

# Shooter Localization and Weapon Classification with Soldier-Wearable Networked Sensors

Peter Volgyesi, Gyorgy Balogh, Andras Nadas, Christopher B. Nash, Akos Ledeczki  
Institute for Software Integrated Systems, Vanderbilt University  
Nashville, TN, USA  
akos.ledeczki@vanderbilt.edu

## ABSTRACT

The paper presents a wireless sensor network-based mobile countersniper system. A sensor node consists of a helmet-mounted microphone array, a COTS MICAz mote for inter-node communication and a custom sensorboard that implements the acoustic detection and Time of Arrival (ToA) estimation algorithms on an FPGA. A 3-axis compass provides self orientation and Bluetooth is used for communication with the soldier's PDA running the data fusion and the user interface. The heterogeneous sensor fusion algorithm can work with data from a single sensor or it can fuse ToA or Angle of Arrival (AoA) observations of muzzle blasts and ballistic shockwaves from multiple sensors. The system estimates the trajectory, the range, the caliber and the weapon type. The paper presents the system design and the results from an independent evaluation at the US Army Aberdeen Test Center. The system performance is characterized by 1-degree trajectory precision and over 95% caliber estimation accuracy for all shots, and close to 100% weapon estimation accuracy for 4 out of 6 guns tested.

## Categories and Subject Descriptors

C.2.4 [Computer-Communications Networks]: Distributed Systems; J.7 [Computers in Other Systems]: Military

**General Terms:** Design, Measurement, Performance

**Keywords:** Sensor networks, data fusion, acoustic source localization, weapon classification, caliber estimation

**Acknowledgments:** The Darpa IpTO ASSIST program has supported this research. We'd like to express our gratitude to Brian A. Weiss, Craig Schlenoff and the US Army Aberdeen Test Center who carried out the independent evaluation of the system. We are also grateful to Miklós Maróti, Gyula Simon, Branislav Kusý, Béla Fehér, Sebestyén Dóra, Ken Pence, Ted Bapty, Jason Scott and the Nashville Police Academy for their contributions. We'd like to thank our shepherd, Dan Siewiorek for his constructive comments.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*MobiSys '07*, June 11-14, 2007, San Juan, Puerto Rico, USA.  
Copyright 2007 ACM 978-1-59593-614-1/07/0006 ...\$5.00.

## 1. INTRODUCTION

The importance of countersniper systems is underscored by the constant stream of news reports coming from the Middle East. In October 2006 CNN reported on a new tactic employed by insurgents. A mobile sniper team moves around busy city streets in a car, positions itself at a good standoff distance from dismounted US military personnel, takes a single well-aimed shot and immediately melts in the city traffic. By the time the soldiers can react, they are gone. A countersniper system that provides almost immediate shooter location to every soldier in the vicinity would provide clear benefits to the warfighters.

Our team introduced PinPtr, the first sensor network-based countersniper system [17, 8] in 2003. The system is based on potentially hundreds of inexpensive sensor nodes deployed in the area of interest forming an ad hoc multihop network. The acoustic sensors measure the Time of Arrival (ToA) of muzzle blasts and ballistic shockwaves, pressure waves induced by the supersonic projectile, send the data to a base station where a sensor fusion algorithm determines the origin of the shot. PinPtr is characterized by high precision: 1m average 3D accuracy for shots originating within or near the sensor network and 1 degree bearing precision for both azimuth and elevation and 10% accuracy in range estimation for longer range shots. The truly unique characteristic of the system is that it works in such reverberant environments as cluttered urban terrain and that it can resolve multiple simultaneous shots at the same time. This capability is due to the widely distributed sensing and the unique sensor fusion approach [8]. The system has been tested several times in US Army MOUT (Military Operations in Urban Terrain) facilities.

The obvious disadvantage of such a system is its static nature. Once the sensors are distributed, they cover a certain area. Depending on the operation, the deployment may be needed for an hour or a month, but eventually the area loses its importance. It is not practical to gather and reuse the sensors, especially under combat conditions. Even if the sensors are cheap, it is still a waste and a logistical problem to provide a continuous stream of sensors as the operations move from place to place. As it is primarily the soldiers that the system protects, a natural extension is to mount the sensors on the soldiers themselves. While there are vehicle-mounted countersniper systems [1] available commercially, we are not aware of a deployed system that protects dismounted soldiers. A helmet-mounted system was developed in the mid 90s by BBN [3], but it was not continued beyond the Darpa program that funded it.

To move from a static sensor network-based solution to a highly mobile one presents significant challenges. The sensor positions and orientation need to be constantly monitored. As soldiers may work in groups of as little as four people, the number of sensors measuring the acoustic phenomena may be an order of magnitude smaller than before. Moreover, the system should be useful to even a single soldier. Finally, additional requirements called for caliber estimation and weapon classification in addition to source localization.

The paper presents the design and evaluation of our soldier-wearable mobile countersniper system. It describes the hardware and software architecture including the custom sensor board equipped with a small microphone array and connected to a COTS MICAz mote [12]. Special emphasis is paid to the sensor fusion technique that estimates the trajectory, range, caliber and weapon type simultaneously. The results and analysis of an independent evaluation of the system at the US Army Aberdeen Test Center are also presented.

## 2. APPROACH

The firing of a typical military rifle, such as the AK47 or M16, produces two distinct acoustic phenomena. The muzzle blast is generated at the muzzle of the gun and travels at the speed of sound. The supersonic projectile generates an acoustic shockwave, a kind of sonic boom. The wavefront has a conical shape, the angle of which depends on the Mach number, the speed of the bullet relative to the speed of sound.

The shockwave has a characteristic shape resembling a capital N. The rise time at both the start and end of the signal is very fast, under 1  $\mu$ sec. The length is determined by the caliber and the miss distance, the distance between the trajectory and the sensor. It is typically a few hundred  $\mu$ sec. Once a trajectory estimate is available, the shockwave length can be used for caliber estimation.

Our system is based on four microphones connected to a sensorboard. The board detects shockwaves and muzzle blasts and measures their ToA. If at least three acoustic channels detect the same event, its AoA is also computed. If both the shockwave and muzzle blast AoA are available, a simple analytical solution gives the shooter location as shown in Section 6. As the microphones are close to each other, typically 2-4", we cannot expect very high precision. Also, this method does not estimate a trajectory. In fact, an infinite number of trajectory-bullet speed pairs satisfy the observations. However, the sensorboards are also connected to COTS MICAz motes and they share their AoA and ToA measurements, as well as their own location and orientation, with each other using a multihop routing service [9]. A hybrid sensor fusion algorithm then estimates the trajectory, the range, the caliber and the weapon type based on all available observations.

The sensorboard is also Bluetooth capable for communication with the soldier's PDA or laptop computer. A wired USB connection is also available. The sensorfusion algorithm and the user interface get their data through one of these channels.

The orientation of the microphone array at the time of detection is provided by a 3-axis digital compass. Currently the system assumes that the soldier's PDA is GPS-capable and it does not provide self localization service itself. However, the accuracy of GPS is a few meters degrading the

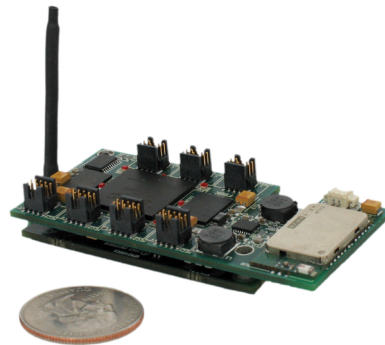


Figure 1: Acoustic sensorboard/mote assembly

overall accuracy of the system. Refer to Section 7 for an analysis. The latest generation sensorboard features a Texas Instruments CC-1000 radio enabling the high-precision radio interferometric self localization approach we have developed separately [7]. However, we leave the integration of the two technologies for future work.

## 3. HARDWARE

Since the first static version of our system in 2003, the sensor nodes have been built upon the UC Berkeley/Crossbow MICA product line [11]. Although rudimentary acoustic signal processing can be done on these microcontroller-based boards, they do not provide the required computational performance for shockwave detection and angle of arrival measurements, where multiple signals from different microphones need to be processed in parallel at a high sampling rate. Our 3rd generation sensorboard is designed to be used with MICAz motes—in fact it has almost the same size as the mote itself (see Figure 1).

The board utilizes a powerful Xilinx XC3S1000 FPGA chip with various standard peripheral IP cores, multiple soft processor cores and custom logic for the acoustic detectors (Figure 2). The onboard Flash (4MB) and PSRAM (8MB) modules allow storing raw samples of several acoustic events, which can be used to build libraries of various acoustic signatures and for refining the detection cores off-line. Also, the external memory blocks can store program code and data used by the soft processor cores on the FPGA.

