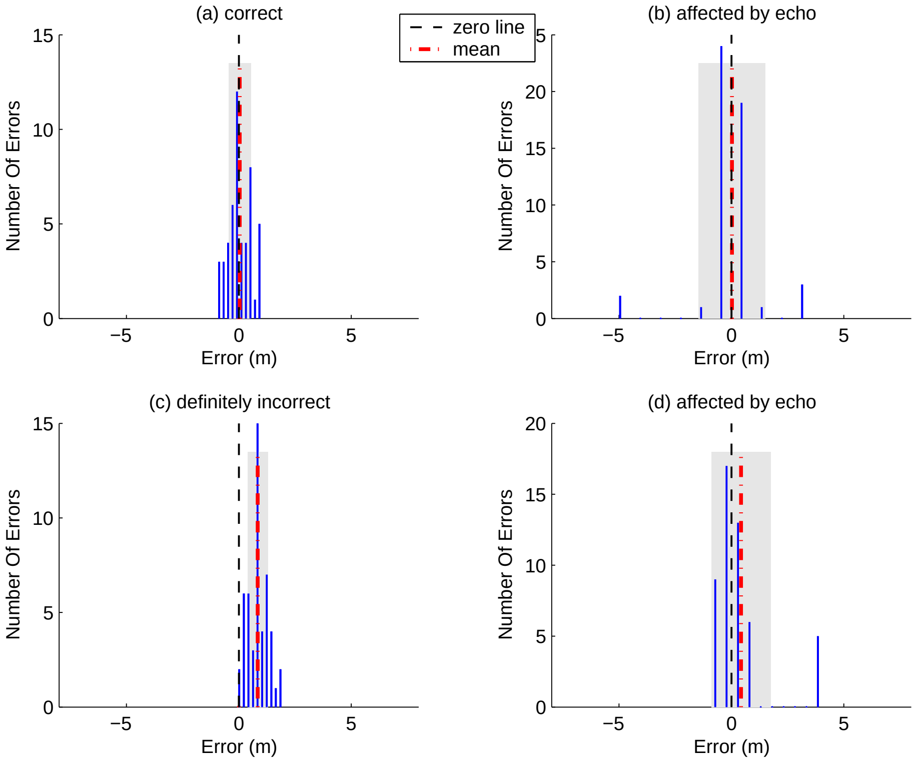


Figure VI.2: Categorization of computed locations based on computed errors. Thin lines show the error distribution while dash-dotted line indicates the mean and grey shaded region show the standard deviation. The dashed line is the zero mean.

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 remove few data points with 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 we followed 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 we followed in case 2 and 4.

The threshold values for mean, μ and standard deviation, σ are set depending on the required localization accuracy. For sub-meter localization, minimum and maximum mean can be 0.05 m and 0.30 m respectively. The minimum and maximum standard deviation can be around 0.15 m and 0.8 m. The above values were used to evaluate the algorithm on simulated topologies, described in next chapter. Figure VI.2 shows the characteristics of each of the above category. Node locations categorized as incorrect or as echoes are considered not localized.

CHAPTER VII

IMPLEMENTATION AND RESULTS

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

Simulated Data Generation

The network topologies and ranging data for localization algorithm were generated in a Java based simulator. The Java simulator generates node locations on sensor network field. The simulator also generates sound source locations arranged on a path. The separation between two consecutive sound source locations is bounded. This simulates the movement of a mobile acoustic beacon that has specified speed and sound signal emitting rate. The simulator then generates ranges between sound sources and nodes in its vicinity. Gaussian noise is added to the ranges according to the experimental observation in Figure IV.4 (Eqn. IV.1). Non-Gaussian errors or echoes and negative-echoes are also added to the ranging data according to the trend observed in experimental data. Experiments show that the probability of a range being an echo grows with the range. The extent of echo, which is the deviation of echo range from actual value, depends entirely on the environment.

The simulator takes number of sensor nodes, number of anchor nodes, number of acoustic beacon paths and number of sound sources on each path as input parameters. Other adjustable parameters include sensor network field size, maximum and minimum separation of consecutive sound sources and maximum range of acoustic signal.

