Sensor Network-Based Countersniper System

Gyula Simon   Miklós Maróti   Ákos Lédeczi
György Balogh   Branislav Kusy   András Nádas
Gábor Pap   János Sallai   Ken Frampton

Institute for Software Integrated Systems
Vanderbilt University
2015 Terrace Place, Nashville, TN 37203, USA
Phone: (+1) 615- 343-7472

e-mail: {gyula.simon, miklos.maroti, akos.ledeczi}@vanderbilt.edu

ABSTRACT

An ad-hoc wireless sensor network-based system is presented that detects and accurately locates shooters even in urban environments. The system consists of a large number of cheap sensors communicating through an ad-hoc wireless network, thus it is capable of tolerating multiple sensor failures, provides good coverage and high accuracy, and is capable of overcoming multipath effects. The performance of the proposed system is superior to that of centralized countersniper systems in such challenging environment as dense urban terrain. In this paper, in addition to the overall system architecture, the acoustic signal detection, the most important middleware services and the unique sensor fusion algorithm are also presented. The system performance is analyzed using real measurement data obtained at a US Army MOUT (Military Operations in Urban Terrain) facility.

Categories and Subject Descriptors

B.7.1 [Integrated Circuits]: Types and Design Styles – *Algorithms implemented in hardware*, C.2.2 [Computer-Communication Networks]: Network Protocols – *Routing protocols,* G.1.0 [Mathematics of Computing]: Numerical Analysis – *Numerical algorithms*, J.7. [Computer Applications]: Computers in Other Systems – *Military*

General Terms

Algorithms, Design, Measurement, Performance

Keywords

Sensor Networks, Middleware Services, Time Synchronization, Message Routing, Data Fusion, Acoustic Source Localization

1. INTRODUCTION

Detecting and accurately locating shooters has been an elusive goal of armed forces and law enforcement agencies for a long time now. Among the several systems developed in the past decade only a few can be used in such challenging environments

as urban terrain. 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 phenomena can be used for sniper detection purposes. The Viper system built by Maryland Advanced Development Lab utilizes an infrared camera to detect the muzzle flash of the weapon [17]. It is augmented with a microphone to detect the muzzle blast for range estimation. Both sensors require direct line of sight. Other limitations include the possibility of flash suppression by the shooter and a relatively high false alarm rate that is reduced by employing two disparate sensors [21]. Another approach measures the thermal signature of the bullet in flight [21]. Illuminating the sniper's scope with a laser and measuring the reflections can also provide accurate bearing estimates [21]. None of these approaches, however, provide a comprehensive solution to the problem.

Despite the efforts of using different information sources for sniper detection, so far acoustic signals, such as muzzle blasts and shock waves provide the easiest and most accurate way to detect shots, and the majority of the existing countersniper systems use them as the primary information source [20]. The most obvious acoustic event generated by the firing of a conventional (non-silenced) weapon is the blast. The muzzle blast is a loud, characteristic noise originating from the end of the muzzle and propagating spherically away at the speed of sound. Typical rifles fire projectiles at supersonic velocities, thereby producing acoustic shocks along their trajectory [20]. Shockwaves can be used to accurately determine projectile trajectories, because the shock waveform is distinctive and cannot be produced by any other natural phenomenon. The simplified geometry of the bullet trajectory and the associated muzzle blast and shockwave fronts are shown in Figure 1.

Commercial acoustic sniper detection systems use these phenomena. They measure the time of arrival (TOA) and some other characteristics of shockwaves and/or the TOA of muzzle blasts. BBN's *Bullet Ears* system utilizes one or two small arrays of microphones, providing estimates of the caliber, speed and trajectory of the projectile, and also a range estimate for the shooter. The average accuracy of the azimuth and elevation estimators is approximately 1.2 and 3 degrees, respectively, while the distance estimator's accuracy is approx. 1.6% [4]. The similar French *Pilar* system uses two microphone arrays achieving bearing and range accuracy of ±2° and ±10%, respectively [11].
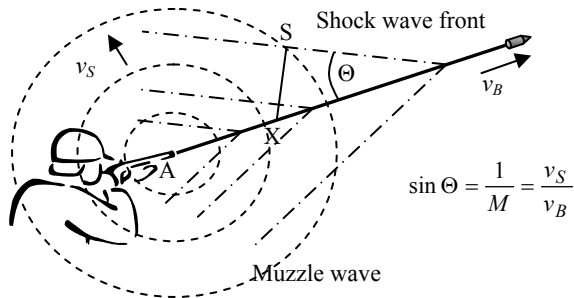
Figure 1. Acoustic events generated by a shot. The muzzle blast produces a spherical wave front, traveling at the speed of sound (v_s) from the muzzle (A) to the sensor (S). The shock wave is generated in every point of the trajectory of the supersonic projectile producing a cone-shaped wave front, assuming the speed of the projectile is constant v_B. (In reality, the wave front is not a cone, rather it resembles the surface of a half football, since the bullet is continuously decelerating.) The shockwave reaching sensor S was generated in point X. The angle of the shockwave cone is determined by the Mach number (M) of the projectile.

The main drawback of the current centralized systems is that if some of the few sensors cannot detect the signal then the system does not have enough data to perform the localization accurately. Measurement errors can easily occur if the sensors do not have direct line-of-sight of the shooter (no muzzle blast detection) or the projectile trajectory is shaded (no shockwave detection). An even more troublesome source of error is when the sensors pick up echoes resulting in poor localization accuracy.

A straightforward solution can be the utilization of many sensors providing good coverage in a large area of interest. In this way there is a high probability that multiple sensors detect the direct signal. The individual sensor measurements can be less accurate, since the measurements are independent and come from different locations; thus the sensors can be less sophisticated and much smaller. Using large number of sensors not only enhances the accuracy, but it also increases the robustness of the overall system.

Based upon the above idea, we developed an experimental countersniper system called PinPtr. The system utilizes an ad hoc wireless sensor network built from inexpensive sensor nodes. After deployment, the sensor nodes synchronize their clocks, perform self-localization and wait for acoustic events. The sensors can detect muzzle blasts and acoustic shockwaves and measure their time of arrival. Utilizing a message routing service, the TOA measurements are delivered to the base station, typically a laptop computer, where the sensor fusion algorithm calculates the shooter location estimate. The base station also acts as the primary user interface. Optional PDAs can act as secondary user interfaces. They get their data from the base station through an 802.11 wireless network. The system was field tested multiple times at the US Army McKenna MOUT (Military Operations in Urban Terrain) facility at Fort Benning, GA. The average localization accuracy was around 1m, while the observed latency was less than 2 seconds.