The board supports four independent analog channels sampled at up to 1 MS/s (million samples per seconds). These channels, featuring an electret microphone (Panasonic WM-64PNT), amplifiers with controllable gain (30-60 dB) and a 12-bit serial ADC (Analog Devices AD7476), reside on separate tiny boards which are connected to the main sensorboard with ribbon cables. This partitioning enables the use of truly different audio channels (eg.: slower sampling frequency, different gain or dynamic range) and also results in less noisy measurements by avoiding long analog signal paths.

The sensor platform offers a rich set of interfaces and can be integrated with existing systems in diverse ways. An RS232 port and a Bluetooth (BlueGiga WT12) wireless link with virtual UART emulation are directly available on the board and provide simple means to connect the sensor to PCs and PDAs. The mote interface consists of an I<sup>2</sup>C bus along with an interrupt and GPIO line (the latter one is used

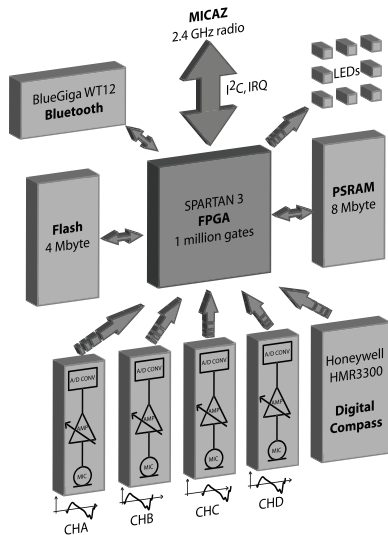


Figure 2: Block diagram of the sensorboard.

for precise time synchronization between the board and the mote). The motes are equipped with IEEE 802.15.4 compliant radio transceivers and support ad-hoc wireless networking among the nodes and to/from the base station. The sensorboard also supports full-speed USB transfers (with custom USB dongles) for uploading recorded audio samples to the PC. The on-board JTAG chain—directly accessible through a dedicated connector—contains the FPGA part and configuration memory and provides in-system programming and debugging facilities.

The integrated Honeywell HMR3300 digital compass module provides heading, pitch and roll information with  $1^\circ$  accuracy, which is essential for calculating and combining directional estimates of the detected events.

Due to the complex voltage requirements of the FPGA, the power supply circuitry is implemented on the sensorboard and provides power both locally and to the mote. We used a quad pack of rechargeable AA batteries as the power source (although any other configuration is viable that meets the voltage requirements). The FPGA core (1.2 V) and I/O (3.3 V) voltages are generated by a highly efficient buck switching regulator. The FPGA configuration (2.5 V) and a separate 3.3 V power net are fed by low current LDOs, the latter one is used to provide independent power to the mote and to the Bluetooth radio. The regulators—except the last one—can be turned on/off from the mote or through the Bluetooth radio (via GPIO lines) to save power.

The first prototype of our system employed 10 sensor nodes. Some of these nodes were mounted on military kevlar helmets with the microphones directly attached to the surface at about 20 cm separation as shown in Figure 3(a). The rest of the nodes were mounted in plastic enclosures (Figure 3(b)) with the microphones placed near the corners of the boxes to form approximately 5 cm $\times$ 10 cm rectangles.

## 4. SOFTWARE ARCHITECTURE

The sensor application relies on three subsystems exploiting three different computing paradigms as they are shown in Figure 4. Although each of these execution models suit their domain specific tasks extremely well, this diversity



Figure 3: Sensor prototypes mounted on a kevlar helmet (a) and in a plastic box on a tripod (b).

presents a challenge for software development and system integration. The sensor fusion and user interface subsystem is running on PDAs and were implemented in Java. The sensing and signal processing tasks are executed by an FPGA, which also acts as a bridge between various wired and wireless communication channels. The ad-hoc internode communication, time synchronization and data sharing are the responsibilities of a microcontroller based radio module. Similarly, the application employs a wide variety of communication protocols such as Bluetooth<sup>®</sup> and IEEE 802.14.5 wireless links, as well as optional UARTs, I<sup>2</sup>C and/or USB buses.

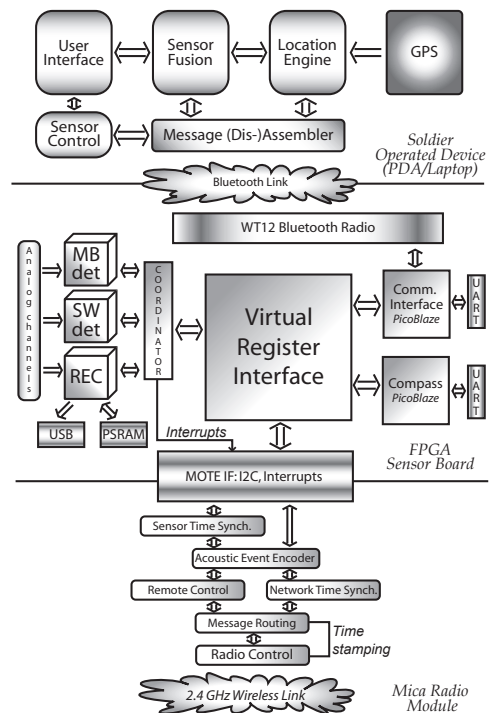


Figure 4: Software architecture diagram.

The sensor fusion module receives and unpacks raw measurements (time stamps and feature vectors) from the sensorboard through the Bluetooth<sup>®</sup> link. Also, it fine tunes the execution of the signal processing cores by setting parameters through the same link. Note that measurements from other nodes along with their location and orientation information also arrive from the sensorboard which acts as a gateway between the PDA and the sensor network. The handheld device obtains its own GPS location data and di-

rectly receives orientation information through the sensor-board. The results of the sensor fusion are displayed on the PDA screen with low latency. Since, the application is implemented in pure Java, it is portable across different PDA platforms.

The border between software and hardware is considerably blurred on the sensor board. The IP cores—implemented in hardware description languages (HDL) on the reconfigurable FPGA fabric—closely resemble hardware building blocks. However, some of them—most notably the soft processor cores—execute true software programs. The primary tasks of the sensor board software are 1) acquiring data samples from the analog channels, 2) processing acoustic data (detection), and 3) providing access to the results and run-time parameters through different interfaces.

As it is shown in Figure 4, a centralized *virtual register file* contains the address decoding logic, the registers for storing parameter values and results and the point to point data buses to and from the peripherals. Thus, it effectively integrates the building blocks within the sensorboard and decouples the various communication interfaces. This architecture enabled us to deploy the same set of sensors in a centralized scenario, where the ad-hoc mote network (using the I<sup>2</sup>C interface) collected and forwarded the results to a base station or to build a decentralized system where the local PDAs execute the sensor fusion on the data obtained through the Bluetooth<sup>®</sup> interface (and optionally from other sensors through the mote interface). The same set of registers are also accessible through a UART link with a terminal emulation program. Also, because the low-level interfaces are hidden by the register file, one can easily add/replace these with new ones (eg.: the first generation of motes supported a standard  $\mu P$  interface bus on the sensor connector, which was dropped in later designs).

The most important results are the time stamps of the detected events. These time stamps and all other timing information (parameters, acoustic event features) are based on a 1 MHz clock and an internal timer on the FPGA. The time conversion and synchronization between the sensor network and the board is done by the mote by periodically requesting the capture of the current timer value through a dedicated GPIO line and reading the captured value from the register file through the I<sup>2</sup>C interface. Based on the the current and previous readings and the corresponding mote local time stamps, the mote can calculate and maintain the scaling factor and offset between the two time domains.

The mote interface is implemented by the I<sup>2</sup>C slave IP core and a thin adaptation layer which provides a data and address bus abstraction on top of it. The maximum effective bandwidth is 100 Kbps through this interface. The FPGA contains several UART cores as well: for communicating with the on-board Bluetooth<sup>®</sup> module, for controlling the digital compass and for providing a wired RS232 link through a dedicated connector. The control, status and data registers of the UART modules are available through the register file. The higher level protocols on these lines are implemented by Xilinx PicoBlaze microcontroller cores [13] and corresponding software programs. One of them provides a command line interface for test and debug purposes, while the other is responsible for parsing compass readings. By default, they are connected to the RS232 port and to the on-board digital compass line respectively, however, they can be rewired to any communication interface by changing

the register file base address in the programs (e.g. the command line interface can be provided through the Bluetooth<sup>®</sup> channel).

Two of the external interfaces are not accessible through the register file: a high speed USB link and the SRAM interface are tied to the recorder block. The USB module implements a simple FIFO with parallel data lines connected to an external FT245R USB device controller. The RAM driver implements data read/write cycles with correct timing and is connected to the on-board pseudo SRAM. These interfaces provide 1 MB/s effective bandwidth for downloading recorded audio samples, for example.

