Localization and Sensing Applications in the Pushpin Computing Network

Masters of Engineering Thesis Proposal

Michael Broxton

December 19, 2003

1 Introduction

In the past few years, the rapid miniaturization of sensing, communication, and actuation technologies has led to considerable interest in networks of small sensors. These sensor nodes are self-sufficient electronic devices that include modest sensing and processing capabilities as well as an ability to communicate with neighboring nodes. Networks of sensor nodes are deployed unobtrusively in the environment, sensing physical processes in the real world and forming global estimates of this data by locally processing data among neighboring nodes.

Many of the phenomena of interest to a sensor network are parameterized in terms of time and space. For example, the canonical target tracking problem involves calculating the location and velocity of an object that is traversing the physical space in which the sensor network is deployed. To this end, nodes must know not only their distances to the target, but also the distances to each other so that their range estimates can be compared and the position of the target triangulated. To compute a velocity estimate, nodes must share a common coordinate system in both space *and* time. In general, sensor networks are expected to observe real world events, so it can be expected that physical time and location will play an important role many sensor network applications. Therefore, the problem of *localization and time synchronization* is one of the first that must be tackled by a sensor network when it is initially deployed.

This paper proposes research in localization and time synchronization that is to be carried out on the Pushpin Computing platform, an existing sensor network hardware testbed at the MIT Media Lab. One hundred Pushpin nodes have been modified to include sensors for light, ultrasound, and audio. To aid in localization, a hand-held device has been designed to emit a simultaneous camera flash and ultrasound ping. This unit can be held above the Pushpin sensor network and triggered by a button press. With its sensor suite, a Pushpin node is capable of measuring the time difference of arrival between the flash of light and the ultrasound ping from this device. With this information, the node can make an estimate of the distance between itself and the point where the ultrasound pinger was triggered. This point is called a *correspondence point*.

It is hypothesized that at least two triggers with the ultrasound pinger held at different correspondence points will provide enough information for the sensor network to form a global coordinate system. In addition, all nodes in the network can synchronize their clocks to the camera flash with very high accuracy. Additional flashes and pings can be used to further improve the accuracy of the coordinate system and to resynchronize the clocks on the nodes. In this research, time synchronization is not thoroughly explored. Rather, time synchronization from the camera flash is considered to be a convenient benefit of the system, and is considered an adequate means of time synchronization for our purposes. However, this claim will be revisited throughout the research to verify that this assumption holds in the applications with which we become concerned.

During the course of this project, the efficacy of the above localization approach will be rigorously tested and characterized according to the performance metrics set forth in this proposal. Optimizations and improvements will be pursued which capitalize on the connected nature of the sensor network. Once a final algorithm is found and thoroughly characterized, applications will be chosen which demonstrates the effectiveness of a fully localized and time synchronized sensor network. For example, one possible application is the use of the Pushpin network as an acoustic phased array to determine the direction and distance of an audible sound. The Pushpin network could also be used as an artificial retina if Pushpin nodes with photodetectors were required to determine characteristics of a shadow cast across the network. Such an algorithm would make extensive use of localization information and distributed estimation to determine a shadow's qualities such as the shape, center of mass, and size.

This proposal is divided as follows: Section 2 provides an overview of existing work in localization in sensor networks. Section 3 describes the Pushpin Computing Platform in more detail, as well as the special sensor package and hand-held pinger device that were specifically designed for this experiment. Section 4 proposes one possible method for implementing localization on the Pushpin platform and suggests possible avenues for improvement. Section 4 also describes the performance criteria with which localization implementations will be evaluated. Finally, section 5 summarizes the proposed research and provides a timeline for its completion.

2 The State of the Art

Localization has a rich history in a number of diverse application fields. In each of these fields, the localization system of choice was chosen to balance factors such as system cost, localization accuracy, infrastructure complexity, receiver complexity, and receiver privacy. For example, the GPS system provides coarse-grained localization on a global scale. The GPS receiver is relatively small, efficient, and private but it relies on a multi-billion dollar satellite infrastructure. Despite the recent availability of small, energy-efficient GPS system-on-chip solutions which bring GPS within the size and cost constraints of a sensor network, commercial GPS yields position estimates with error bounds of several meters. This is too inaccurate for use in the Pushpin computing environment, where the entire network is contained in an area of two square meters.