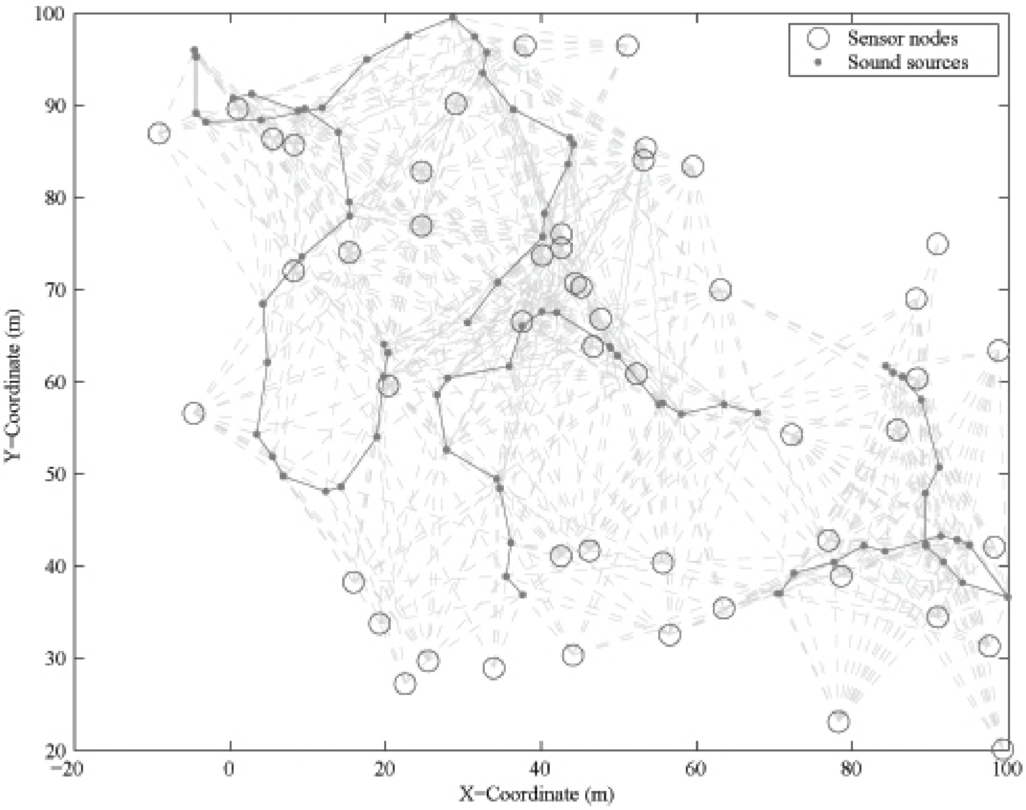


Figure VII.1: XY view of a typical sensor network topology.

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 four random paths such that the separation between successive sound sources is 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. This constraint can be relaxed if we consider mobile beacons to be on a UAV that can vary its altitude. Maximum acoustic range was set to 30 m which is in accordance with our experimental observations. 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 same topology. Figure VII.1 shows XY view of a typical sensor network topology. Faint dashed lines indicate range measurements. Simulated ranging errors are shown in Figure VII.2. Note that the standard deviation in ranging error increases with ranges. Number of non-Gaussian errors also increases with range.

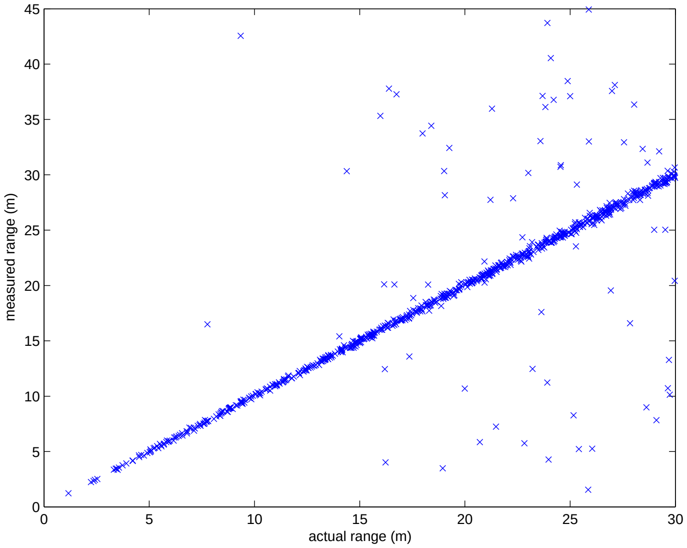


Figure VII.2: Simulated ranging errors

The ranging data generated by the simulator closely emulates the behavior of actual acoustic ranging.