The data acquisition and signal processing paths exhibit clear symmetry: the same set of IP cores are instantiated four times (i.e. the number of acoustic channels) and run independently. The signal paths "meet" only just before the register file. Each of the analog channels is driven by a serial A/D core for providing a 20 MHz serial clock and shifting in 8-bit data samples at 1 MS/s and a digital potentiometer driver for setting the required gain. Each channel has its own shockwave and muzzle blast detector, which are described in Section 5. The detectors fetch run-time parameter values from the register file and store their results there as well. The coordinator core constantly monitors the detection results and generates a mote interrupt promptly upon "full detection" or after a reasonable timeout after "partial detection".

The recorder component is not used in the final deployment, however, it is essential for development purposes for refining parameter values for new types of weapons or for other acoustic sources. This component receives the samples from all channels and stores them in circular buffers in the PSRAM device. If the signal amplitude on one of the channels crosses a predefined threshold, the recorder component suspends the sample collection with a predefined delay and dumps the contents of the buffers through the USB link. The length of these buffers and delays, the sampling rate, the threshold level and the set of recorded channels can be (re)configured run-time through the register file. Note that the core operates independently from the other signal processing modules, therefore, it can be used to validate the detection results off-line.

The FPGA cores are implemented in VHDL, the PicoBlaze programs are written in assembly. The complete configuration occupies 40% of the resources (slices) of the FPGA and the maximum clock speed is 30 MHz, which is safely higher than the speed used with the actual device (20MHz).

The MICAz motes are responsible for distributing measurement data across the network, which drastically improves the localization and classification results at each node. Besides a robust radio (MAC) layer, the motes require two essential middleware services to achieve this goal. The messages need to be propagated in the ad-hoc multihop network using a routing service. We successfully integrated the Directed Flood-Routing Framework (DFRF) [9] in our application. Apart from automatic message aggregation and efficient buffer management, the most unique feature of DFRF is its plug-in architecture, which accepts custom routing policies. Routing policies are state machines that govern how received messages are stored, resent or discarded. Example policies include spanning tree routing, broadcast, geographic routing, etc. Different policies can be used for different messages concurrently, and the application is able to

change the underlying policies at run-time (eg.: because of the changing RF environment or power budget). In fact, we switched several times between a simple but lavish broadcast policy and a more efficient gradient routing on the field.

Correlating ToA measurements requires a common time base and precise time synchronization in the sensor network. The Routing Integrated Time Synchronization (RITS) [15] protocol relies on very accurate MAC-layer time-stamping to embed the cumulative delay that a data message accrued since the time of the detection in the message itself. That is, at every node it measures the time the message spent there and adds this to the number in the time delay slot of the message, right before it leaves the current node. Every receiving node can subtract the delay from its current time to obtain the detection time in its local time reference. The service provides very accurate time conversion (few  $\mu\text{s}$  per hop error), which is more than adequate for this application. Note, that the motes also need to convert the sensorboard time stamps to mote time as it is described earlier.

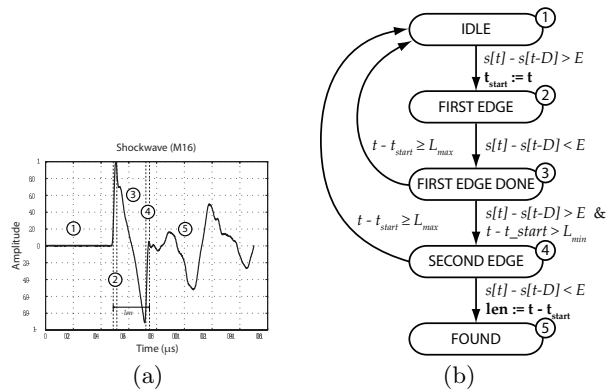
The mote application is implemented in nesC [5] and is running on top of TinyOS [6]. With its 3 KB RAM and 28 KB program space (ROM) requirement, it easily fits on the MICAz motes.

## 5. DETECTION ALGORITHM

There are several characteristics of acoustic shockwaves and muzzle blasts which distinguish their detection and signal processing algorithms from regular audio applications. Both events are transient by their nature and present very intense stimuli to the microphones. This is increasingly problematic with low cost electret microphones—designed for picking up regular speech or music. Although mechanical damping of the microphone membranes can mitigate the problem, this approach is not without side effects. The detection algorithms have to be robust enough to handle severe nonlinear distortion and transitory oscillations. Since the muzzle blast signature closely follows the shockwave signal and because of potential automatic weapon bursts, it is extremely important to settle the audio channels and the detection logic as soon as possible after an event. Also, precise angle of arrival estimation necessitates high sampling frequency (in the MHz range) and accurate event detection. Moreover, the detection logic needs to process multiple channels in parallel (4 channels on our existing hardware).

These requirements dictated simple and robust algorithms both for muzzle blast and shockwave detections. Instead of using mundane energy detectors—which might not be able to distinguish the two different events—the applied detectors strive to find the most important characteristics of the two signals in the time-domain using simple state machine logic. The detectors are implemented as independent IP cores within the FPGA—one pair for each channel. The cores are run-time configurable and provide detection event signals with high precision time stamps and event specific feature vectors. Although the cores are running independently and in parallel, a crude local fusion module integrates them by shutting down those cores which missed their events after a reasonable timeout and by generating a single detection message towards the mote. At this point, the mote can read and forward the detection times and features and is responsible to restart the cores afterwards.

The most conspicuous characteristics of an acoustic shockwave (see Figure 5(a)) are the steep rising edges at the be-

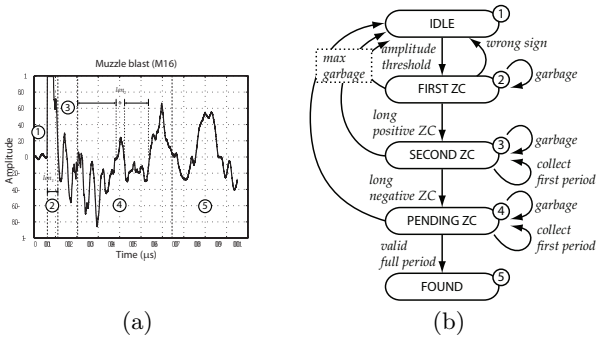


**Figure 5: Shockwave signal generated by a  $5.56 \times 45$  mm NATO projectile (a) and the state machine of the detection algorithm (b).**

ginning and end of the signal. Also, the length of the N-wave is fairly predictable—as it is described in Section 6.5—and is relatively short (200-300  $\mu\text{s}$ ). The shockwave detection core is continuously looking for two rising edges within a given interval. The state machine of the algorithm is shown in Figure 5(b). The input parameters are the minimum steepness of the edges ( $D$ ,  $E$ ), and the bounds on the length of the wave ( $L_{min}$ ,  $L_{max}$ ). The only feature calculated by the core is the length of the observed shockwave signal.

In contrast to shockwaves, the muzzle blast signatures are characterized by a long initial period (1-5 ms) where the first half period is significantly shorter than the second half [4]. Due to the physical limitations of the analog circuitry described at the beginning of this section, irregular oscillations and glitches might show up within this longer time window as they can be clearly seen in Figure 6(a). Therefore, the real challenge for the matching detection core is to identify the first and second half periods properly. The state machine (Figure 6(b)) does not work on the raw samples directly but is fed by a zero crossing (ZC) encoder. After the initial triggering, the detector attempts to collect those ZC segments which belong to the first period (positive amplitude) while discarding too short (in our terminology: garbage) segments—effectively implementing a rudimentary low-pass filter in the ZC domain. After it encounters a sufficiently long negative segment, it runs the same collection logic for the second half period. If too much garbage is discarded in the collection phases, the core resets itself to prevent the (false) detection of the halves from completely different periods separated by rapid oscillation or noise. Finally, if the constraints on the total length and on the length ratio hold, the core generates a detection event along with the actual length, amplitude and energy of the period calculated concurrently. The initial triggering mechanism is based on two amplitude thresholds: one static (but configurable) amplitude level and a dynamically computed one. The latter one is essential to adapt the sensor to different ambient noise environments and to temporarily suspend the muzzle blast detector after a shock wave event (oscillations in the analog section or reverberations in the sensor enclosure might otherwise trigger false muzzle blast detections). The dynamic noise level is estimated by a single pole recursive low-pass filter (cutoff @ 0.5 kHz) on the FPGA.





**Figure 6: Muzzle blast signature (a) produced by an M16 assault rifle and the corresponding detection logic (b).**

The detection cores were originally implemented in Java and evaluated on pre-recorded signals because of much faster test runs and more convenient debugging facilities. Later on, they were ported to VHDL and synthesized using the Xilinx ISE tool suite. The functional equivalence between the two implementations were tested by VHDL test benches and Python scripts which provided an automated way to exercise the detection cores on the same set of pre-recorded signals and to compare the results.