An approach that is more applicable to Pushpin localization is the MIT Cricket localization system [1], which provides localization and orientation with centimeter accuracy for indoor pervasive computing applications. A Cricket receiver wishing to know its position listens passively to several Cricket beacons that have been distributed throughout a room. By measuring the time difference of arrival between a coupled RF signal (which includes the beacon position) and ultrasound chirp from each beacon, the receiver's position can be calculated using triangulation. With five ultrasound transducers arrayed in a "V" shape, orientation can also be measured to within 3 degrees of its actual value [2]. Cricket receivers do not have an innate ability to communicate with each other, thus several Cricket receivers do not constitute a sensor network. However, the Cricket research demonstrates that ultrasonic ranging techniques are viable for providing centimeter-accuracy localization.

There is a growing body of literature that specifically explores localization in sensor networks. The terminology and commonalities in these approaches will be explained next.

Sensor localization typically begins with the establishment of *anchor nodes*. These are a small subset of nodes in the network that have somehow determined their location absolutely or relative to each other. This is a challenging task for a typical sensor node which has limiting processing and power constraints. To work around this, anchor nodes are often "heavy weight" nodes that have been equipped with additional sensing hardware, such as a GPS receiver [3], or with additional processing capability [4]. However, these solutions are not ideal since they require that a second type of node be designed and that these nodes are evenly distributed amongst the non-anchor sensor nodes. For an example of a completely ad hoc approach in a homogenous sensor network, see [5].

Once anchor nodes have been established, the remaining nodes, which are termed *mobile nodes*, must determine their position relative to the anchor nodes. In [6], Langendoen and Reijers assert that in a wide range of localization approaches, the process of determining node location from anchor points can be summarized in three general steps.

For each node,

1. Estimate the distance to several anchor nodes and store anchor coordinates.
2. Calculate node position from anchor distance estimates and anchor coordinates.
3. Refine the node position estimate using additional information (i.e., more anchor measurements, additional sensor readings, etc.).

In the first step, mobile nodes must estimate their distance to several anchor nodes in the network. Various approaches for doing so have been explored, but most utilize logical distance on the network to approximate physical distance [6, 7] . Logical distance in the network is measured minimum number of communication hops between two nodes. The disadvantage to this approach is that the message hop count is not an accurate measurement of physical distance in a sparsely connected network or a network with non-uniform node density. In another approach, a mobile node sends a broadcast packet to the surrounding anchor nodes using its radio transceiver. Anchor nodes reply back with their locations and the the received signal strength from the mobile node [8]. This approach works well in simulation, but performs poorly in practice due to RF asymmetry in a real environment.

The second and third steps can commence once approximate distances to anchor nodes has been established. One approach used in [3] and studied in [6] uses anchor position and distance to establish a bounding box on the coordinates of the mobile node. The area of this bound decreases with an increasing number of anchor nodes. The benefit of this approach is that it is computationally simple. The disadvantage is that the algorithm requires many anchor points, each of which must flood the network with packets containing its location.

Another common approach is called multilateration, which is similar to triangulation and can be formulated as either a direct or iterative approach [9, 6]. In this approach, the coordinates at a node are determined to be the least squares solution to a system of linearized Euclidian distance equations. One advantage of this approach is that a residue of the least squares estimate can be calculated as a confidence check, and any node with a large residue can reject its coordinate estimate and attempt it again. Unfortunately, sensor nodes may not always have the processing power or math libraries necessary to compute a least squared estimate.

The research outlined in this proposal is most similar in design to the Calimari localization system [5]. In the Calimari project, Mica wireless sensor motes [10] perform localization without any external infrastructure to aid them. This is accomplished through a series of inter-node acoustic time of flight measurements made between an RF transmission and a $5.4KHz$ audio signal. Audio ranging in Calimari was chosen because it was less directional than a similarly compact ultrasonic ranging technique, an ideal property in inter-node signaling between sensors on a flat surface. Unfortunately, the ten-fold increase in wavelength between ultrasound and audio causes an order of magnitude decrease in accuracy from a few centimeters to a few tens of centimeters. In the Pushpin Computing platform, where nodes are spaced on the order of single centimeters, lateral ranging with audio is not a sufficiently accurate technique.