Results

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

$$\sigma_{p,i}^2 = (x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2 + (z_i - \tilde{z}_i)^2 \qquad \text{(VII.1)}$$

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 VII.3 and Figure VII.4 compare the computed node locations to their true locations in XY and XZ views for ranging data with echoes. Different solid lines show multiple paths of the sound source. Solid arrows in Figure VII.3 indicate the sensor nodes that have the highest localization error. Notice that all such nodes are very far from their nearest sound source.

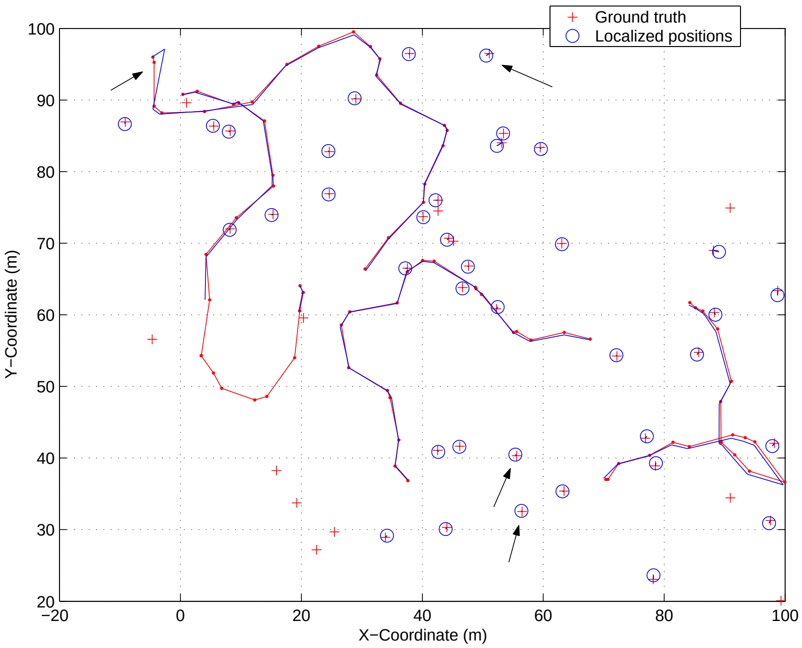
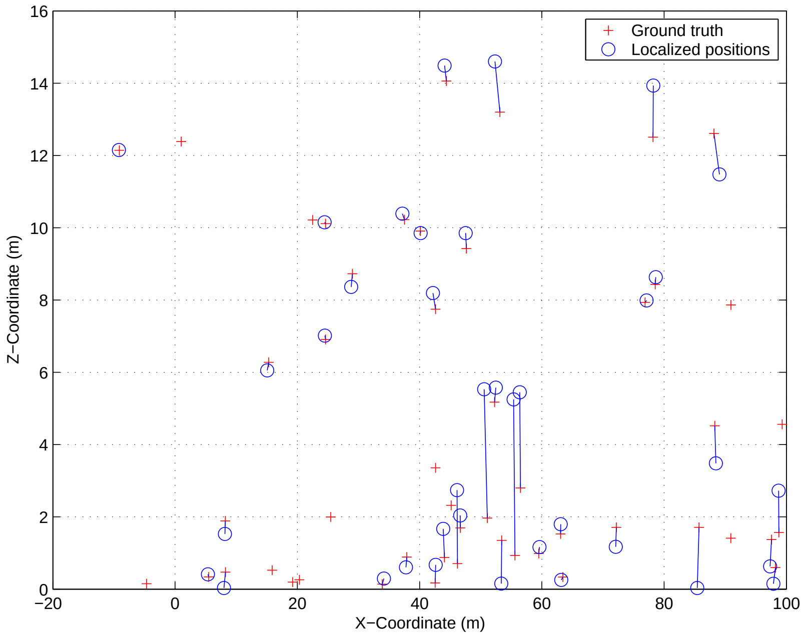


Figure VII.3: Comparison of computed node locations to their true values in XY plane for ranging data w/ echoes.