## 6. SENSOR FUSION

The sensor fusion algorithm receives detection messages from the sensor network and estimates the bullet trajectory, the shooter position, the caliber of the projectile and the type of the weapon. The algorithm consists of well separated computational tasks outlined below:

1. *Compute* muzzle blast and shockwave directions of arrivals for each individual sensor (see 6.1).
2. *Compute* range estimates. This algorithm can analytically fuse a pair of shockwave and muzzle blast AoA estimates. (see 6.2).
3. *Compute* a single trajectory from all shockwave measurements (see 6.3).
4. *If* trajectory available compute range (see 6.4).  
*else* compute shooter position first and then trajectory based on it. (see 6.4)
5. *If* trajectory available compute caliber (see 6.5).
6. *If* caliber available compute weapon type (see 6.6).

We describe each step in the following sections in detail.

### 6.1 Direction of arrival

The first step of the sensor fusion is to calculate the muzzle blast and shockwave AoA-s for each sensorboard. Each sensorboard has four microphones that measure the ToA-s. Since the microphone spacing is orders of magnitude smaller than the distance to the sound source, we can approximate the approaching sound wave front with a plane (far field assumption).

Let us formalize the problem for 3 microphones first. Let  $P_1$ ,  $P_2$  and  $P_3$  be the position of the microphones ordered by

time of arrival  $t_1 < t_2 < t_3$ . First we apply a simple geometry validation step. The measured time difference between two microphones cannot be larger than the sound propagation time between the two microphones:

$$|t_i - t_j| \leq |P_i - P_j|/c + \varepsilon$$

Where  $c$  is the speed of sound and  $\varepsilon$  is the maximum measurement error. If this condition does not hold, the corresponding detections are discarded. Let  $v(x, y, z)$  be the normal vector of the unknown direction of arrival. We also use  $r_1(x_1, y_1, z_1)$ , the vector from  $P_1$  to  $P_2$  and  $r_2(x_2, y_2, z_2)$ , the vector from  $P_1$  to  $P_3$ . Let's consider the projection of the direction of the motion of the wave front ( $v$ ) to  $r_1$  divided by the speed of sound ( $c$ ). This gives us how long it takes the wave front to propagate from  $P_1$  to  $P_2$ :

$$vr_1 = c(t_2 - t_1)$$

The same relationship holds for  $r_2$  and  $v$ :

$$vr_2 = c(t_3 - t_1)$$

We also know that  $v$  is a normal vector:

$$vv = 1$$

Moving from vectors to coordinates using the dot product definition leads to a quadratic system:

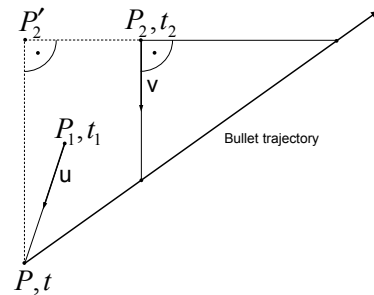
$$xx_1 + yy_1 + zz_1 = c(t_2 - t_1)$$

$$xx_2 + yy_2 + zz_2 = c(t_3 - t_1)$$

$$x^2 + y^2 + z^2 = 1$$

We omit the solution steps here, as they are straightforward, but long. There are two solutions (if the source is on the  $P_1P_2P_3$  plane the two solutions coincide). We use the fourth microphone's measurement—if there is one—to eliminate one of them. Otherwise, both solutions are considered for further processing.

### 6.2 Muzzle-shock fusion



**Figure 7: Section plane of a shot (at  $P$ ) and two sensors (at  $P_1$  and at  $P_2$ ). One sensor detects the muzzle blast's, the other the shockwave's time and direction of arrivals.**

Consider the situation in Figure 7. A shot was fired from  $P$  at time  $t$ . Both  $P$  and  $t$  are unknown. We have one muzzle blast and one shockwave detections by two different sensors

with AoA and hence, ToA information available. The muzzle blast detection is at position  $P_1$  with time  $t_1$  and AoA  $u$ . The shockwave detection is at  $P_2$  with time  $t_2$  and AoA  $v$ .  $u$  and  $v$  are normal vectors. It is shown below that these measurements are sufficient to compute the position of the shooter ( $P$ ).

Let  $P'_2$  be the point on the extended shockwave cone surface where  $PP'_2$  is perpendicular to the surface. Note that  $PP'_2$  is parallel with  $v$ . Since  $P'_2$  is on the cone surface which hits  $P_2$ , a sensor at  $P'_2$  would detect the same shockwave time of arrival ( $t_2$ ). The cone surface travels at the speed of sound ( $c$ ), so we can express  $P$  using  $P'_2$ :

$$P = P'_2 + cv(t_2 - t).$$

$P$  can also be expressed from  $P_1$ :

$$P = P_1 + cu(t_1 - t)$$

yielding

$$P_1 + cu(t_1 - t) = P'_2 + cv(t_2 - t).$$

$P_2P'_2$  is perpendicular to  $v$ :

$$(P'_2 - P_2)v = 0$$

yielding

$$(P_1 + cu(t_1 - t) - cv(t_2 - t) - P_2)v = 0$$

containing only one unknown  $t$ . One obtains:

$$t = \frac{(P_1 - P_2)v + uv t_1 - t_2}{c - uv - 1}.$$

From here we can calculate the shooter position  $P$ .

Let's consider the special single sensor case where  $P_1 = P_2$  (one sensor detects both shockwave and muzzle blast AoA). In this case:

$$t = \frac{uv t_1 - t_2}{uv - 1}.$$

Since  $u$  and  $v$  are not used separately only  $uv$ , the absolute orientation of the sensor can be arbitrary, we still get  $t$  which gives us the range.

Here we assumed that the shockwave is a cone which is only true for constant projectile speeds. In reality, the angle of the cone slowly grows; the surface resembles one half of an American football. The decelerating bullet results in a smaller time difference between the shockwave and the muzzle blast detections because the shockwave generation slows down with the bullet. A smaller time difference results in a smaller range, so the above formula underestimates the true range. However, it can still be used with a proper deceleration correction function. We leave this for future work.

### 6.3 Trajectory estimation

Danicki showed that the bullet trajectory and speed can be computed analytically from two independent shockwave measurements where both ToA and AoA are measured [2]. The method gets more sensitive to measurement errors as the two shockwave directions get closer to each other. In the special case when both directions are the same, the trajectory cannot be computed. In a real world application, the sensors are typically deployed on a plane approximately. In this case, all sensors located on one "side" of the trajectory measure almost the same shockwave AoA. To avoid

this error sensitivity problem, we consider shockwave measurement pairs only if the direction of arrival difference is larger than a certain threshold.

We have multiple sensors and one sensor can report two different directions (when only three microphones detect the shockwave). Hence, we typically have several trajectory candidates, i.e. one for each AoA pair over the threshold. We applied an outlier filtering and averaging method to fuse together the shockwave direction and time information and come up with a single trajectory. Assume that we have  $N$  individual shockwave AoA measurements. Let's take all possible unordered pairs where the direction difference is above the mentioned threshold and compute the trajectory for each. This gives us at most  $\frac{N(N-1)}{2}$  trajectories. A trajectory is represented by one point  $p_i$  and the normal vector  $v_i$  (where  $i$  is the trajectory index). We define the distance of two trajectories as the dot product of their normal vectors:

$$D(i, j) = v_i v_j$$

For each trajectory a neighbor set is defined:

$$N(i) := \{j | D(i, j) < R\}$$

where  $R$  is a radius parameter. The largest neighbor set is considered to be the core set  $C$ , all other trajectories are outliers. The core set can be found in  $O(N^2)$  time. The trajectories in the core set are then averaged to get the final trajectory.

It can happen that we cannot form any sensor pairs because of the direction difference threshold. It means all sensors are on the same side of the trajectory. In this case, we first compute the shooter position (described in the next section) that fixes  $p$  making  $v$  the only unknown. To find  $v$  in this case, we use a simple high resolution grid search and minimize an error function based on the shockwave directions.

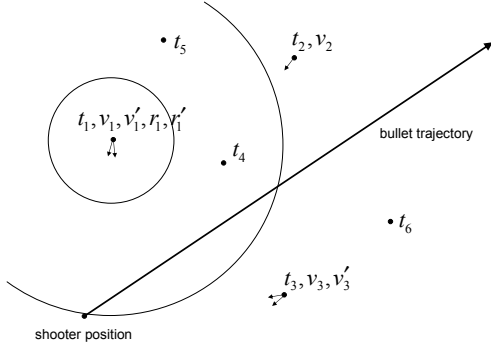
We have made experiments to utilize the measured shockwave length in the trajectory estimation. There are some promising results, but it needs further research.

### 6.4 Shooter position estimation

The shooter position estimation algorithm aggregates the following heterogenous information generated by earlier computational steps:

1. trajectory,
2. muzzle blast ToA at a sensor,
3. muzzle blast AoA at a sensor, which is effectively a bearing estimate to the shooter, and
4. range estimate at a sensor (when both shockwave and muzzle blast AoA are available).