3 Pushpin Computing

The proposed research will be carried out on hardware dubbed the Pushpin Computing Platform, a sensor network with 100 individual wireless sensor nodes. The Pushpin Computing nodes (or Pushpins, as they are often abbreviated) are part of an ongoing project in the Responsive Environments research group at the Media Lab aimed at exploring the applications of dense, closely localized sensor nodes distributed on a surface such as a tabletop or a wall. The Pushpin platform is unique in the world of sensor network hardware due to its size, node number and density. Most other platforms involve nodes which are distributed several meters apart, while the Pushpins are spaced on the order of centimeters. Because Pushpins act as a particularly dense array of sensing elements, they are one of the first platforms available for exploring sensor network applications that require high position resolution. The application discussed in section 5 will capitalize on this property.

I have been personally involved with the Pushpin research as an undergraduate researcher since shortly after its inception in 2002. During this time, the Pushpins have evolved into a mature and stable research platform with a user base that includes several research groups at the Media Lab as well as research groups at NASA and Ricoh. The Pushpins are also being used as a distributed particle detector by Professor Peter Fisher in the MIT Physics department. The proposed research is the logical extension of the current Pushpin research effort. It will benefit both the users of the Pushpin platform and, more broadly, general sensor network research in localization.

3.1 Hardware

The "Pushpin" moniker was adopted because each node receives its power and ground connection through two pins that stick out of the bottom of each device. These pins are pressed into a 1.2 meter by 1.2 meter substrate that contain two parallel metallic sheets for power and ground sandwiched between layers of a electrically insulating polyurethane foam. The two metal prongs make contact with these sheets when the pushpin is pressed into the substrate. A network of Pushpins on this substrate is shown in Figure 1

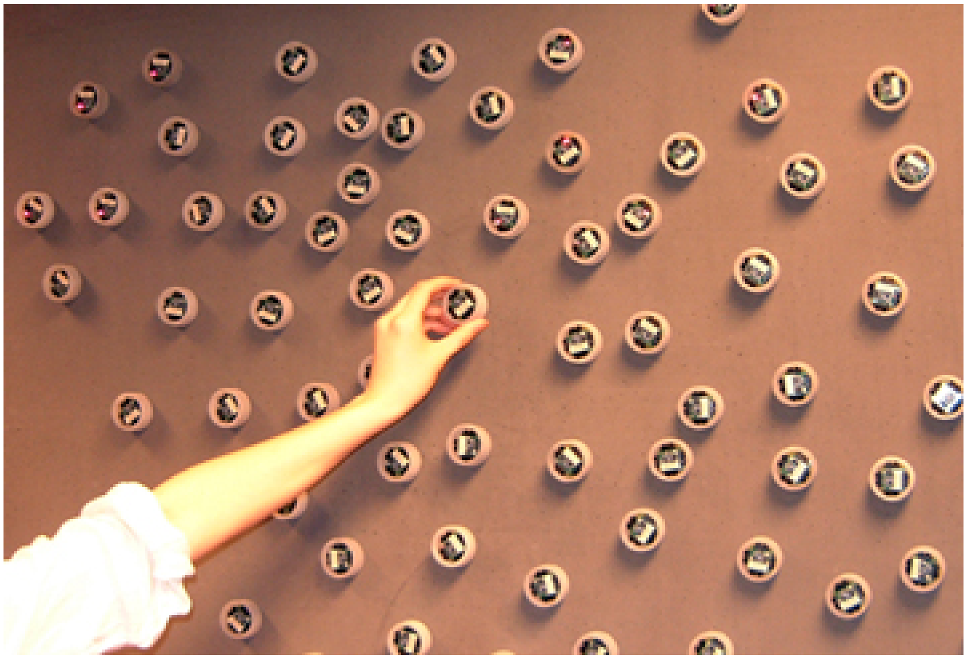

Figure 1: The Pushpin Computing Network

The Pushpins themselves are comprised of three stacked circuit boards; one each for communication, processing, and sensing. The communication layer features infrared tranceivers that enable a Pushpin to communicate to adjacent Pushpins within a radius of roughly 10 centimeters. The core of the pushpin, the processing layer, contains a mixed analog/digital 8-bit micro-controller from Silicon Laboratories (previously Cygnal Systems) [11] running at 22 MIPS. The processor package includes 2.25k of RAM and 32k of non-volatile flash memory.