Figures VII.5 and VII.6 show the histograms of 3D localization error without and with simulated echoes respectively. Table VII.1 summarizes the localization results. The steeper distribution in case (a) indicates that number of nodes with lower localization accuracy is more than that in case (b), where the number of nodes with higher localization errors in significant. In other words, more nodes were localized with better accuracy when we did not have echoes in ranging data.

	Ranges w/o echoes	Ranges w/ echoes
Unlocalized sensors	7	9
Mean error (2D) [m]	0.3041	0.4871
Max error (2D) [m]	2.5436	4.4795
Mean error (3D) [m]	0.8962	1.0664
Max error (3D) [m]	4.3252	4.5119

Table VII.1: Localization results

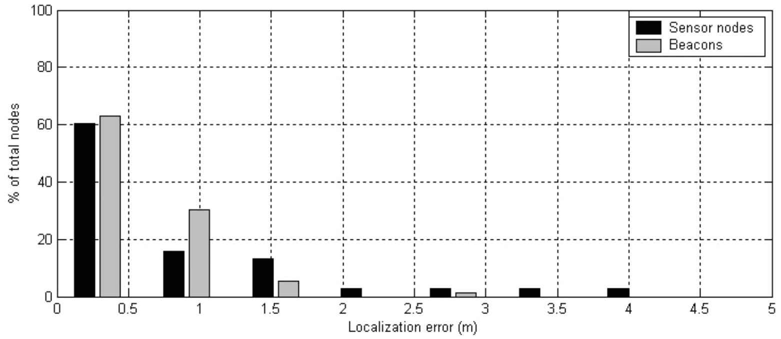


(b)

Figure VII.4: Comparison of computed node locations to their true values in XZ plane for ranging data w/ echoes.

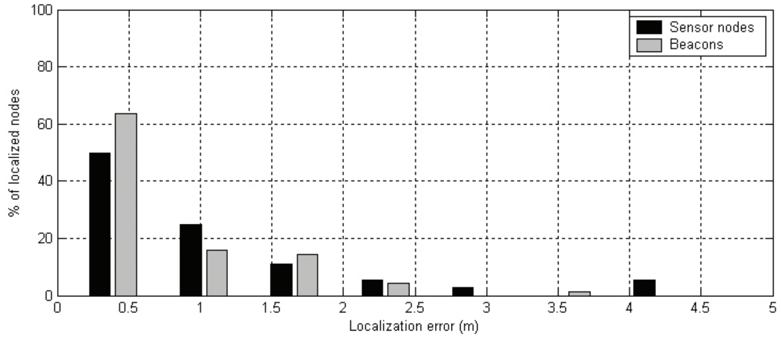
From Figures VII.5 and VII.6 we can see that the computed locations of sound sources are more accurate than that of sensor nodes. This high accuracy can be attributed to the topological fact that sensor nodes are distributed *around* the sound sources. For node localization application we are actually not concerned about the computed sound source 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.

The localization evaluation scheme described in chapter VI was used to evaluate and categorize localization results. Figure VII.7 shows number of sensor nodes in each category. The figure shows localization error distribution for each category. As expected, most of the nodes under correct localization category have small localization error. The categorization algorithm was able to identify four incorrect localization results, two of them have location error less than 1.0 m. Few localization results



case (a)

Figure VII.5: Histograms of 3D localization error for all sensor nodes and sound sources without echoes in ranging data.



case (b)

Figure VII.6: Histograms of 3D localization error for all sensor nodes and sound sources with echoes in ranging data.

with high location errors were categorized as others. Nodes that were classified under affected-by-echo category have 1.0 m or more location error.