The rest of the paper is organized as follows. In the next section we describe the sensor network platform – both hardware and
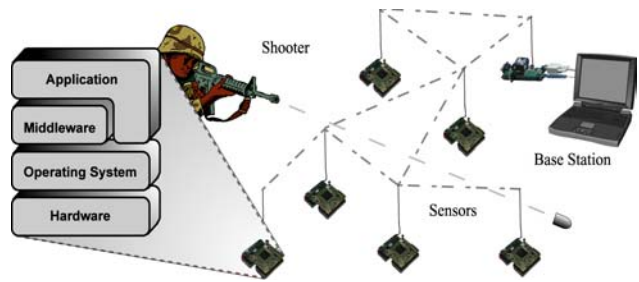

Figure 2. System architecture

software – and the overall system architecture. Next the middleware services utilized in the application are presented. Then we briefly summarize the signal detection algorithm performed on the sensor nodes. The sensor fusion algorithm is also presented followed by a comprehensive analysis of the experimental results gathered during field trials in an urban environment. Finally, we present our future plans and conclusions.

2. SYSTEM ARCHITECTURE

The countersniper application utilizes the traditional layered architecture, as shown in Figure 2. The hardware layer is built upon the widely used Mica mote platform, developed by UC Berkeley [10]. The second generation Mica2 features a 7.3 MHz 8-bit Atmel ATmega 128L low power microcontroller, a 433 MHz Chipcon CC1000 multi-channel transceiver with 38.4 kbps transfer rate and a maximum practical range of 200 feet, 4 kB RAM and 128 kB flash memory. The motes also have an extension interface that can be used to connect various sensor boards containing photo-, temperature-, humidity-, pressure sensors, accelerometers, magnetometers and microphones.

The Mica motes run a small, embedded, open source operating system called TinyOS by UC Berkeley [7], specifically designed for resource limited networked sensors [9]. Despite its small footprint, this event driven OS can handle task scheduling, radio communication, clocks and timers, ADC, I/O and EEPROM abstractions, and power management. These services are implemented as components and the application can be composed from them in a hierarchical manner. The system resources are preserved by using only those OS components that are actually needed by the application.

The Mica2 motes are connected to our multi-purpose acoustic sensor boards (see Figure 3), designed with three independent acoustic channels (each with a microphone, amplifier with controllable gain, and ADC operating at up to 1MHz), and a Xilinx Spartan II FPGA. The FPGA chip implements the signal processing

Figure 3. Custom sensor board and Mica2 mote.

algorithms to classify acoustic events as muzzle blasts, shockwaves (or none of the above) and measures their times of arrival. The three acoustic channels utilizing the microphones placed 2 inches apart provide means to measure direction of arrival accurately. However, this feature is not utilized in the current application because it would require knowing the orientation of each node in addition to its position. Performing accurate self orientation is a non-trivial task. Instead, only a single acoustic channel is used on each board to measure TOA information and then data from multiple nodes are fused on the base station.

3. MIDDLEWARE SERVICES

The PinPtr application uses several middleware services implemented on TinyOS; the most important ones being time synchronization, message routing with data aggregation and self-localization.

Precise time synchronization is essential in the application, since TOA data are used to determine the location of the shooter, and the measurements are provided by a large number of independent sensors only linked through a radio channel. The more precise the alignment of events in time is, the more accurate the sensor fusion is, and the less sensitive to multipath effects the solution becomes. PinPtr uses the Flooding Time Synchronization Protocol described in [15] and summarized in section 3.1.

To provide message delivery with small latency using the limited-bandwidth radio channel of the Mica motes in a multi-hop network is not a straightforward task. The problem is further exacerbated by the inherent nature of the application: namely the sensors try to send their measurements at approximately the same time and to the same destination. The applied solution is tailored to this situation and combines the routing with a data aggregation protocol, providing small response time, even if a large number of sensors are triggered and need to send data simultaneously.

A self-localization service is used to provide sensor location data for the fusion algorithm. This service uses radio and acoustic signals for pair-wise ranging and an optimization algorithm to determine the relative positions of sensors.

3.1 Time synchronization

The foundation of all time synchronization protocols is to obtain reference points: the local times of nodes corresponding to a simultaneously observed event. The natural choice for wireless sensor networks is to synchronize by the transmission of a radio message. This can be accomplished by time stamping the message at both the sender and the receiver sides. Where and how this is done affects the accuracy.

The well-known RBS approach [5] time-stamps messages only on the receiver side; therefore, it effectively eliminates random delays on the sender side. However, time-stamping the radio messages in the low layers of the radio stack has practically the same effect [6]. The TPSN approach [6] eliminates the *access time*, the delay incurred waiting for the availability of the channel, and the *propagation time* by low-level time-stamping and making use of implicit acknowledgments to transmit information back to the sender. This protocol gains additional accuracy over RBS due to time-stamping two radio messages and averaging these time-stamps. A disadvantage of the TPSN protocol is that the two-way communication prohibits the use of message broadcasting, which results in higher communication overhead.

The accuracy of the RBS time-stamping reported by the authors is ~11μs. Least squares linear regression is used to account for clock drift which results in 7.4μs average error between two motes after a 60-second interval. Multi-hop time synchronization for RBS is achieved by transferring the local time through intermediary nodes. However, in their experiment the function of the Berkeley motes was limited to providing wireless communication to PDAs (iPAQs). The authors of the TPSN algorithm implemented both TPSN and RBS on the Mica platform using a 4 MHz clock for time-stamping, and compared the precision of the two algorithms. The resulting average errors for a single hop case for two nodes are 16.9μs and 29.1μs for the TPSN and RBS algorithms, respectively [6]. No data on the performance of TPSN for the multi-hop case is available.

There are other factors, not addressed by either the RBS or TPSN protocol, that introduce random and deterministic delays in the process of message transmission. The *encoding time* is the delay it takes for the radio chip on the sender side to encode and transform a part of the message to electromagnetic waves. The analogue of this on the receiver side is the *decoding time*. The *interrupt handling time* is the delay between the radio chip raising and the microcontroller responding to an interrupt that signals the reception or transmission of a part of the message. These delays are mostly deterministic with less than 10μs variance on the Berkeley motes, but nevertheless they introduce time-stamping errors an order of magnitude larger than the propagation time.

The Flooding Time Synchronization Protocol (FTSP) [15] improves the time-stamping precision of both RBS and TPSN by time-stamping a broadcasted message multiple times, both on the sender and receiver sides. These time-stamps are made when individual bytes of the message are sent or received and then combined into a single time-stamp to reduce the uncertainties of the encoding/decoding and interrupt handling times. This final error corrected value is then embedded into the same message which is being time-stamped before the end of the transmission. Note that the FTSP time-stamping has less communication overhead than that of both RBS and TPSN, as it can synchronize multiple receivers with a single broadcasted message. The accuracy of the FTSP time-stamping is 1.4μs on the Mica2 platform.

The described pair-wise clock synchronization approach is utilized to maintain global time in a multi-hop network in the following way. The FTSP makes all nodes synchronize their clocks to the clock of a selected node, called the root. An election mechanism ensures that there is exactly one root in the network. A dynamic time synchronization hierarchy rooted at this node is created where the root is at Level 0; nodes in the broadcast range of the root are at Level 1 and so on. Every node will estimate the global time by synchronizing its clock to nodes one level higher than itself.

Each node first enters an initialization phase by listening to the radio and waiting for a time sync message. When the message arrives the node updates its linear regression table with the new data point, recalculates its clock skew and offset estimates relative to the global time, and broadcasts a time synch radio message. A node adds a data point to its regression table only if it belongs to a new time sync message as determined by the sequence number of the message that the root increments at each period.