The top layer in the stack, the sensing layer, contains a variety of different sensors and actuators that allow the pushpin to interact with the world around it. A typical sensing layer includes a photo-transistor or microphone as well as some configuration of LEDs which are used to infer the state of the Pushpin.

The three layers of a Pushpin are modular, which allows different communication and sensing layers to be designed and used for different applications. New sensing layers are particularly easy to design. Because each new sensing layer gives the Pushpin a different sensor package, these layers are often referred to as Pushpin expansion modules. A particular expansion module has been designed to further the research outlined in this proposal. The features of this module are outlined below.

3.1.1 Time of Flight Expansion Module

An algorithm for solving the problem of localization and time synchronization will require accurate range measurements between each Pushpin and a point above the plane containing the Pushpin network. Figure 2 shows the time of flight (TOF) expansion module, which has been designed to make this type of measurement by measuring the time difference of arrival between a camera flash and an ultrasound pulse. The camera flash is measured using a standard photo transistor followed by a differentiator circuit. The module contains a 40 KHz ultrasound transducer and amplifier circuit for receiving the ultrasound pulse. Both the flash and ultrasound detector circuits furnish digital step outputs used to trigger external interrupt lines on the micro-controller.

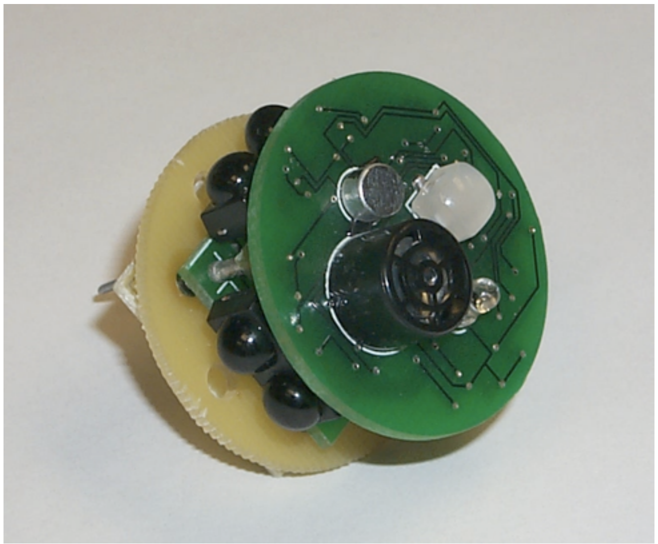

Figure 2: Pushpin node with attached time of flight expansion module.

In addition to these sensing facilities, the TOF module has several other sensing modalities that can be used in future Pushpin network applications. These include an analog light level from the phototransistor and a sonar amplitude envelope from the sonar transducer. There is also an electret microphone circuit that outputs both raw audio and an audio amplitude envelope. Finally, for providing a continuum of colored output information from the Pushpin, the TOF module is equipped with a PWM-controlled RGB LED.

3.1.2 Ultrasound Pinger

The Ultrasound Pinger, or Pinger, is a hand-held device that serves as a common point of reference, a correspondence point, for all Pushpins. It simultaneously transmits a flash from a strobe lamp and a

5ms pulse of 40KHz ultrasound at the press of a button. The time difference of arrival between the light and ultrasound can be measured by each Pushpin. From this measurement, Pushpins can easily infer their distance relative to the point where the Pinger is held. Additional pings from the Pinger at different locations provides additional correspondence points used to improve previous localization estimates. The camera flash on the Pinger is a physical event that is noted nearly simultaneously by all Pushpins. The flash can be used to synchronize the clocks on the Pushpins when time synchronization is desired.

3.2 Software

All Pushpin nodes run an identical operating system called "Bertha." In its current incarnation, Bertha provides the necessary software libraries to access the Pushpin hardware, plus several basic operating system services. Of these, the Bertha communication libraries are the most critical for sensor network applications. Bertha libraries provide access to a rudimentary communication stack that can be easily used to send and receive messages to and from adjacent Pushpins. Though it is written to make use of these libraries and is therefore abstracted and separate from the Bertha operation system, application software, or user code as it is often called, is combined with the operating system when Bertha is compiled. The hybrid of operating system and application code is placed on the Pushpin as one piece of software.