Sub-meter localization accuracy is a requirement for correct operation of shooter-localization application. An important concern for node localization is the identification of nodes with high location error. The algorithm localized approximately 75% of the nodes with sub-meter accuracy and less than 5 m maximum error for simulated sensor networks. The categorization scheme identified approximately 50% of the node locations with 2 m or more location error. Based on localization on simulated data,

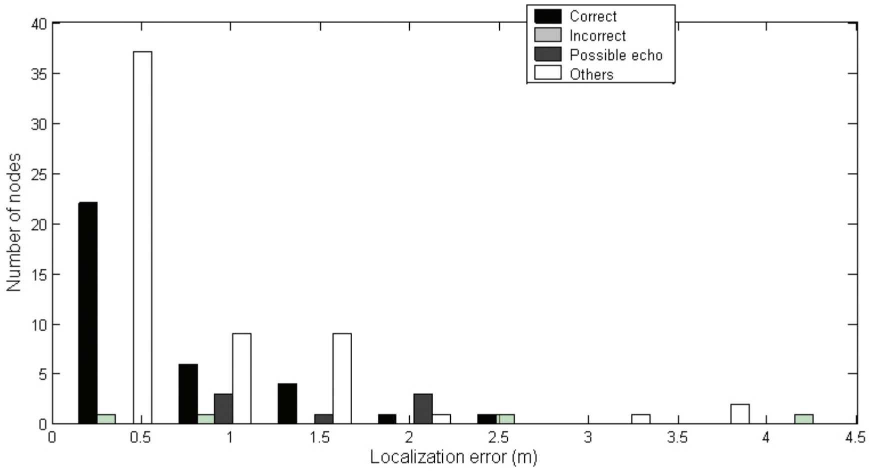


Figure VII.7: Categorization of sensor nodes

the presented localization and categorization algorithms show a potential of close to sub-meter accuracy in a complex urban environment.

CHAPTER VIII

CONCLUSIONS

The presented sensor node localization technique has several contributions. The aerial distribution of sensor nodes from a low-flying UAV and acoustic ranging using a mobile acoustic beacon mounted on UAV is a realistic deployment strategy with many advantages in complex, urban environments.

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. The applied acoustic ranging method has the longest range for mote class devices, even when normalized by the emitted sound energy. This is due to the signal processing algorithms implemented on the sensor board. The sound source installed on mobile UAV can have significant height variation thereby facilitating 3D localization. The uniform speed of UAV with uniform acoustic signal emitting rate provides consistent separation between sound source locations.

The iterative and incremental non-linear optimization technique introduced 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. The algorithm provides sufficient localization results even when number of sound source locations is kept low for practicality. Pruned least-square optimization and other mathematical tools are formally defined and used in localization algorithm. The idea behind pruned least-square optimization is to eliminate the outlier data points in case the error distribution is not Gaussian.

The simulated ranging data used in the evaluation of the localization algorithms are based on experimental observation and are realistic. The localization results produced by the algorithm shows mean localization error under 50 cm in XY and close to 1 m in 3D. The results are satisfactory and assuring for centralized localization.

As a future direction in this work, the localization algorithm can be implemented in Java and integrated with online ranging to provide online node localization in a real urban deployment.

BIBLIOGRAPHY

[1] Mainwaring A., Polastre J., Szewczyk R., Culler D., and Anderson J. Wireless Sensor Networks for Habitat Monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, 2002.

[2] Savvides A., Han C., and B. Strivastava M. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. *Mobile Computing and Networking*, pages 166–179, 2001.

[3] Savvides A., Park H., and Srivastava M. B. The Bits and Flops of the N-hop Multilateration Primitive for Node Localization Problems. In *WSNA*, 2002.

[4] Savvides A., Park H., and Srivastava M. B. The N-hop Multilateration Primitive for Node Localization Problems. In *Mobile Networks and Applications*, pages 443–451, 2003.

[5] Savarese C., Rabay J., and Langendoen K. Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks. In *USENIX Technical Annual Conference*, 2002.

[6] Culler D. and Mulder H. Smart Sensors to Network the World. *Scientific American*, Volume 290, June 2004.

[7] Estrin D., Govindan R., and Heidemann J. Embedding the Internet: Introduction. *Communications of the ACM*, 43(5), 2000.

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

[9] Moore D., Leonard J., Rus D., and Teller S. Robust Distributed Network Localization with Noisy Range Measurements. In *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 50–61, 2004.

[10] Niculescu D. and Nath B. Ad Hoc Positioning System (APS) Using AOA. In *Proceedings of INFOCOM*, 2001.