The FTSP maintains the broadcast hierarchy even in the presence of hardware failures, dynamic position changes and partitioning of

the network as detailed in [15]. The performance of the protocol was evaluated in a 60-mote 11-hop network using a 4-hour long experiment. The average time synchronization error stayed below 17.2μs (or 1.6μs per hop). The maximum time synchronization error was below 67μs, which occurred only when multiple nodes were turned off and on to test the robustness of the protocol. Switching off nodes and introducing new ones (other than the root) did not affect the time synchronization error of other nodes.

3.2 Message routing

The routing service in PinPtr faces application-specific demands. In order for the system to be accurate and responsive, data packets containing TOA measurements must be routed to a single destination node with maximum delivery ratio within the first second. This soft real-time constraint is necessary to meet the requirement of 2 seconds latency for the overall system. The measurements originate from the same area around the shooter and the trajectory of the bullet, and are transmitted approximately the same time, producing a burst-like load in the network.

To allow the study a family of protocols, the Directed Flood Routing Framework was created [16], which enables the use of different routing policies. We developed a fast, gradient-based, 'best-effort' converge-cast protocol with built-in data aggregation [16] for PinPtr.

Convergecast policies are used to route data packets from all nodes of the network to a selected node, called the root. Intermediate nodes rebroadcast a data packet zero, one or more times until it is received from a node "closer" to the root than the current node. In the gradient convergecast policy, being closer means that the hop-count distance from the root is smaller. The same data packet can reach the root through several different paths, always descending in the gradient field. This guarantees robustness and fast message delivery at the expense of higher communication overhead.

Each node retransmits a data packet up to three times. The delay between the first and second transmissions is relatively long but it leaves the nodes receiving the first transmission enough time and radio channel bandwidth to retransmit the packet. The policy remembers each data packet for a certain time period since the last time it was received from a node further from the root.

Clearly, this policy does not guarantee message delivery, but is best effort only. This is not a serious limitation for PinPtr because of the availability of multiple sensor readings. The gradient convergecast policy yields a very fast and robust routing protocol to deliver messages to a root node, but at the expense of significant message overhead. Depending on the topology of the network, the number of transmissions during the routing of a single data packet can grow as the square of the distance between the sender and the root.

3.3 Sensor localization

The self-localization algorithm is based on acoustic range estimation between pairs of nodes. This design decision was primarily based on the hardware availability. While there are many acoustic localization schemes proposed in the literature [18], [8], there were only two available on the mote platform at the time. The accuracy and range of the first one utilizing a hardware tone detector was not satisfactory for our requirements [22]. The second approach – our own – improved the accuracy and the effective range significantly, while it utilized well-known

techniques for the localization procedure [19]. The approach is outlined below:

The standard sensor board of the MICA2 mote is equipped with a sounder and a microphone. We did not use our own acoustic board because the sensitivity of the microphones on it is very low in order to be able to handle muzzle blasts and shockwaves.

The ranging procedure starts by the source node broadcasting a radio message and emitting multiple chirps. The destination node samples each of the chirps by streaming the microphone and adds these samples together to increase the signal to noise ratio. Once the recording is done, a digital band-pass filter and a peak detector are used to determine the start of the first chirp. Finally, the range is computed using time of flight of the chirp, since the radio propagation delay is negligible in this case. Special attention was paid to the implementation of the digital signal processing on the severely resource constrained nodes [19].

This approach increases the range of acoustic distance measurement almost five-fold to 9 meters and the accuracy by over an order of magnitude to 10 centimeters over existing methods on the same hardware [22]. This attained accuracy was found to be independent of the actual range.

The self-localization procedure utilizes a time slot negotiation algorithm to schedule acoustic ranging measurements of a large number of nodes. The procedure assigns all nodes unique time slots within the radius of two radio hops, which is a safe upper bound on the acoustic range in practical cases. In the case when a node has more two-hop neighbors than the total number of available time slots, then some of the neighbors do not get time slots assigned. In each time slot the appropriate nodes initiate the acoustic ranging procedure with all of their neighbors at once. The measurement results are propagated back to the base, which performs an optimization procedure, iteratively placing the nodes relative to the known anchor points while performing a least squares minimization.

In our experimental setup in a 30x15 meter area covered by 50 nodes including four anchor points in moderate urban noise the average error of the self-localization procedure was 11 centimeters, while the largest experienced error was 25 centimeters. While this performance satisfies the countersniper application requirements, there are significant limitations that make this approach less than ideal. The requirement of all nodes having 4 neighbors within the 10-meter range to get unambiguous 3D location is not practical. Also the sounder makes the sensor board larger and consumes extra power. The audible frequency of the sounder makes the nodes easier to detect by the adversary. On the other hand, ultrasonic sounders have even more limited range. Due to these limitations all PinPtr tests so far have been performed using hand placed nodes on surveyed locations.

In order to address these limitations we are exploring a new approach we call passive acoustic sensor localization. The technique is based on using external acoustic sources. For the countersniper application, the straightforward choice is weapons fire. The task is then to solve the inverse problem; instead of estimating the unknown shooter position using sensors at known positions, estimate the sensor positions using shots.

In the most general case, it is possible to estimate the sensor positions using shots from known positions taken at unknown times. There exists an analytical solution utilizing linearization that needs six shots from different known positions to determine

the locations of four sensors. However, all four sensors need to be in direct line of sight to all the shots. Furthermore, the solution is very sensitive to even small individual measurement errors. A non-analytical alternative solution based on a heuristic search needs only four shots to estimate the location of the four sensors. However, in its current implementation the procedure is too slow. At this point, none of these approaches seem practical.

At the other end of the spectrum of passive acoustic localization techniques is the simplest technique of producing external acoustic events at known positions and known times. This is in effect the same problem as active acoustic localization outlined above. If the shots are taken right next to the motes with known positions (i.e. anchor points) then the given mote will detect the sound first and it can provide the starting time for the time-of-flight measurements for the other nodes. Of course, the range is not limited to 10 meters, but can be as large as hundreds of meters. However, multipath effects are more problematic in this case. Passive acoustic localization is an ongoing research effort.

Note that sensor localization is not a one-time process. After deployment full sensor localization needs to be performed. But during the lifetime of the system, additional sensor nodes may be deployed to replace failed ones. The sensors may also get relocated either accidentally or intentionally by the adversary. This can be detected by an on-board accelerometer if it is available. Alternatively, the sensor fusion algorithm may keep track of how individual sensors contributed to the position estimation of shooters. If the results for several consecutive shots from different positions do not agree with a given sensors data, it can be flagged as a possible misplaced sensor. In this case, four shots can be used to recompute the suspected sensor's position. Again, care must be taken to avoid using non line-of-sight shots.