The Bertha operating system runs on top of a boot loader that allows the operating system itself to be easily replaced when newer versions become available. The boot-loader resides permanently in Pushpin flash memory where it intercepts incoming communication, looking for a Bertha update. Since all program code is part of the operating system, this boot loader is the primary means of updating the software on each Pushpin when new algorithms are developed.

Updates are beamed to one Pushpin at a time using an infrared transceiver dongle connected to a serial port of a PC. This update process can be time consuming, so it may become necessary in the course of the proposed research to improve the operating system so that software updates cascade through the entire network after only one Pushpin has been updated by hand.

4 Pushpin Localization

The primary objective in Pushpin localization research will be to create a robust, accurate localization system for the Pushpin Computing platform. The infrastructure necessary for doing this was outlined in section 3. This hardware has been built and tested, so further research will focus on solving the problem outlined in this section.

4.1 Design Goals

Before the Pushpin localization problem is explained in its entirety, it may be helpful to consider the design goals that will guide this research effort.

- *The localization algorithm must be accurate.* Nodes must know their position with an accuracy of one or two centimeters.
- *The localization algorithm must be ad hoc.* Aside from Pinger correspondence points, the network must establish a global coordinate system entirely on its own.
- *The localization algorithm must have low overhead.* Localization and time synchronization must be achieved without significantly impacting communication bandwidth that might be needed for other sensor network tasks. For similar reasons, processor utilization must also be minimal.

- *The localization algorithm must be robust.* It should be relatively unaffected by node failure and range error [6].

These guidelines are not specific to the Pushpin platform; they apply to all sensor network localization. By measuring potential Pushpin localization algorithms according to these general criteria, it is hoped that results will emerge that apply in many sensor network localization contexts.

4.2 Problem Statement

Pushpin localization begins with a flash of light. The Pinger, which is held above the plane of the Pushpin network, simultaneously emits light from a camera flash and a brief ultrasound pulse. The light travels almost instantly to the Pushpins. On seeing a flash, each Pushpin begins a timer . The ultrasound, which is slow by comparison, arrives after the light pulse. When it arrives at a node, the node stops its timer and multiplies the timer value by the speed of sound to determine its distance from the Pinger. Once the ultrasound pulse has reached all nodes, each node will have a distance estimate to a common point, the correspondence point defined by the location of the Pinger. The Pinger can now be moved and triggered again, creating another common point in the space above the plane of the Pushpin network.

Figure 3 shows a basic Pushpin network with four nodes. We will describe a basic approach at localizing all nodes in this network. With this approach, localization of all nodes can be accomplished using three correspondence points from the pinger. We make the simplifying assumption that all correspondence points occur directly above some node in the network, and that these correspondence points are not colinear. In the relatively dense Pushpin network, these assumptions are not unreasonable.