Some sensors report only ToA, some has bearing estimate(s) also and some has range estimate(s) as well, depending on the number of successful muzzle blast and shockwave detections by the sensor. For an example, refer to Figure 8. Note that a sensor may have two different bearing and range estimates. 3 detections gives two possible AoA-s for muzzle blast (i.e. bearing) and/or shockwave. Furthermore, the combination of two different muzzle blast and shockwave AoA-s may result in two different ranges.



**Figure 8: Example of heterogenous input data for the shooter position estimation algorithm. All sensors have ToA measurements ( $t_1, t_2, t_3, t_4, t_5$ ), one sensor has a single bearing estimate ( $v_2$ ), one sensor has two possible bearings ( $v_3, v_3'$ ) and one sensor has two bearing and two range estimates ( $v_1, v_1', r_1, r_1'$ )**

In a multipath environment, these detections will not only contain gaussian noise, but also possibly large errors due to echoes. It has been showed in our earlier work that a similar problem can be solved efficiently with an interval arithmetic based bisection search algorithm [8]. The basic idea is to define a discrete consistency function over the area of interest and subdivide the space into 3D boxes. For any given 3D box, this function gives the number of measurements supporting the hypothesis that the shooter was within that box. The search starts with a box large enough to contain the whole area of interest, then zooms in by dividing and evaluating boxes. The box with the maximum consistency is divided until the desired precision is reached. Backtracking is possible to avoid getting stuck in a local maximum.

This approach has been shown to be fast enough for on-line processing. Note, however, that when the trajectory has already been calculated in previous steps, the search needs to be done only on the trajectory making it orders of magnitude faster.

Next let us describe how the consistency function is calculated in detail. Consider  $B$ , a three dimensional box, we would like to compute the consistency value of. First we consider only the ToA information. If one sensor has multiple ToA detections, we use the average of those times, so one sensor supplies at most one ToA estimate. For each ToA, we can calculate the corresponding time of the shot, since the origin is assumed to be in box  $B$ . Since it is a box and not a single point, this gives us an interval for the shot time. The maximum number of overlapping time intervals gives us the value of the consistency function for  $B$ . For a detailed description of the consistency function and search algorithm, refer to [8].

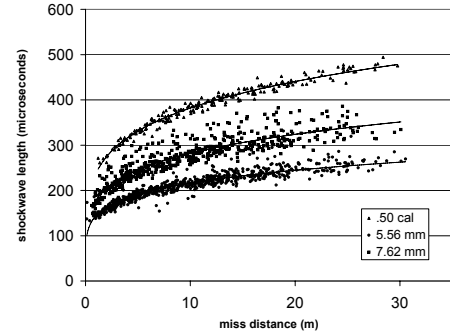
Here we extend the approach the following way. We modify the consistency function based on the bearing and range data from individual sensors. A bearing estimate supports  $B$  if the line segment starting from the sensor with the measured direction intersects the  $B$  box. A range supports  $B$ , if the sphere with the radius of the range and origin of the

sensor intersects  $B$ . Instead of simply checking whether the position specified by the corresponding bearing-range pairs falls within  $B$ , this eliminates the sensor's possible orientation error. The value of the consistency function is incremented by one for each bearing and range estimate that is consistent with  $B$ .

## 6.5 Caliber estimation

The shockwave signal characteristics has been studied before by Whitham [20]. He showed that the shockwave period  $T$  is related to the projectile diameter  $d$ , the length  $l$ , the perpendicular miss distance  $b$  from the bullet trajectory to the sensor, the Mach number  $M$  and the speed of sound  $c$ .

$$T = \frac{1.82Mb^{1/4}}{c(M^2-1)^{3/8}} \frac{d}{l^{1/4}} \approx \frac{1.82d}{c} \left(\frac{Mb}{l}\right)^{1/4}$$



**Figure 9: Shockwave length and miss distance relationship. Each data point represents one sensor-board after an aggregation of the individual measurements of the four acoustic channels. Three different caliber projectiles have been tested (196 shots, 10 sensors).**

To illustrate the relationship between miss distance and shockwave length, here we use all 196 shots with three different caliber projectiles fired during the evaluation. (During the evaluation we used data obtained previously using a few practice shots per weapon.) 10 sensors (4 microphones by sensor) measured the shockwave length. For each sensor, we considered the shockwave length estimation valid if at least three out of four microphones agreed on a value with at most 5 microsecond variance. This filtering leads to a 86% report rate per sensor and gets rid of large measurement errors. The experimental data is shown in Figure 9. Whitham's formula suggests that the shockwave length for a given caliber can be approximated with a power function of the miss distance (with a 1/4 exponent). Best fit functions on our data are:

$$.50 \text{ cal: } T = 237.75b^{0.2059}$$

$$7.62 \text{ mm: } T = 178.11b^{0.1996}$$

$$5.56 \text{ mm: } T = 144.39b^{0.1757}$$

To evaluate a shot, we take the caliber whose approximation function results in the smallest RMS error of the filtered sensor readings. This method has less than 1% caliber estimation error when an accurate trajectory estimate is available. In other words, caliber estimation only works if enough shockwave detections are made by the system to compute a trajectory.



## 6.6 Weapon estimation

We analyzed all measured signal characteristics to find weapon specific information. Unfortunately, we concluded that the observed muzzle blast signature is not characteristic enough of the weapon for classification purposes. The reflections of the high energy muzzle blast from the environment have much higher impact on the muzzle blast signal shape than the weapon itself. Shooting the same weapon from different places caused larger differences on the recorded signal than shooting different weapons from the same place.

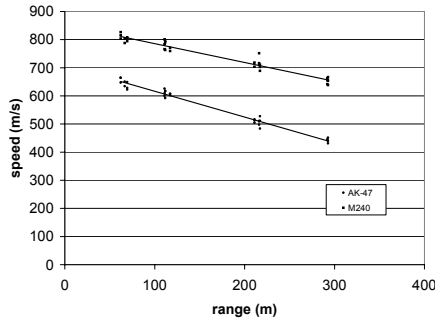


Figure 10: AK47 and M240 bullet deceleration measurements. Both weapons have the same caliber. Data is approximated using simple linear regression.

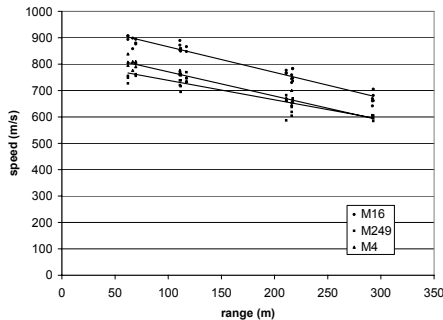


Figure 11: M16, M249 and M4 bullet deceleration measurements. All weapons have the same caliber. Data is approximated using simple linear regression.

However, the measured speed of the projectile and its caliber showed good correlation with the weapon type. This is because for a given weapon type and ammunition pair, the muzzle velocity is nearly constant. In Figures 10 and 11 we can see the relationship between the range and the measured bullet speed for different calibers and weapons. In the supersonic speed range, the bullet deceleration can be approximated with a linear function. In case of the 7.62 mm caliber, the two tested weapons (AK47, M240) can be clearly separated (Figure 10). Unfortunately, this is not necessarily true for the 5.56 mm caliber. The M16 with its higher muzzle speed can still be well classified, but the M4 and M249 weapons seem practically undistinguishable (Figure 11). However, this may be partially due to the limited number of practice shots we were able to take before the actual testing began. More training data may reveal better separation between the two weapons since their published muzzle velocities do differ somewhat.

The system carries out weapon classification in the following manner. Once the trajectory is known, the speed can be calculated for each sensor based on the shockwave geometry. To evaluate a shot, we choose the weapon type whose deceleration function results in the smallest RMS error of the estimated range-speed pairs for the estimated caliber class.

## 7. RESULTS

An independent evaluation of the system was carried out by a team from NIST at the US Army Aberdeen Test Center in April 2006 [19]. The experiment was setup on a shooting range with mock-up wooden buildings and walls for supporting elevated shooter positions and generating multipath effects. Figure 12 shows the user interface with an aerial photograph of the site. 10 sensor nodes were deployed on surveyed points in an approximately 30x30 m area. There were five fixed targets behind the sensor network. Several firing positions were located at each of the firing lines at 50, 100, 200 and 300 meters. These positions were known to the evaluators, but not to the operators of the system. Six different weapons were utilized: AK47 and M240 firing 7.62 mm projectiles, M16, M4 and M249 with 5.56mm ammunition and the .50 caliber M107.

Note that the sensors remained static during the test. The primary reason for this is that nobody is allowed downrange during live fire tests. Utilizing some kind of remote control platform would have been too involved for the limited time the range was available for the test. The experiment, therefore, did not test the mobility aspect of the system.

During the one day test, there were 196 shots fired. The results are summarized in Table 1. The system detected all shots successfully. Since a ballistic shockwave is a unique acoustic phenomenon, it makes the detection very robust. There were no false positives for shockwaves, but there were a handful of false muzzle blast detections due to parallel tests of artillery at a nearby range.