4. SIGNAL DETECTION

The acoustic sensor board designed for the countersniper system has three independent acoustic channels. The three microphones are located exactly 2 inches from each other as shown in Figure 3. Our original plan was to detect the Angle of Arrival (AOA) using Time Difference of Arrival (TDOA) of both the muzzle blast and the shockwave on every node. Each channel can be sampled at 1 MHz providing microsecond time resolution needed for the accuracy in the one degree range. Early experiments showed that this can be achieved using the board. However, in order to obtain an absolute angle measurement, the exact orientation of the board itself needs to be known at the same accuracy. This can be done using a precision magnetometer (assuming again horizontal placement), but requires extra hardware and adds unnecessary complexity to the system. Instead we decided to use a single channel TOA measurement and to let the base station fuse the data from multiple sensors.

The signal processing algorithms are implemented on a Xilinx Spartan II FPGA. The incoming raw acoustic signal sampled at 1 MHz is compressed using zero-crossing (ZC) coding, a widely used technique in speech recognition [13]. An interval between zero-crossings is coded by storing the start time of the interval (T), length of the interval (L), the minimum or maximum signal value (Mm), the previous signal average amplitude (P) and the rise time (τ) of the signal, as shown in Figure 4. These features are used to detect possible occurrences of shockwave and muzzle blast patterns in the coded signal stream.

Both shockwave and muzzle blast events are individually modeled by state machines. The states include IDLE, POSSIBLE_START,
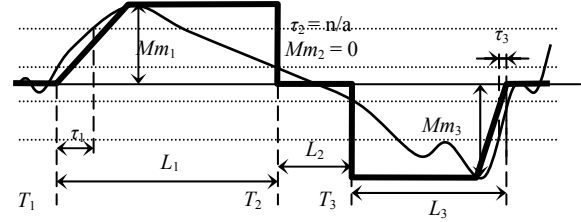

Figure 4. Zero crossing coding of the audio signal. A thin solid line shows the original signal, dashed lines are comparison levels, and a thick solid line represents the coded signal. In addition to the starting time (T), amplitude (Mm), length (L), and the rise time (τ) shown on the plot, the ZC code also contains the previous average amplitude (P) values

DETECTED, and several intermediate states representing the (warped) evaluation of time. The transitions are guarded by Boolean expressions using combinations of ZC properties. The states machines are traversed as ZC codes arrive. If the DETECTED state is reached, the start of interval (T) data stored at the POSSIBLE_START state is returned as the TOA of the corresponding detected event. The state machines were optimized using an extensive acoustic library of shots, converted to the ZC domain.

The TOA of the detected acoustic event using the on-board clock is stored and the mote is notified. The Mica mote reads the measurement data (TOA and optionally signal characteristics) and also performs time synchronization between its own clock and that of the acoustic board. The measurement data is then propagated back to the base station using middleware services of the sensor network.

The signal detection algorithm proved to be quite robust. It recognized 100% of the training events and more than 90% of the other recorded shot events. (Note that a shot may be detected by some sensors and may not be recognized by others, depending on the location of the sensor.) It was quite difficult to produce any acoustic event other than shots without hitting the microphone to produce false positives. During the tests a few sensors proved to be sensitive to wind, causing some false positives, but it turned out to be the problem of loose microphone sealing.

5. SENSOR FUSION

There are a multitude of techniques for locating a transmitting source by an array of listening devices. Near-field beam forming methods are successfully used to detect multiple sources in noisy reverberant areas [2], [3]. However, the most sophisticated methods require the transmission of data records between nodes and/or the base station, e.g. [1]. Our sensor network does not have the necessary communication bandwidth to support this alternative. There exist similar two-step techniques where in the first step the TDOA data is calculated (or alternatively, measured), and in the second step the location is calculated, [2], [12], [14]. The communication burden of transmitting measured TOA data is acceptable.

Since a pair of sensor readings defines a hyperboloid surface in space, in theory four appropriate measurements are enough to identify a 3D location, provided the speed of sound is known. Unfortunately, errors in detection, sensor localization, and time synchronization all affect the accuracy of the solution. Using more

measurements and solving the over-determined equations help to overcome this problem [3], [12], [14].

Conventional methods (e.g. ones using LS or maximum likelihood criteria) work fine with noisy or even reverberant data, but in many cases sensors not having direct line of sight detect echoes *only*, resulting in large errors in localization. In our experiments in urban terrains typically 10-50% of the sensor readings provide erroneous data. Unfortunately, published localization methods do not address the problem of incorrect (TOA or TDOA) measurements. Simply applying the analytical solution or any other solution technique using the whole data set possibly containing a large number of incorrect measurements is not an option when high accuracy is required. Searching for the maximal set of consistent measurement data by repeatedly applying the solver on different sets of input data is not straightforward but a possible solution; no computationally efficient way to do it is known so far.

The proposed solution utilizes time of arrival data of the measured shockwaves and muzzle blasts. From the measurements and the sensor positions a four-dimensional *consistency function* is defined. A quick search algorithm finds the maximum of this function. The location corresponding to the maximum is the shooter position estimate. The consistency function is defined such a way that it automatically classifies and eliminates erroneous measurements and multipath effects. Another beneficial property of the consistency function is that multiple shots appear as multiple local maxima.

5.1 Consistency function

Let N be the number of TOA muzzle blast measurements, and for each $i = 1, \text{K}, N$ let (x_i, y_i, z_i) be the coordinates of the sensor making the ith measurement and t_i the time of arrival of the detected muzzle blast. We cannot assume that the sensors make only one measurement per shot, because some sensors can detect both a direct line of sight and a delayed echo. Neither can we assume that the N measurements correspond to a single shot, as several shots can be fired in a few seconds during urban combat. To find the position of the shooter(s), first we define a *consistency function* on the four-dimensional space-time space and search for its local maxima that correspond to the location and time of possible shots. Then these maxima are further analyzed to eliminate false positives caused by consistent echoes.

For any hypothetical shooter position (x, y, z) and shot time t the theoretical time of arrival of the muzzle blast at the sensor that recorded the ith measurement is

$$t_i(x, y, z, t) = t + \frac{\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}}{v},$$

where v is the speed of sound. If the ith measurement is a direct line of sight detection of this hypothetical shot, then in theory the times $t_i(x, y, z, t)$ and t_i must be equal. In practice however, due to errors in sensor localization, time synchronization and signal detection, only the following inequality is satisfied

$$| t_i(x, y, z, t) - t_i | \leq \tau, \tag{1}$$

where $\tau = \delta_1 / v + \tau_2 + \tau_3$ is an *uncertainty value*, δ_1 is the maximum localization error, τ_2 is the maximum time synchronization error, and τ_3 is the maximum allowed signal detection uncertainty. For practical purposes the localization error dominates τ. We assume that an upper bound for τ is known based on an a priori evaluation of the self localization, time synchronization and signal detection algorithms. The consistency function $C_\tau(x, y, z, t)$ is defined as the number of measurements for which (1) holds:

$$C_\tau(x, y, z, t) = \underset{i=1, \text{K}, N}{\text{count}} \left(\left| t_i(x, y, z, t) - t_i \right| \leq \tau \right).$$

The value of the consistency function for any (x, y, z, t) defines the number of measurements supporting the hypothesis that the shot was taken from (x, y, z) at time t, with uncertainty τ. The consistency function is integer valued and always less than or equal to N. It is additive for the list of TOA measurements, and increasing in τ. Although $C_\tau(x, y, z, t)$ is not continuous, it satisfies the crucial property utilized in the discrete search algorithm (see Section 5.2):

$$C_{\tau/4}(x', y', z', t') \leq C_\tau(x, y, z, t) \tag{2}$$

whenever $|x - x'|/v, |y - y'|/v, |z - z'|/v, |t - t'| \leq \tau/2$.