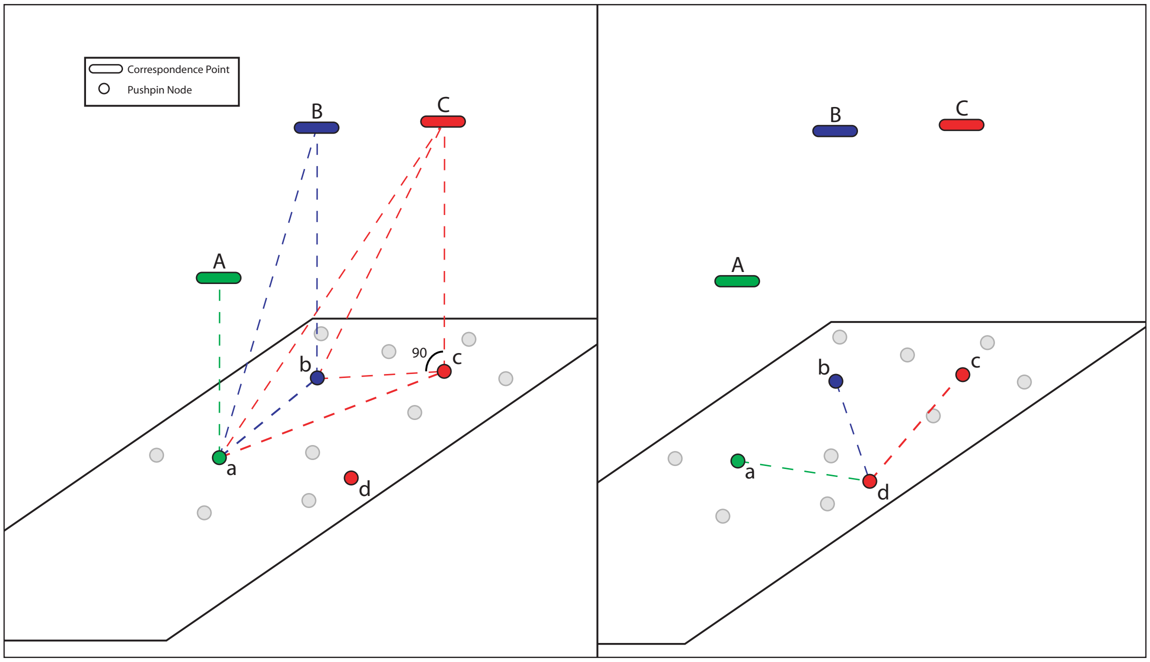

Figure 3: Localization on a simple Pushpin network.

First, we create correspondence point **A** by transmitting a light/sonar ping. The nodes communicate, determining the node which was closest to the pinger. This node is called node **a**, and it is elected the origin of the new coordinate system. Next, we create a correspondence point above node **b**, which we will place arbitrarily on the y-axis. Nodes **a** and **b** communicate, determining the distance α between them using simple right triangle relations. Node **b** now has the coordinates $(0, \alpha)$. Finally, we place a third correspondence point above a node **c**. The lengths of the line segments $\bar{ab}$, $\bar{bc}$, and $\bar{ac}$ are all known. More trigonometry produces a pair of possible coordinates for node **c** that are reflections across the y-axis. For now, this ambiguity will be resolved arbitrarily by picking one of the two sets coordinate randomly. More rigorous approaches will be explored in the course of this research.

At this point, we have three anchor points with known positions and the distance between these anchor points and all the other nodes in the network. We can now use the lateration approach suggested in [6] to compute the coordinates (x, y) of any node in the network.

To find the coordinates (x, y) of node **d**, we proceed as follows. Using Euclidean distance metrics, we see that,

$$\begin{aligned} (x_a - x)^2 + (y_a - y)^2 &= d_a^2 \\ (x_b - x)^2 + (y_b - y)^2 &= d_b^2 \\ (x_c - x)^2 + (y_c - y)^2 &= d_c^2 \end{aligned}$$

All the terms x_i, y_i and d_i for $i \in [a, b, c]$ are known. We can linearize this system by subtracting the third equation from the first two and collecting terms:

$$\begin{aligned} 2(x_a - x_c)x + 2(y_a - y_c)y &= x_a^2 - x_c^2 + y_a^2 - y_c^2 + d_c^2 - d_a^2 \\ 2(x_b - x_c)x + 2(y_b - y_c)y &= x_b^2 - x_c^2 + y_b^2 - y_c^2 + d_c^2 - d_b^2 \end{aligned}$$

From this system of equations, x and y can be computed.

The method above shows that three properly placed correspondence points provide enough information to localize any node in the sensor network. Even with this simple approach, two of our criteria for a localization algorithm have already been satisfied. First, the approach is accurate. All distance information is based on ultrasonic ranging, which has centimeter accuracy. Second, the approach is ad hoc. The coordinate system is built entirely within the sensor network once the correspondence points have been set.

The third criterion, low overhead, is more difficult to assess. The exact amount of communication depends on the implementation, but clearly after each flash and ping from the Pinger, the network must communicate to determine which node was beneath the pinger. This can be cast as a distributed minimization problem, and its dynamics will be a subject of study in this research. In addition to choosing anchor points through distributed minimization, the positions of the three anchor points must at some point be distributed to the entire network so that mobile nodes can estimate their position. The precise overhead incurred by these two communication tasks is not clear.