Shooter Range (m)	Localization Rate	Caliber Accuracy	Trajectory Azimuth Error (deg)	Trajectory Distance Error (m)	Distance Error (m)	No. of Shots
50	93%	100%	0.86	0.91	2.2	54
100	100%	100%	0.66	1.34	8.7	54
200	96%	100%	0.74	2.71	32.8	54
300	97%	97%	1.49	6.29	70.6	34
All	96%	99.5%	0.88	2.47	23.0	196

Table 1: Summary of results fusing all available sensor observations. All shots were successfully detected, so the detection rate is omitted. Localization rate means the percentage of shots that the sensor fusion was able to estimate the trajectory of. The caliber accuracy rate is relative to the shots localized and not all the shots because caliber estimation requires the trajectory. The trajectory error is broken down to azimuth in degrees and the actual distance of the shooter from the trajectory. The distance error shows the distance between the real shooter position and the estimated shooter position. As such, it includes the error caused by both the trajectory and that of the range estimation. Note that the traditional bearing and range measures are not good ones for a distributed system such as ours because of the lack of a single reference point.



Figure 12: The user interface of the system showing the experimental setup. The 10 sensor nodes are labeled by their ID and marked by dark circles. The targets are black squares marked T-1 through T-5. The long white arrows point to the shooter position estimated by each sensor. Where it is missing, the corresponding sensor did not have enough detections to measure the AoA of either the muzzle blast, the shockwave or both. The thick black line and large circle indicate the estimated trajectory and the shooter position as estimated by fusing all available detections from the network. This shot from the 100-meter line at target T-3 was localized almost perfectly by the sensor network. The caliber and weapon were also identified correctly. 6 out of 10 nodes were able to estimate the location alone. Their bearing accuracy is within a degree, while the range is off by less than 10% in the worst case.

The localization rate characterizes the system’s ability to successfully estimate the trajectory of shots. Since caliber estimation and weapon classification relies on the trajectory, non-localized shots are not classified either. There were 7 shots out of 196 that were not localized. The reason for missed shots is the trajectory ambiguity problem that occurs when the projectile passes on one side of all the sensors. In this case, two significantly different trajectories can generate the same set of observations (see [8] and also Section 6.3). Instead of estimating which one is more likely or displaying both possibilities, we decided not to provide a trajectory at all. It is better not to give an answer other than a shot alarm than misleading the soldier.

Localization accuracy is broken down to trajectory accuracy and range estimation precision. The angle of the estimated trajectory was better than 1 degree except for the 300 m range. Since the range should not affect trajectory estimation as long as the projectile passes over the network, we suspect that the slightly worse angle precision for 300 m is due to the hurried shots we witnessed the soldiers took

near the end of the day. This is also indicated by another datapoint: the estimated trajectory distance from the actual targets has an average error of 1.3 m for 300 m shots, 0.75 m for 200 m shots and 0.6 m for all but 300 m shots. As the distance between the targets and the sensor network was fixed, this number should not show a 2× improvement just because the shooter is closer.

Since the angle of the trajectory itself does not characterize the overall error—there can be a translation also—Table 1 also gives the distance of the shooter from the estimated trajectory. These indicate an error which is about 1-2% of the range. To put this into perspective, a trajectory estimate for a 100 m shot will very likely go through or very near the window the shooter is located at. Again, we believe that the disproportionately larger errors at 300 m are due to human errors in aiming. As the ground truth was obtained by knowing the precise location of the shooter and the target, any inaccuracy in the actual trajectory directly adds to the perceived error of the system.

We call the estimation of the shooter’s position on the calculated trajectory *range estimation* due to the lack of a better term. The range estimates are better than 5% accurate from 50 m and 10% for 100 m. However, this goes to 20% or worse for longer distances. We did not have a facility to test system before the evaluation for ranges beyond 100 m. During the evaluation, we ran into the problem of mistaking shockwave echoes for muzzle blasts. These echoes reached the sensors before the real muzzle blast for long range shots only, since the projectile travels 2-3× faster than the speed of sound, so the time between the shockwave (and its possible echo from nearby objects) and the muzzle blast increases with increasing ranges. This resulted in underestimating the range, since the system measured shorter times than the real ones. Since the evaluation we finetuned the muzzle blast detection algorithm to avoid this problem.

Distance	M16	AK47	M240	M107	M4	M249	M4-M249
50m	100%	100%	100%	100%	11%	25%	94%
100m	100%	100%	100%	100%	22%	33%	100%
200m	100%	100%	100%	100%	50%	22%	100%
300m	67%	100%	83%	100%	33%	0%	57%
All	96%	100%	97%	100%	23%	23%	93%

Table 2: Weapon classification results. The percentages are relative to the number of shots localized and not all shots, as the classification algorithm needs to know the trajectory and the range. Note that the difference is small; there were 189 shots localized out of the total 196.

The caliber and weapon estimation accuracy rates are based on the 189 shots that were successfully localized. Note that there was a single shot that was falsely classified by the caliber estimator. The 73% overall weapon classification accuracy does not seem impressive. But if we break it down to the six different weapons tested, the picture changes dramatically as shown in Table 2. For four of the weapons (AK14, M16, M240 and M107), the classification rate is almost 100%. There were only two shots out of approximately 140 that were missed. The M4 and M249 proved to be too similar and they were mistaken for each other most of the time. One possible explanation is that we had only a limited number of test shots taken with these weapons right before the evaluation and used the wrong deceleration approximation function. Either this or a similar mistake was made

since if we simply used the opposite of the system’s answer where one of these weapons were indicated, the accuracy would have improved 3x. If we consider these two weapons a single weapon class, then the classification accuracy for it becomes 93%.

Note that the AK47 and M240 have the same caliber (7.62 mm), just as the M16, M4 and M249 do (5.56 mm). That is, the system is able to differentiate between weapons of the same caliber. We are not aware of any system that classifies weapons this accurately.

### 7.1 Single sensor performance

As was shown previously, a single sensor alone is able to localize the shooter if it can determine both the muzzle blast and the shockwave AoA, that is, it needs to measure the ToA of both on at least three acoustic channels. While shockwave detection is independent of the range—unless the projectile becomes subsonic—, the likelihood of muzzle blast detection beyond 150 meters is not enough for consistently getting at least three per sensor node for AoA estimation. Hence, we only evaluate the single sensor performance for the 104 shots that were taken from 50 and 100 m. Note that we use the same test data as in the previous section, but we evaluate individually for each sensor.

Table 3 summarizes the results broken down by the ten sensors utilized. Since this is now not a distributed system, the results are given relative to the position of the given sensor, that is, a bearing and range estimate is provided. Note that many of the common error sources of the networked system do not play a role here. Time synchronization is not applicable. The sensor’s absolute location is irrelevant (just as the relative location of multiple sensors). The sensor’s orientation is still important though. There are several disadvantages of the single sensor case compared to the networked system: there is no redundancy to compensate for other errors and to perform outlier rejection, the localization rate is markedly lower, and a single sensor alone is not able to estimate the caliber or classify the weapon.

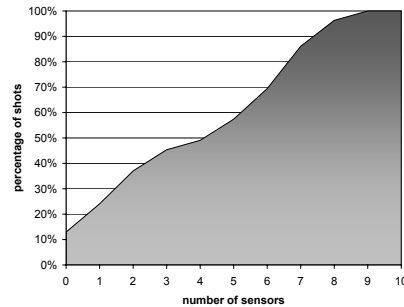
Sensor id	1	2	3	5	7	8	9	10	11	12
Loc. rate	44%	37%	53%	52%	19%	63%	51%	31%	23%	44%
Bearing (deg)	0.80	1.25	0.60	0.85	1.02	0.92	0.73	0.71	1.28	1.44
Range (m)	3.2	6.1	4.4	4.7	4.6	4.6	4.1	5.2	4.8	8.2

**Table 3: Single sensor accuracy for 108 shots fired from 50 and 100 meters. Localization rate refers to the percentage of shots the given sensor alone was able to localize. The bearing and range values are average errors. They characterize the accuracy of localization from the given sensor’s perspective.**

The data indicates that the performance of the sensors varied significantly especially considering the localization rate. One factor has to be the location of the given sensor including how far it was from the firing lines and how obstructed its view was. Also, the sensors were hand-built prototypes utilizing nowhere near production quality packaging/mounting. In light of these factors, the overall average bearing error of 0.9 degrees and range error of 5 m with a microphone spacing of less than 10 cm are excellent. We believe that professional manufacturing and better microphones could easily achieve better performance than the best sensor in our experiment (>60% localization rate and 3 m range error).

Interestingly, the largest error in range was a huge 90 m clearly due to some erroneous detection, yet the largest bearing error was less than 12 degrees which is still a good indication for the soldier where to look.