The consistency function usually takes its maximum value not in a single point but in a 4-dimensional area, called the *max area*. The size of the max area depends on τ. If erroneous measurements are present, it is theoretically possible that there are more unconnected max areas, and it is also possible that the true location of the shot is not contained in the max area. A simple counterexample can easily be generated even for the ideal situation with $\tau = 0$, M correct measurements and $M - 2$ bad measurements, where the maximum consistency value becomes $M + 1$, and the optimum is not at the true location. In practical situations, multipath effects can create strong local maxima (mirror effect), but based on empirical evaluation the uncertainty value τ must be higher for mirror images to reach the same $C_\tau(x, y, z, t)$ value.

The counter-sniper system utilizes the maximum of the consistency function as the location (and time) estimate of the shot. Since gradient-type search methods do not guarantee global convergence on a surface with multiple local maxima, an exhaustive-like search method is utilized. The fast search algorithm finds the global maximum by searching the relevant sections of the space-time only, iteratively zooming to the global maximum.

5.2 Search Algorithm

The time complexity of finding the maxima of the consistency function in the guarded area $[X_{\min}, X_{\max}] \times [Y_{\min}, Y_{\max}] \times [Z_{\min}, Z_{\max}]$ and in the appropriate time window $[T_{\min}, T_{\max}]$ is linear in terms of $X_{\max} - X_{\min}$, $Y_{\max} - Y_{\min}$, $Z_{\max} - Z_{\min}$, $T_{\max} - T_{\min}$ and N, because by (2) it is enough to evaluate $C_{\tau/4}(x', y', z', t')$ at grid points of the search space with uniform distance $v\tau/2$ for the x, y and z coordinates and $\tau/2$ for

t, and then finding the maxima among these points. However, the number of computation steps quickly becomes astronomical in practice, exceeding 10^{12}, rendering this simple algorithm not viable.

There is an extensive literature on several well known algorithms for finding the local and global maxima of nonlinear functions, such as the Newton, Levenberg-Marquardt and Generalized Bisection methods. Since the consistency function is not continuous and we are interested in finding its global maxima, we applied the Generalized Bisection method based on interval arithmetic [23]. Interval arithmetic introduces algebraic operations on closed intervals that represent possible values of variables. Every algebraic expression, including our definition of the consistency function, can be evaluated for intervals. For intervals $[x_{min}, x_{max}]$, $[y_{min}, y_{max}]$, $[z_{min}, z_{max}]$ and $[t_{min}, t_{max}]$ the consistency function yields the interval

$$[C_\tau^{min}, C_\tau^{max}] =$$
$$C_\tau([x_{min}, x_{max}], [y_{min}, y_{max}], [z_{min}, z_{max}], [t_{min}, t_{max}])$$

that have the property that for every $x_{min} \leq x \leq x_{max}$, $y_{min} \leq y \leq y_{max}$, $z_{min} \leq z \leq z_{max}$ and $t_{min} \leq t \leq t_{max}$

$$C_\tau^{min} \leq C_\tau(x, y, z, t) \leq C_\tau^{max}.$$

The value C_τ^{min} is the number of measurements that satisfy (1) for some point of the 4-dimensional rectangular region determined by $[x_{min}, x_{max}] \times \Lambda \times [t_{min}, t_{max}]$, while C_τ^{max} is the number of measurements that satisfy (1) for all points of the same region.

During the search we maintain a list of 4-dimensional rectangular regions ('boxes'), initially containing only $[X_{min}, X_{max}] \times [Y_{min}, Y_{max}] \times [Z_{min}, Z_{max}] \times [T_{min}, T_{max}]$, together with their evaluation under the consistency function. At each step we remove the region that has maximum C_τ^{max} value from the list, bisect it into two equal parts along its longest dimension, and insert the two resulting regions back to the list. We stop this procedure when the size of the maximum region is less than $v\tau/2$ for the space and $\tau/2$ for the time coordinate. The resulting 4-dimensional region is guaranteed to contain at least one global maximum point of the consistency function $C_\tau(x, y, z, t)$. Note that there may be several boxes with the same C_τ^{max} value, usually covering a small area around the true location. When displayed, this area provides an easily understandable visual representation of the uncertainty region of the location estimate.

The consistency function may have several local minima, resulting from echoes or multiple shots. The iterative application of the above search method with appropriate echo detection can provide a powerful tool for simultaneous shooter localization. This is an ongoing research area with encouraging preliminary results.

It is very difficult to asses the time complexity of this search algorithm based on the Generalized Bisection method in a complex urban environment. Nevertheless, this algorithm is guaranteed to be faster than the simple linear algorithm, which requires

$$N(X_{max} - X_{min})(Y_{max} - Y_{min})(Z_{max} - Z_{min})(T_{max} - T_{min})/(v^3\tau^4)$$

computational steps. For a 200x100x20 meter urban area with a 2-second time window and uncertainty value of 0.3 milliseconds and 30 TOA measurements, this translates to $8 \cdot 10^{13}$ steps. However, the Generalized Bisection method always required less than 10^5 steps during our experimental verification.

Currently, the shockwave measurements are used to estimate the trajectory of the projectile but not in the localization algorithm. The trajectory estimation utilizes an approach similar to the localization. Further research is required on how to incorporate the shockwave measurements into the location search algorithm. Clearly, a precise ballistic model and additional search space dimensions seem inevitable.

6. RESULTS

The performance of the system was tested in a series of field trials in the McKenna MOUT training facility in Ft. Benning, GA. The measurement results used in this section were gathered in July, 2003. The system used FTSP for time synchronization and the gradient convergecast algorithm for message routing. The motes, however, were hand placed at surveyed points, as the range of the self localization technique proved to be inadequate. The setup utilized 56 motes deployed in the central area of the McKenna village as shown in Figure 5, a screen dump of the system graphical user interface that includes an overhead picture of the MOUT site. The estimated position of the shooter is shown by the large circle, while the direction of the shot is indicated by an arrow. Other circles indicate the sensor positions where medium sized ones denote sensor locations whose data were utilized in the current location estimation.

For error analysis purposes, 20 different known shooter positions were used in the experiment. During the test, 171 shots were fired, 101 of which were blanks and 70 were short range training ammunitions (SRTAs). Since the performance of the system was similar for both types of ammunition, only the unified results are presented.