The final performance metric, robustness, is clearly not addressed in this approach. Pushpin nodes are inexpensive and numerous; the quality of every node cannot be assured. Pushpins sometimes fail, and when they do, the localization must be robust to these failures. In the approach above, a distance error between a correspondence point and an anchor point would result in a set of inaccurate anchor coordinates. This error would then affect the position estimate of every mobile node. Even worse, if an anchor node were to fail before it had shared its coordinates with the network, localization would become impossible using this technique. Clearly, this approach does not meet the criterion of robustness.

In summary, this solution to the Pushpin localization problem is not complete. Not all the performance criteria were met. Additional restrictions were placed on the location and number of the correspondence points. There is clearly room for improvement.

This approach might be improved if we took additional constraints into account. One optimization, for example, would be to use logical adjacency on the network to disambiguate mirror image coordinates such as those encountered when determining the location of node **c**. A possible improvement to robustness would be to check an anchor node's range measurements with the nodes adjacent to it. If the measurements differ by a large amount, the anchor node could signal a distance error. Ideally, enhancements like these will lead to an approach with increased robustness, decreased overhead, and fewer constraints on correspondence points.

5 Applications

Once the critical objective of forming a global coordinate system is complete, the Pushpin platform will be ready for distributed sensing applications. The second objective of this thesis research will be to develop and characterize one or more of these applications on the Pushpin Computing platform. The following two sections explain possible applications that capitalize on accurate position and timing information.

5.1 Acoustic Phased Array

In addition to the light and ultrasound sensors, the time of flight expansion module includes a microphone. Since each Pushpin node is equipped with a microphone, its location in a global coordinate system, and a synchronized global clock, it is reasonable to think of the Pushpin network as an acoustic phased array.

The phased array is a well understood tool for localizing and ranging acoustic, RF, magnetic, infrared, and other signals. The capabilities of a phased array depend on the number of receivers, frequency distribution and distance of the source phenomenon, and properties of the propagation medium. Various techniques are useful in different situations. An overview of phased arrays can be found in [12].

In general, acoustic phased arrays detect signals with wide-band frequency content and a known propagation velocity. Localization of an ongoing sound typically requires some form of distributed signal correlation; a process which is too processor intensive for the computationally constrained Pushpin nodes. However, the more tractable problem of localizing brief, distinct acoustic phenomenon can can be solved without intensive computation. Sounds of this type can be detected in the time domain with simple techniques, such as thresholding. By measuring the incidence of a sound using a threshold, four localized nodes can measure the time difference of arrival of an audio signal and determine the direction and range of a sound using triangulation. Measurements from additional microphone-equipped nodes can be used improve the accuracy of this estimate.

Some previous work is worth mentioning. An acoustic phased array as applied to a wearable computing application is described in [13]. This phased array, which was composed of three microphones, could succesfully determine the direction (but not the distance) of a sound in a room. The use of sensor networks as an acoustic phased array is specifically explored in [14]. Here, a sensor network of time synchronized iPaq handheld computers was used to collect acoustic data for later off-line processing. Their results showed this data could be used to localize a recording of a noisy truck in an outdoor environment.

5.2 Artificial Retina

Another intriguing application of the Pushpin network would be to use the Pushpins as a distributed array of photo-detectors. Such an array could be used to explore the topologies of shapes cast on the network using either shadows or lights. Localization will be useful for determining the physical characteristics of these shapes.

Conventionally, an artificial retina is a densely packed array of CMOS imaging sensors, each of which is attached to a programmable processing element formed from some small number of transistors. Each processing element is programmable and can communicate with the processing element of adjacent pixels. The processing elements of an artificial retina array are programmed to perform simple real-time image processing tasks. One such array capable of real-time edge detection and feature extraction is described in [15] In general, artificial retinas can be used to localize and characterize the shape, center of gravity, size, and motion of an image. These solid state devices are strikingly similar to sensor networks in which nodes are equipped with photodiodes. This suggests that the sensor network may perform well in real time image processing applications.

A sensor network algorithm for shape recognition was demonstrated on the Paintable Computing simulator

[16]. The sensor network in this simulation was localized using logical network distance [7] and used to discriminate between simple shapes by counting the number of corners. This problem is interesting because it involves many regimes of sensor network operation. In addition to requiring localization, shape recognition requires both local collaboration among sensor nodes for detecting a convex object edge and global collaboration for determining the number of distinct edges in an object.