The overall localization rate over all single sensors was 42%, while for 50 m shots only, this jumped to 61%. Note that the firing range was prepared to simulate an urban area to some extent: there were a few single- and two-storey wooden structures built both in and around the sensor deployment area and the firing lines. Hence, not all sensors had line-of-sight to all shooting positions. We estimate that 10% of the sensors had obstructed view to the shooter on average. Hence, we can claim that a given sensor had about 50% chance of localizing a shot within 130 m. (Since the sensor deployment area was 30 m deep, 100 m shots correspond to actual distances between 100 and 130 m.) Again, we emphasize that localization needs at least three muzzle blast and three shockwave detections out of a possible four for each per sensor. The detection rate for single sensors—corresponding to at least one shockwave detection per sensor—was practically 100%.



**Figure 13: Histogram showing what fraction of the 104 shots taken from 50 and 100 meters were localized by at most how many individual sensors alone. 13% of the shots were missed by every single sensor, i.e., none of them had both muzzle blast and shockwave AoA detections. Note that almost all of these shots were still accurately localized by the networked system, i.e. the sensor fusion using all available observations in the sensor network.**

It would be misleading to interpret these results as the system missing half the shots. As soldiers never work alone and the sensor node is relatively cheap to afford having every soldier equipped with one, we also need to look at the overall detection rates for every shot. Figure 13 shows the histogram of the percentage of shots vs. the number of individual sensors that localized it. 13% of shots were not localized by any sensor alone, but 87% was localized by at least one sensor out of ten.

### 7.2 Error sources

In this section, we analyze the most significant sources of error that affect the performance of the networked shooter localization and weapon classification system. In order to correlate the distributed observations of the acoustic events, the nodes need to have a common time and space reference. Hence, errors in the time synchronization, node localization and node orientation all degrade the overall accuracy of the system.

Our time synchronization approach yields errors significantly less than 100 microseconds. As the sound travels about 3 cm in that time, time synchronization errors have a negligible effect on the system.

On the other hand, node location and orientation can have a direct effect on the overall system performance. Notice that to analyze this, we do not have to resort to simulation, instead we can utilize the real test data gathered at Aberdeen. But instead of using the real sensor locations known very accurately and the measured and calibrated almost perfect node orientations, we can add error terms to them and run the sensor fusion. This exactly replicates how the system would have performed during the test using the imprecisely known locations and orientations.

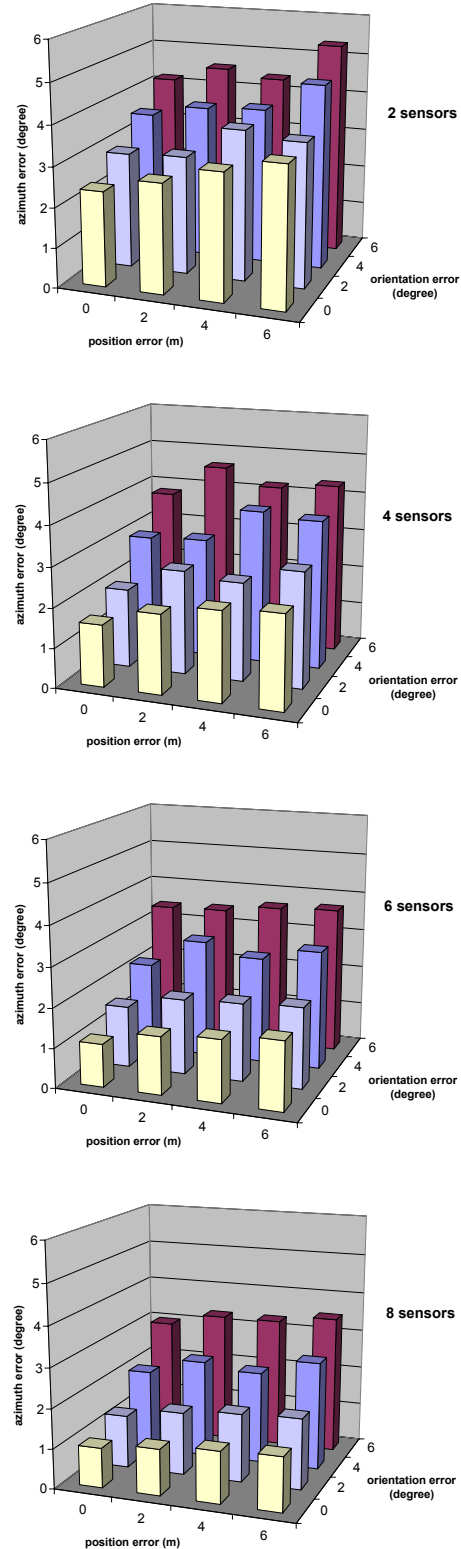
Another aspect of the system performance that can be evaluated this way is the effect of the number of available sensors. Instead of using all ten sensors in the data fusion, we can pick any subset of the nodes to see how the accuracy degrades as we decrease the number of nodes.

The following experiment was carried out. The number of sensors were varied from 2 to 10 in increments of 2. Each run picked the sensors randomly using a uniform distribution. At each run each node was randomly "moved" to a new location within a circle around its true position with a radius determined by a zero-mean Gaussian distribution. Finally, the node orientations were perturbed using a zero-mean Gaussian distribution. Each combination of parameters were generated 100 times and utilized for all 196 shots.

The results are summarized in Figure 14. There is one 3D bar chart for each of the experiment sets with the given fixed number of sensors. The x-axis shows the node location error, that is, the standard deviation of the corresponding Gaussian distribution that was varied between 0 and 6 meters. The y-axis shows the standard deviation of the node orientation error that was varied between 0 and 6 degrees. The z-axis is the resulting trajectory azimuth error. Note that the elevation angles showed somewhat larger errors than the azimuth. Since all the sensors were in approximately a horizontal plane and only a few shooter positions were out of the same plane and only by 2 m or so, the test was not sufficient to evaluate this aspect of the system.

There are many interesting observations one can make by analyzing these charts. Node location errors in this range have a small effect on accuracy. Node orientation errors, on the other hand, noticeably degrade the performance. Still the largest errors in this experiment of 3.5 degrees for 6 sensors and 5 degrees for 2 sensors are still very good.

Note that as the location and orientation errors increase and the number of sensors decrease, the most significantly affected performance metric is the localization rate. See Table 4 for a summary. Successful localization goes down from almost 100% to 50% when we go from 10 sensors to 2 even without additional errors. This is primarily caused by geometry: for a successful localization, the bullet needs to pass over the sensor network, that is, at least one sensor should be on the side of the trajectory other than the rest of the nodes. (This is a simplification for illustrative purposes. If all the sensors and the trajectory are not coplanar, localization may be successful even if the projectile passes on one side of the network. See Section 6.3.) As the numbers of sensors decreased in the experiment by randomly selecting a subset, the probability of trajectories abiding by this rule decreased. This also means that even if there are



**Figure 14:** The effect of node localization and orientation errors on azimuth accuracy with 2, 4, 6 and 8 nodes. Note that the chart for 10 nodes is almost identical for the 8-node case, hence, it is omitted.

many sensors (i.e. soldiers), but all of them are right next to each other, the localization rate will suffer. However, when the sensor fusion does provide a result, it is still accurate even with few available sensors and relatively large individual errors. A very few consistent observation lead to good accuracy as the inconsistent ones are discarded by the algorithm. This is also supported by the observation that for the cases with the higher number of sensors (8 or 10), the localization rate is hardly affected by even large errors.

Errors/Sensors	2	4	6	8	10
0 m, 0 deg	54%	87%	94%	95%	96%
2 m, 2 deg	53%	80%	91%	96%	96%
6 m, 0 deg	43%	79%	88%	94%	94%
0 m, 6 deg	44%	78%	90%	93%	94%
6 m, 6 deg	41%	73%	85%	89%	92%

**Table 4: Localization rate as a function of the number of sensors used, the sensor node location and orientation errors.**

One of the most significant observations on Figure 14 and Table 4 is that there is hardly any difference in the data for 6, 8 and 10 sensors. This means that there is little advantage of adding more nodes beyond 6 sensors as far as the accuracy is concerned.

The speed of sound depends on the ambient temperature. The current prototype considers it constant that is typically set before a test. It would be straightforward to employ a temperature sensor to update the value of the speed of sound periodically during operation. Note also that wind may adversely affect the accuracy of the system. The sensor fusion, however, could incorporate wind speed into its calculations. It would be more complicated than temperature compensation, but could be done.

Other practical issues also need to be looked at before a real world deployment. Silencers reduce the muzzle blast energy and hence, the effective range the system can detect it at. However, silencers do not effect the shockwave and the system would still detect the trajectory and caliber accurately. The range and weapon type could not be estimated without muzzle blast detections. Subsonic weapons do not produce a shockwave. However, this is not of great significance, since they have shorter range, lower accuracy and much less lethality. Hence, their use is not widespread and they pose less danger in any case.