Figure 5. 2D System Display

The shooter localization error of the system is shown in Figure 6, where the 3D error is the total localization error, while in the 2D error the elevation information is omitted. The system accuracy is remarkably good in 2D. The average 2D error was 0.6m, 83% of shots had less than one meter, and 98% had less than 2 meters of error.
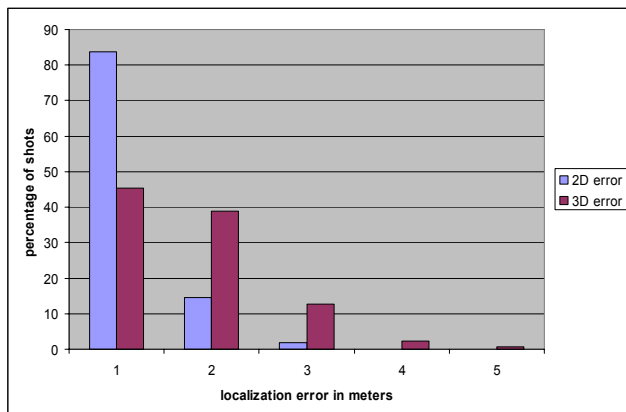

Figure 6. Histogram of localization accuracy in 3D and 2D

The elevation detection was not as accurate because the sensors were mostly positioned on the ground, approximately in a plane. There were only a few sensors located on rooftops or window ledges. This lack of variation in sensor node elevation resulted in the 3D accuracy being worse than the 2D accuracy. It is expected that this could be significantly improved by locating a larger fraction of the sensor nodes in elevated positions. As Figure 6 shows, 46% of the shots had less than 1m, and 84% of shots had less than 2m position error in 3D. The average 3D error was 1.3m.

6.1 Error sources

The sensor fusion algorithm uses TOA measurements recorded by different sensors at different locations. Hence, two potential sources of measurement error are imperfect time synchronization and inaccurate sensor locations. The data gathered at the field trials enabled us to experiment with the effect these have on the overall system accuracy. The effects of time synchronization error are summarized in Figure 7. For each simulated time synchronization error value of T, the detection time for each sensor was modified by t where $-T/2 < t < T/2$ using uniform random distribution. Then the sensor fusion algorithm estimated the shooter position. Each shot was used ten times; therefore, each data point in the diagram represents 1710 experiments.

The results in Figure 7 clearly show that the time synchronization accuracy of FTSP is much better than what is needed by this

application. The added 3D localization error of 10cm in the presence of 0.5ms time synchronization error is insignificant. On the other hand, for future multiple shot detection and echo discrimination, well synchronized measurements are advantageous.
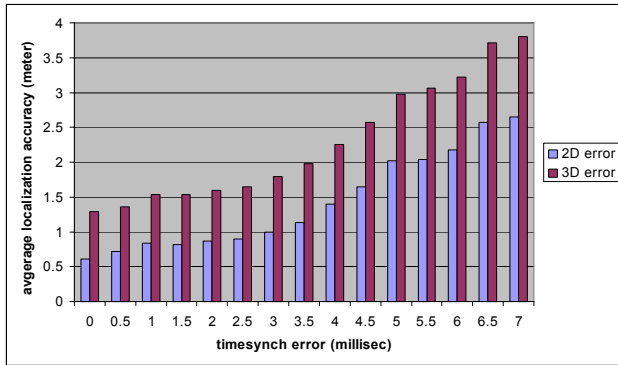

Figure 7. Localization accuracy vs. time synch error

The effects of sensor location errors are similar in that a time synchronization error of 1 millisecond translates to 1 foot of sensor location error using the speed of sound. In fact, it is in the worst case only, since the position error vector is usually not parallel with the shooter-sensor line. A uniform distribution of time synchronization error is a different distribution of sensor location error. Nevertheless, we performed similar experiments for sensor location error that indicated very similar results. For example, 3msec time synch error resulted in 1.79m average shooter localization error, while the same value for 1m sensor location error was 1.94m.

6.2 Sensor Density

We have also analyzed the effects of sensor density. Again, we used the real data gathered on the field and then removed sensors randomly. The results are shown in Figure 8 and Figure 9. For each N, where N is the number of sensors and $N \leq 56$, we generated a random selection of the 56 available nodes, ran the sensor fusion for all 171 shots and repeated the procedure ten times. Since N was decreased by two at a time and we stopped at 8 nodes, we tested 250 different sensor network configurations.
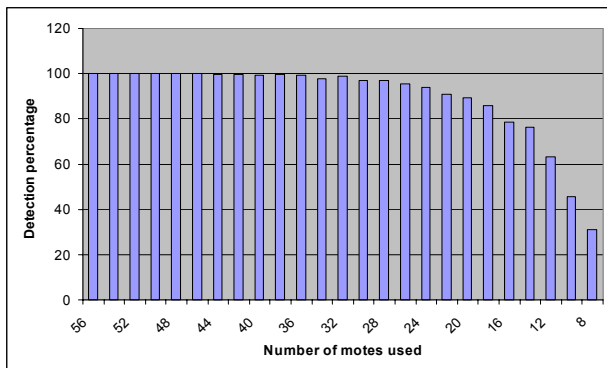

Figure 8. Detection rate vs. number of sensors used

We consider a shot undetected if there are less than six sensors detecting a muzzle blast. As the numbers of sensor decreased, so did the number of successfully detected shots as shown in Figure 8. Hence, the data in Figure 9 only uses the successfully

detected shots. The diagram indicates that the error has an exponential characteristic. Close to our original setup, the error hardly increases. At 36 nodes the average 3D error is still less than 2 meters. Beyond this point, however, the accuracy starts to rapidly decrease.
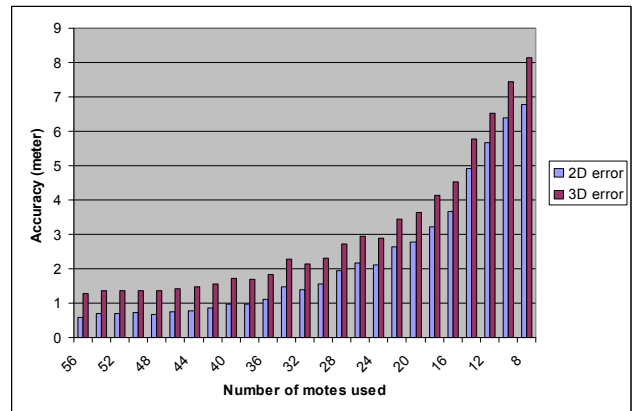

Figure 9. Localization accuracy vs. number of sensor used

The raw results could lead to a premature conclusion that we could decrease the node density by 40% and still get very good accuracy. However, there are other considerations. Node failures decrease sensor density over time, so the planned deployment length needs to be considered. It is not enough to measure the acoustic events; the data also needs to be propagated back to the base station. There must be enough nodes to ensure a connected network with redundancy for robustness and good response time.

6.3 Sensor Fusion

The overall accuracy of PinPtr during the field tests in an urban environment indicates its tolerance to multipath effects. Of the 171 shots used in the analysis above, the average rate of bad measurements, i.e. TOA data that were not consistent with the final shooter location estimate, was 24%. In our experience, the vast majority of erroneous TOA data were due to multipath.