[11] Whitehouse C. D. The Design of Calamari: An Ad-Hoc Localization System for Sensor Networks. Master's thesis, University of California at Berkeley, 2002.

[12] Simon G., Maróti M., Lédeczi A., Balogh G., Kusy B., Nádas A., Pap G., Sallai J., and Frampton K. Sensor Network-Based Countersniper System. In *SenSys 04*, November 2004.

[13] Dana P. H. Global Positioning System Overview (last modified 5.1.2000). Available from: `www.colorado.edu/geography/gcraft/notes/gps/gps.html`.

[14] Hightower J. and Borriella G. Location Systems for Ubiquitous Computing. *IEEE Computer*, pages 57–66, 2001.

[15] Hightower J., Want R., and Borriello G. SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength. Technical report, 2000.

[16] Hill J. and Culler D. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro*, 22(6):12–24, 2002.

[17] Nodedal J. and Wright S. J. *Numerical Optimization*, pages 250–275. Springer-Verlag New York, Inc., 1999.

[18] Sallai J., Balogh G., Maroti M., Ledeczi A., and Kusy B. Acoustic Ranging in Resource-Constrained Sensor Networks. In *International Conference on Wireless Networks*, 2004.

[19] Teller S. J., Chen J., and Balakrishnan H. IEEE Computer Graphics & Applications: Projects In VR Pervasive Pose-Aware Applications and Infrastructure. *IEEE Distributed Systems Online*, 4(8), 2003.

[20] Römer K., Kasten O., and Mattern F. Middleware Challenges for Wireless Sensor Networks. *SIGMOBILE Mobile Computing and Communications Review*, 6, 2002.

[21] Doherty L., Pister K. S. J., and Ghaoui L. E. Convex Optimization Methods for Sensor Node Position Estimation. In *Proceedings of INFOCOM*, 2001.

[22] Girod L. and Estrin D. Robust Range Estimation Using Acoustic and Multimodal Sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2001.

[23] R. Rabiner L. and Gold B. *Theory and Application of Digital Signal Processing*, pages 715–716. Prentice-Hall, Englewood Cliffs, NJ, 1975.

[24] B. Srivastava M., Muntz R. R., and Potkonjak M. Smart Kindergarten: Sensor-Based Wireless Networks for Smart Developmental Problem-Solving Enviroments. In *MOBICOM*, 2001.

[25] Kwon Y. M., Mechitov K., Sundresh S., Kim W., and Agha G. Resilient Localization for Sensor Networks in Outdoor Environments. Technical Report UIUCDCS-R-2004-2449, Department of Computer Science, University of Illinois at Urbana Champaign, 2004.

[26] Maróti M., Kusý B., Simon G., and Lédeczi Á. The Flooding Time Synchronization Protocol. In *SenSys 04*, November 2004.

[27] Mauve M., Widmer J., and Hartenstein H. A Survey on Position-Based Routing in Mobile Ad-Hoc Networks, 2001.

[28] Bulusu N., Heidemann J., and Estrin D. GPS-less Low Cost Outdoor Localization for Very Small Devices. Technical report, Computer Science Department, University of Southern California, April 2000.

[29] Levanon N. and Mozeson E. *Radar Signals.* Hoboken, NJ John Wiley & Sons, Inc., 2004.

[30] Priyantha N., Chakraborty A., and Balakrishnan H. The Cricket Location-Support System. In *MobiCom'00: Proceedings of the 6th Annual ACM International Conference on Mobile Computing and Networking*, 2000.

[31] Priyantha N., Miu A., Balakrishnan H., and Teller S. The Cricket Compass for Context-Aware Mobile Applications. In *MobiCom'01: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, 2001.

[32] Capkun S., Hamdi M., and Hubaux J. P. GPS-Free Positioning in Mobile Ad-Hoc Networks. In *Proceedings of HICSS*, 2001.

[33] Meguerdichian S., Koushanfar F., Qu G., and Potkonjak M. Exposure in Wireless Ad-Hoc Sensor Networks. In *MOBICOM*, 2001.

[34] He T., Huang C., Blum B., Stankovic J., and Abdelzaher T. Range-Free Localization Schemes in Large Scale Sensor Networks, 2003.