Another issue is the type of ammunition used. Irregular armies may use substandard, even hand manufactured bullets. This effects the muzzle velocity of the weapon. For weapon classification to work accurately, the system would need to be calibrated with the typical ammunition used by the given adversary.

## 8. RELATED WORK

Acoustic detection and recognition has been under research since the early fifties. The area has a close relevance to the topic of supersonic flow mechanics [20]. Fansler analyzed the complex near-field pressure waves that occur within a foot of the muzzle blast. Fansler’s work gives a good idea of the ideal muzzle blast pressure wave without contamination from echoes or propagation effects [4]. Experiments with greater distances from the muzzle were con-

ducted by Stoughton [18]. The measurements of the ballistic shockwaves using calibrated pressure transducers at known locations, measured bullet speeds, and miss distances of 3 - 55 meters for 5.56 mm and 7.62 mm projectiles were made. Results indicate that ground interaction becomes a problem for miss distances of 30 meters or larger.

Another area of research is the signal processing of gunfire acoustics. The focus is on the robust detection and length estimation of small caliber acoustic shockwaves and muzzle blasts. Possible techniques for classifying signals as either shockwaves or muzzle blasts includes short-time Fourier Transform (STFT), the Smoothed Pseudo Wigner-Ville distribution (SPWVD), and a discrete wavelet transformation (DWT). Joint time-frequency (JTF) spectrograms are used to analyze the typical separation of the shockwave and muzzle blast transients in both time and frequency. Mays concludes that the DWT is the best method for classifying signals as either shockwaves or muzzle blasts because it works well and is less expensive to compute than the SPWVD [10]. The edges of the shockwave are typically well defined and the shockwave length is directly related to the bullet characteristics. A paper by Sadler [14] compares two shockwave edge detection methods: a simple gradient-based detector, and a multi-scale wavelet detector. It also demonstrates how the length of the shockwave, as determined by the edge detectors, can be used along with Whithams equations [20] to estimate the caliber of a projectile. Note that the available computational performance on the sensor nodes, the limited wireless bandwidth and real-time requirements render these approaches infeasible on our platform.

A related topic is the research and development of experimental and prototype shooter location systems. Researchers at BBN have developed the Bullet Ears system [3] which has the capability to be installed in a fixed position or worn by soldiers. The fixed system has tetrahedron shaped microphone arrays with 1.5 meter spacing. The overall system consists of two to three of these arrays spaced 20 to 100 meters from each other. The soldier-worn system has 12 microphones as well as a GPS antenna and orientation sensors mounted on a helmet. There is a low speed RF connection from the helmet to the processing body. An extensive test has been conducted to measure the performance of both type of systems. The fixed systems performance was one order of magnitude better in the angle calculations while their range performance were matched. The angle accuracy of the fixed system was dominantly less than one degree while it was around five degrees for the helmet mounted one. The range accuracy was around 5 percent for both of the systems. The problem with this and similar centralized systems is the need of the one or handful of microphone arrays to be in line-of-sight of the shooter. A sensor networked based solution has the advantage of widely distributed sensing for better coverage, multipath effect compensation and multiple simultaneous shot resolution [8]. This is especially important for operation in acoustically reverberant urban areas. Note that BBN’s current vehicle-mounted system called BOOMERANG, a modified version of Bullet Ears, is currently used in Iraq [1].

The company ShotSpotter specializes in law enforcement systems that report the location of gunfire to police within seconds. The goal of the system is significantly different than that of military systems. Shotspotter reports 25 m typical accuracy which is more than enough for police to

respond. They are also manufacturing experimental soldier wearable and UAV mounted systems for military use [16], but no specifications or evaluation results are publicly available.

## 9. CONCLUSIONS

The main contribution of this work is twofold. First, the performance of the overall distributed networked system is excellent. Most noteworthy are the trajectory accuracy of one degree, the correct caliber estimation rate of well over 90% and the close to 100% weapon classification rate for 4 of the 6 weapons tested. The system proved to be very robust when increasing the node location and orientation errors and decreasing the number of available sensors all the way down to a couple. The key factor behind this is the sensor fusion algorithm's ability to reject erroneous measurements. It is also worth mentioning that the results presented here correspond to the first and only test of the system beyond 100 m and with six different weapons. We believe that with the lessons learned in the test, a consecutive field experiment could have showed significantly improved results especially in range estimation beyond 100 m and weapon classification for the remaining two weapons that were mistaken for each other the majority of the times during the test.

Second, the performance of the system when used in standalone mode, that is, when single sensors alone provided localization, was also very good. While the overall localization rate of 42% per sensor for shots up to 130 m could be improved, the bearing accuracy of less than a degree and the average 5% range error are remarkable using the hand-made prototypes of the low-cost nodes. Note that 87% of the shots were successfully localized by at least one of the ten sensors utilized in standalone mode.

We believe that the technology is mature enough that a next revision of the system could be a commercial one. However, important aspects of the system would still need to be worked on. We have not addressed power management yet. A current node runs on 4 AA batteries for about 12 hours of continuous operation. A deployable version of the sensor node would need to be asleep during normal operation and only wake up when an interesting event occurs. An analog trigger circuit could solve this problem, however, the system would miss the first shot. Instead, the acoustic channels would need to be sampled and stored in a circular buffer. The rest of the board could be turned off. When a trigger wakes up the board, the acoustic data would be immediately available. Experiments with a previous generation sensor board indicated that this could provide a 10x increase in battery life. Other outstanding issues include weatherproof packaging and ruggedization, as well as integration with current military infrastructure.

## 10. REFERENCES

- [1] BBN technologies website. <http://www.bbn.com>.
- [2] E. Danicki. Acoustic sniper localization. *Archives of Acoustics*, 30(2):233–245, 2005.
- [3] G. L. Duckworth et al. Fixed and wearable acoustic counter-sniper systems for law enforcement. In E. M. Carapezza and D. B. Law, editors, *Proc. SPIE Vol. 3577*, p. 210-230, pages 210–230, Jan. 1999.
- [4] K. Fansler. Description of muzzle blast by modified scaling models. *Shock and Vibration*, 5(1):1–12, 1998.
- [5] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesC language: a holistic approach to networked embedded systems. *Proceedings of Programming Language Design and Implementation (PLDI)*, June 2003.
- [6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. in *Proc. of ASPLOS 2000*, Nov. 2000.
- [7] B. Kusý, G. Balogh, P. Völgyesi, J. Sallai, A. Nádas, A. Lédeczi, M. Maróti, and L. Meertens. Node-density independent localization. *Information Processing in Sensor Networks (IPSN 06) SPOTS Track*, Apr. 2006.
- [8] A. Lédeczi, A. Nádas, P. Völgyesi, G. Balogh, B. Kusý, J. Sallai, G. Pap, S. Dóra, K. Molnár, M. Maróti, and G. Simon. Countersniper system for urban warfare. *ACM Transactions on Sensor Networks*, 1(1):153–177, Nov. 2005.
- [9] M. Maróti. Directed flood-routing framework for wireless sensor networks. In *Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware*, pages 99–114, New York, NY, USA, 2004. Springer-Verlag New York, Inc.
- [10] B. Mays. Shockwave and muzzle blast classification via joint time frequency and wavelet analysis. Technical report, Army Research Lab Adelphi MD 20783-1197, Sept. 2001.
- [11] TinyOS Hardware Platforms. <http://tinycos.net/scoop/special/hardware>.
- [12] Crossbow MICAz (MPR2400) Radio Module. <http://www.xbow.com/Products/productsdetails.aspx?sid=101>.
- [13] PicoBlaze User Resources. [http://www.xilinx.com/ipcenter/processor\\_central/picoblaze/picoblaze\\_user\\_resources.htm](http://www.xilinx.com/ipcenter/processor_central/picoblaze/picoblaze_user_resources.htm).
- [14] B. M. Sadler, T. Pham, and L. C. Sadler. Optimal and wavelet-based shock wave detection and estimation. *Acoustical Society of America Journal*, 104:955–963, Aug. 1998.
- [15] J. Sallai, B. Kusý, A. Lédeczi, and P. Dutta. On the scalability of routing-integrated time synchronization. *3rd European Workshop on Wireless Sensor Networks (EWSN 2006)*, Feb. 2006.
- [16] ShotSpotter website. <http://www.shotspotter.com/products/military.html>.
- [17] G. Simon, M. Maróti, A. Lédeczi, G. Balogh, B. Kusý, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 1–12, New York, NY, USA, 2004. ACM Press.
- [18] R. Stoughton. Measurements of small-caliber ballistic shock waves in air. *Acoustical Society of America Journal*, 102:781–787, Aug. 1997.
- [19] B. A. Weiss, C. Schlenoff, M. Shneier, and A. Virts. Technology evaluations and performance metrics for soldier-worn sensors for assist. In *Performance Metrics for Intelligent Systems Workshop*, Aug. 2006.
- [20] G. Whitham. Flow pattern of a supersonic projectile. *Communications on pure and applied mathematics*, 5(3):301, 1952.