It is possible to solve the TDOA-based localization problem analytically, e.g. as in [14], where the constraints from measurements are converted to a linear equation system. This solution requires five measurements to determine the 3D position of a source, but it is straightforward to extend the solution in [14] for more sensor readings. The solution of the over-determined equation system provides a least-squares estimation of the shooter location. We used this approach to evaluate our sensor fusion technique.

To compare the accuracy of the fusion algorithm to that of the analytical solution, field sensor measurements of 46 shots with known positions were used as test cases. In the first test all bad measurements resulting from multipath effects or sensor failure were removed from the data set. Each of the remaining set of good measurements was consistent with the known shooter position; the time error was less than 0.5ms for each sensor reading. The shooter positions were estimated using both methods. The accuracies of the two solutions were very close to each other, as the histogram of errors shows in Figure 10. The mean 3D localization error for the fusion algorithm and the analytical solution were 1.2m and 1.3m, respectively, for the 46-shot test set. The difference is much less than the sensor and the reference shooter position measurement errors, thus the

performance of the two solutions can be considered to be equally good in this test scenario.
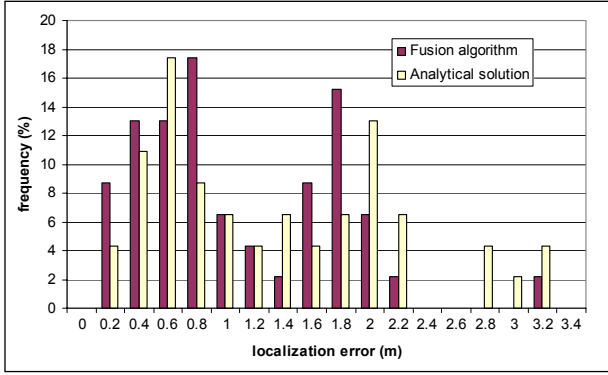

Figure 10. Histogram of the localization errors using the fusion algorithm and the analytical solution.

In the previous test the input contained only correct measurements. In practical cases, however, inconsistent measurements are present primarily due to multipath effects, even after careful pre-filtering of the sensor readings. To illustrate the sensitivity of the methods to measurement errors, bad sensor readings were added back to the input data set from the previously removed bad data set. For each shot, 2^B test sets were generated by combining the good measurement set with all possible combinations of the bad sensor readings containing B measurements. The number of good and bad sensor readings varied between 8 and 29, and between 1 and 10, respectively. Using all the 46 shots 325 experiments were generated as test cases. Figure 11 shows the performances of the two methods, as a function of the ratio of the bad and good measurements.
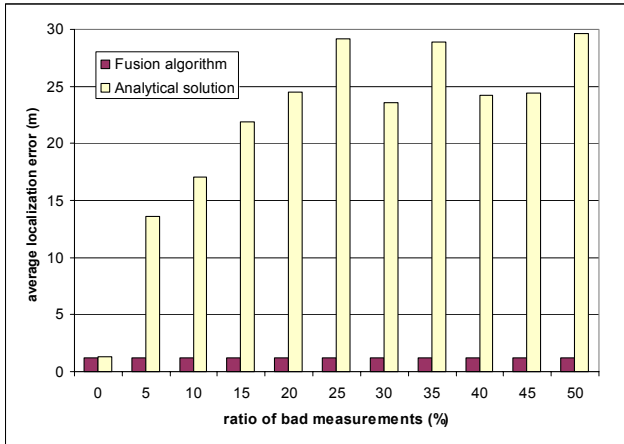

Figure 11 The average localization error vs. the ratio of bad and good measurements

It is clearly visible that the precision of the analytical solution was severely degraded when bad measurements were present. Our fusion algorithm, however, was able to successfully eliminate the bad measurements, and its performance was the same as in the first test, independently of the ratio of the bad and good measurements.

6.4 Time synchronization

PinPtr should be able to operate over several weeks or even months. It is not required to be continuously active, and should be powered down most of the time to save energy. The question arising naturally is whether continuous time synchronization is really necessary.

As it is pointed out in [5], post facto synchronization is enough in many cases, no continuous synchronization is required. Systems collecting data or reacting to rare events, but requiring exact time measurements belong to this class of applications. A post facto synchronization approach described in [5] utilizes explicit pair-wise synchronization after message passing. We propose an alternative method embedded into the message routing protocol which does not require any additional message exchange apart from the routing messages. The proposed solution requires precise message time-stamping on both the transmitter and receiver, e.g. the method described in [15].

The basic problem is the following: a sensor detects an event and time stamps it using its local clock. However, the target node needs to know the time of the event in its own local time. The sensor and the target nodes may be several hops apart from each other. Still, it is possible to solve the problem without any explicit time synchronization in the network. An implicit synchronization may be performed during the routing process.
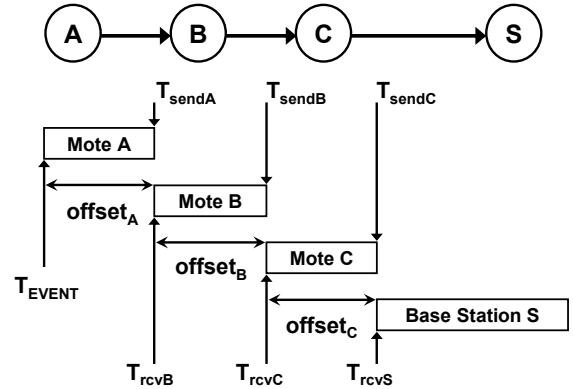

Figure 12. Estimation of detection time T_{EVENT} can be iteratively determined along a routing path A, B, C, S.

Along with the sensor reading, a radio message includes an *age* field, which contains the elapsed time since the occurrence of the event. This additional information adds only a very small overhead to the message. Each intermediate mote measures the offset, which is the elapsed time from the reception of a sensor reading till its retransmission. The *age* field is updated upon transmission using a precise time stamping method described in [15]. When the sensor reading arrives at the destination, the *age* field contains the sum of the offsets measured by each of the motes along the path. The destination node can determine the time of the event by subtracting *age* from the time of arrival of the message. The concept is illustrated in Figure 12. An event is detected at node A at time instant T_{EVENT}, then a notification message is sent to destination node S through nodes B and C. The message delays at the nodes are *offset$_A$, offset$_B$,* and *offset$_C$,* respectively. The message arrives at S at time instant T_{rcvS} containing an age field of *offset$_A$+offset$_B$+offset$_C$*. The time of the event can be calculated as $T_{EVENT} = T_{rcvS}$- *age*.

One possible problem with this approach is that the time measurement units of the intermediate nodes are not of the same length, because of the slight differences in their clock frequencies. Since this method does not compensate for skew errors, significant error can accumulate if the routing of the sensor reading takes a long time. The crystal used in the Mica2 mote has accuracy better than 50ppm, thus the clock skew error is less than 50μs per second. Thus, the worst-case post-facto synchronization error can be estimated as $5*10^{-5}T_R$, where T_R is the worst-case time of the message routing.