6 Summary of Proposed Research

During the course of this project, I aim to accomplish the following objectives:

- Design a hardware expansion module for the Pushpin platform with sensors that enable a Pushpin to make a Time Difference of Arrival (TDoA) measurement between a flash of light and an ultrasound pulse.
- Write a robust and accurate algorithm for distributed localization using TDoA range measurements.
- Rigorously characterize the above algorithm based on the performance criteria set forth in section 4 of this proposal.
- Create one or more sensor network applications that rely on the globally shared coordinate system, starting with an acoustic phased array.

The first objective was completed during the fall of 2003. The remaining tasks involve developing and characterizing localization and time synchronization software and designing a location-aware application for the Pushpin platform as outlined in section 5. The timeline for completing these objectives can be found in Table 1.

In addition to the primary objectives listed above, I plan to submit a paper to the ACM Conference on Embedded Networked Sensor Systems (SENSYS) during the month of March. In this paper, I plan to summarize work on the first three objectives. Writing this paper will serve as a concrete point of focus, allowing me to judge the progress of this research during the middle of the research term.

Table 1: Proposed Timeline

Date	Task
January 11	Implement the localization algorithm described in section 4.
February 20	Conclude active development of new localization algorithms.
March 15	Complete algorithm characterization. Begin writing paper for SENSYS conference.
April 2	SENSYS 2004 paper submission deadline.
May 1	Conclude work on the sensor network application outlined in section 5.
May 10	Complete first draft of thesis.
May 20	Thesis Due.

References

[1] Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: Mobile Computing and Networking. (2000) 32–43

[2] Priyantha, N.B., Miu, A.K.L., Balakrishnan, H., Teller, S.: The Cricket Compass for Context-Aware Mobile Applications. In: Proceedings of the 6th ACM MOBICOM Conference. (2001)

[3] He, T., Huang, C., Blum, B., Stankovic, J., Abdelzaher, T.: Range-Free Localization Schemes in Large Scale Sensor Networks. In: Proceedings of the 9th annual international conference on Mobile computing and networking. (2003) 81–95

[4] Girod, L., Bychkobskiy, V., Elson, J., Estrin, D.: Locating tiny sensors in time and space: A case study. In: Proceedings of the International Conference of Computer Design (ICCD) 2002. (2002)

[5] Whitehouse, K.: The Design of Calamari: an Ad-hoc Localization System for Sensor Networks. Master's thesis, University of California at Berkeley (2002)

[6] Langendoen, K., Reijers, N.: Distributed localization in wireless sensor networks: a quantitative comparison. The International Journal of Computer and Telecommunication Networking **43** (2003) 499 – 518

[7] Nagpal, R., Shrobe, H., Bachrach, J.: Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03) (2002)

[8] Simic, S.N., Sastry, S.: Localization in dense random sensor networks . Submitted to Information Processing in Sensor Networks (IPSN) 2004 (2003)

[9] Savvides, A., Han, C.C., Strivastava, M.B.: Dynamic fine-grained localization in ad-hoc networks of sensors. In: Mobile Computing and Networking. (2001) 166–179

[10] Crossbow Technologies: Intertial, Gyro, and Wireless Sensors from Crossbow. http://www.xbow.com (2003)

[11] Silicon Laboratories: Silicon Laboratories Homepage . http://www.silabs.com (2003)

[12] P.Mailloux: Phased Array Antenna Handbook . Artech House, Boston (2004)

[13] Basu, S., Schwartz, S., Pentland, A.: Wearable Phased Arrays for Sound Localization and Enhancement. In: Fourth International Symposium on Wearable Computers (ISWC'00). (2000) 103

[14] Chen, J., Yip, L., Elson, J., Wang, H., Maniezzo, D., Hudson, R., Yao, K., Estrin, D.: Coherent Acoustic Array Processing and Localization on Wireless Sensor Network. Proceedings of the IEEE **98** (2003)

[15] Paillet, F., Mercier, D., Bernard, T.: Second Generation Programmable Artificial Retina. In: Proceedings of the IEEE ASIC/SOC Conference. (1999) 304–309

[16] Butera, W.: Programming a Paintable Computer. PhD thesis, Massachusetts Institute of Technology (2002)