This time synchronization algorithm can be further refined by exploiting the usual properties of certain wireless routing protocols. Because of unreliable radio channels the same radio message may be (re)broadcasted several times at intermediate nodes, and it can arrive to the base station multiple times along different paths. Even though these multiple messages hold the same sensor reading, the calculated T_{EVENT} times can vary, mainly caused by the different clock frequencies of the nodes along the different routes. The destination node can use a statistical analysis of the received elapsed times to get a better estimate of the time the event occurred.

The main advantage of the proposed integrated time synchronization and routing algorithm is that it does not require additional radio messages, enables power management, and the overhead imposed on the original routing messages is very low.

7. FUTURE PLANS AND CONCLUSIONS

The performance of the presented sensor network-based countersniper system is on par with existing centralized systems in regular settings. However, PinPtr outperforms other systems in urban environments because of its widely distributed sensing capability mitigating multipath effects. During field trials the 3D localization accuracy of 1.3 meter and latency under 2 seconds were achieved on average. The system shows great promise, however, the hardware needs to be miniaturized and the packaging needs to be made weather-proof and rugged before its potential deployment. We are also working on the following improvements:

The current system does not perform any kind of power management. A new generation sensorboard will have a watchdog circuitry that wakes up the board when an acoustic event of potential interest is detected. Only if the unit detects an actual shot will it wake up the mote. A technical challenge yet to be overcome is how the messages will be routed to the base station in the presence of sleeping motes, i.e. ones that did not detect the shot. Using the radio to wake up motes is currently not supported on the Mica2 platform.

The current sensor fusion implementation does not handle multiple shots. Furthermore, it does not use the shockwave for localization. Both of these capabilities will be added to future versions of the system. Note that to our knowledge there are no existing countersniper systems that can distinguish multiple (almost) simultaneous shots.

Continuous time synchronization uses power and makes the system easier to detect. In the future versions of the system the post facto synchronization proposed in section 6.4. will be used. The estimated accuracy of this method is still within the required 0.5 ms.

Dynamic passive acoustic relocalization using detected shots as beacons can help to improve the initial sensor localization data, and also can help to maintain its consistency when sensors are moved.

Longer-term plans depend on the Concept of Operations (CONOPS) that is currently being developed. One scenario calls for the protection of convoy routes in a city. In this case thousands of sensors are deployed by UAVs and the nodes are expected to remain operational for months. The convoys themselves would carry the mobile base stations. This scenario calls for a flexible and dynamic message routing technique. Power management does not have to be acoustic-based; the base station can wake up the motes in its proximity.

Another potential CONOPS calls for the protection of reconnaissance units. When they come under fire they can quickly deploy a system in the area utilizing a UAV or by simply tossing the nodes around their location. The most challenging aspect of this deployment scenario is the need for rapid sensor localization. Power management is not needed at all.

Both of these scenarios call for disposable nodes because recovery operations are costly and risky because of the potential for ambushes. This means that the individual nodes must be inexpensive.

8. ACKNOWLEDGEMENTS

9. REFERENCES

[1] Chen, J. C., Hudson, R. E. and Yao, K. Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field. *IEEE Trans. Signal Processing*, vol. 50, pp. 1843–1854, Aug. 2002.

[2] Chen, J. C., Yao, K., and Hudson, R. E. Source localization and beamforming. *IEEE Signal Processing Magazine*, vol. 19, no. 2, March 2002, pp. 30-39.

[3] Chen, J., Yip, L., Elson, J., Wang, H., Maniezzo, D., Hudson, R., Yao, K., and Estrin, D. Coherent acoustic array processing and localization on wireless sensor networks. *Proceedings of the IEEE*, vol. 91, Aug. 2003, pp. 1154—1162.

[4] Duckworth et al. Acoustic counter-sniper system. In *Proc. of SPIE International Symposium on Enabling Technologies for Law Enforcement and Security*, 1996

[5] Elson, J., Girod, L., Estrin, D. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, vol. 36, issue SI, 2002

[6] Ganeriwal, S., Kumar, R., Srivastava, M. B. Timing-sync protocol for sensor networks. *Proc. First ACM SenSys*, 2003

[7] Gay, D. et al. The nesC Language: A Holistic Approach to Networked Embedded Systems, In *Proc. of PLDI*, 2003

[8] Harter, A., Hopper, A., Steggles, P., Ward, A., Webster, P. The Anatomy of a Context-Aware Application. In *Proc. 5th ACM MOBICOM Conf.* (Seattle,WA, Aug. 1999).

[9] Hill, J. at al. System Architecture Directions for Networked Sensors, In *Proc. of ASPLOS*, 2000.

[10] Hill, J., Culler, D. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro*, Vol. 22, No. 6, 2002, pp. 12–24.

[11] http://www.army-technology.com/contractors/surveillance/metravib/

[12] Huang, Y., Benesty, J., and Elko, G. W. Passive acoustic source localization for video camera steering. *Proc. IEEE ICASSP*, vol. 2, June 2000, pp. 909–912.

[13] Lupu, E. et al. Speaker Verification Rate Study Using the TESPAR Coding Method. In *Proc. of COST 276 Workshop on Information and Knowledge Management for Integrated Media Communication*, 2002

[14] Mahajan, A. and Walworth, M. 3-D Position Sensing Using the Differences in the Time-of-Flights from a Wave Source to Various Receivers. *IEEE Trans. On Robotics and Automation*, Vol. 17, No. 1, February 2001, pp. 91-94

[15] Maroti, M., Kusy, B., Simon, G., Ledeczi, A. The Flooding Time Synchronization Protocol. In *Proc of The Second ACM Conference on Embedded Networked Sensor Systems (Sensys)*, November 2004.

[16] Maroti, M. The Directed Flood Routing Framework. In *Proc of ACM/IFIP/USENIX 5th International Middleware Conference*, October 2004.

[17] Moroz, S. A. et al. Airborne Deployment of and Recent Improvements to the Viper Counter Sniper System. In *Proc of the IRIS Passive Sensors*, February 1999

[18] Priyantha, N., Chakraborty, A., AND Balakrishnan, H. The Cricket Location-Support System. In *Proc. 6th ACM MOBICOM Conf.* (Boston, MA, Aug. 2000)

[19] Sallai, J., Balogh, G., Maroti, M., Ledeczi, A. *Acoustic Ranging in Resource Constrained Sensor Networks*. Technical Report, ISIS-04-504, February 25, 2004 (available at http://www.isis.vanderbilt.edu/publications.asp)

[20] Stoughton, R. Measurements of Small Caliber Ballistic Shock Waves in Air. *JASA 102 (2)*, Pt. 1, 1997

[21] Vick, A. et al. Aerospace Operations in Urban Environments: Exploring New Concepts. In *RAND MR-1187*, 2000

[22] Whitehouse, K. and Culler, D. Calibration as Parameter Estimation in Sensor Networks. In *Proc of ACM International Workshop on Wireless Sensor Networks and Applications*, 2002

[23] A Review on Interval Computation -- Software and Applications, (with S. Xu, X. Yang), *Int. J. of Computational and Numerical Analysis and Applications*, Vol. 1, No. 2, pp. 149-162, 2002.