Time of Arrival Data Fusion for Source Localization

György Balogh, Ákos Lédeczi,
Miklós Maróti
Institute for Software Integrated Systems
Vanderbilt University, USA
akos.ledeczi@vanderbilt.edu

Gyula Simon
Department of Computer Science
University of Veszprem,
Hungary
simon@dcs.vein.hu

Abstract

The paper presents two sensor fusion algorithms that were utilized in a wireless sensor network-based acoustic countersniper system. The technique for fusing muzzle blast Time Of Arrival (TOA) data is able to eliminate multipath effects prevalent in urban environments and to resolve multiple simultaneous acoustic sources. The approach for fusing shockwave TOA data that are generated by supersonic projectiles is able to reconstruct the trajectory with an accuracy of 1 degree in both azimuth and elevation for long range shots. If a few muzzle blast detections are also available then accurate range estimation is also performed. The system performance was demonstrated multiple times at different MOUT (Military Operations in Urban Terrain) facilities of the US Army.

1 Introduction

The sensor fusion algorithms developed for an ad-hoc wireless sensor network-based acoustic counter-sniper system are presented. The system is able to localize the shooter position along with the projectile trajectory using observed acoustic events, the muzzle blast of the gun and the ballistic shockwave of the supersonic projectile. The sensor network contains a large number of cheap sensors communicating through an ad-hoc wireless network, thus it tolerates multiple sensor failures and provides good coverage and high accuracy, even in challenging urban environments.

Acoustic source localization in noisy and reverberant environments has been an important application area, lately addressed by sensor networks as well. Traditional approaches successfully attack the problem by sophisticated near-field beam forming methods, providing accurate and optimal solutions [2], [3]. As opposed to current solutions using only a handful of measurement sources, sensor networks could utilize a few orders of magnitude higher number of sensors, thus providing potentially higher performance, fault tolerance, and better coverage. Unfortunately, beam forming techniques require the transmission of large amount of data between sensors and the base station [1].

Note that Yao et al. employed sensor networks for source localization using beamforming [13]. However, the hardware platform applied was based on PDA class devices that have orders of magnitude more resources in terms of memory, computational power and communication bandwidth than typical sensor networks. Furthermore, neither echoes nor multiple simultaneous sources were addressed in their work.

Two-step techniques [2, 4, 5] represent an alternative to beamforming. Here in the first step, characteristic points of the signal are identified (e.g. start of the signal) and time of arrival (TOA) or time difference of arrival (TDOA) time-stamps are assigned to them. In the second step, the source location is calculated, using the measured TOA/TDOA data set only. This solution is feasible even in networks with very limited bandwidth, since only a few bytes have to be transmitted from each sensor.

Errors of TOA or TDOA data can be classified as small inaccuracies called measurement noise, usually resulting from ambient noise, time synchronization error, or imprecise sensor localization; and large errors from erroneous signal detection or echoes. (Here the terms small and large refer to the measurement error with respect to the desired localization accuracy.) Note that in an experiment performed by the authors in urban terrain [6], on average 10-25% of the sensor readings contained large erroneous data typically due to multipath propagation.

Calculating the source location from TOA or TDOA data is a theoretically simple task. Each pair of data determines a hyperboloid surface, thus four measurements identify a single point of solution in space. However, this solution can be very sensitive to inaccurate input data. Fortunately, solving an over-determined problem defined by many measurements decrease the effect of measurement noise [3, 5, 8]. Large measurement errors, however, may cause unacceptably high localization error using traditional solvers, thus, the elimination of the erroneous data seems to be inevitable. Since usually no a priori information is available on the accuracy of the measurements, a possible solution can be an iterative search for the maximal set of consistent measurement data. Although it is a theoretically possible solution, no computationally efficient way is known to perform this operation, especially in case of a large input data set easily provided by dense sensor networks.

Another serious problem arises when multiple concurrent sources are present. Traditional solutions cannot handle TOA/TDOA measurements resulting from simultaneous sources. To our best knowledge, no two-step system has been able to localize multiple concurrent sources in the past.

The rest of the paper is organized as follows. Section II presents a brief overview the countersniper system. It is followed by a detailed discussion of the muzzle blast sensor fusion algorithm and its evaluation. Section IV introduces the shockwave fusion technique and the results we gathered during field testing. Finally, we present our conclusions.

2 Countersniper system overview

The firing of a typical rifle results in two acoustic phenomena. The muzzle blast produces a spherical wave front, traveling at the speed of sound from the muzzle of the gun. 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 bullet is constant). The sensors autonomously detect the shockwave and/or the muzzle blast, measure their times of arrival (TOA), and send the measured results to a base station through the ad-hoc wireless sensor network. Then the fusion algorithms running on the base station determine the location of the shooter and the trajectory of the projectile.

The sensor nodes are based on the UC Berkeley MICA2 mote device running the TinyOS embedded operating system [7], a widely used component-based architecture targeting scalable wireless sensor network applications. Each MICA2 mote is furnished with an ATmega 128L 8-bit microcontroller with 128 Kbytes instruction memory, 4 Kbytes data memory and typical embedded peripherals built in. The on-board radio transceiver operates in the 433 MHz ISM band and has a maximum transfer rate of 38.4 Kbits/sec with the maximum range of about 300 feet.

Real-time detection and classification of acoustic events require computational resources not available in the sensor node. Thus, we have designed an application specific sensor board built around a low-power fixed point ADSP-218x digital signal processor running at 50MHz. Its internal program (48KB) and data (56KB) memory buffers with advanced addressing modes and DMA controllers enable sophisticated signal processing and advanced power management methods.

Two independent analog input channels with low-cost electret microphones pick up the incoming acoustic signals utilizing 2-stage amplification with software programmable gain (0-54dB). The A/D converters sample at up to 100kSPS at 12-bit resolution. Analog comparators with software adjustable thresholds can be used to wake up the signal processor from low-power sleep mode, enabling continuous operation for weeks on two AA batteries. A standard I2C bus connection and programmable interrupt and acknowledgement lines enable integration with the MICA2 mote [7].

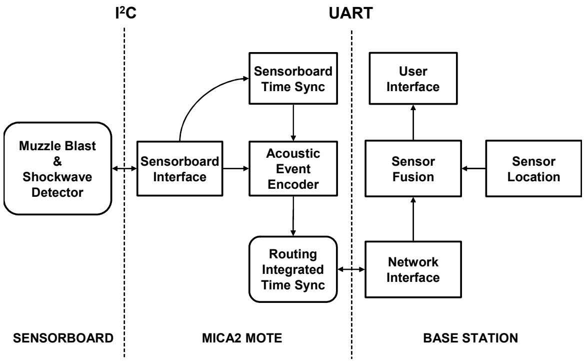

Figure 1. Software Architecture

The software architecture of the system is depicted in Figure 1. The Muzzle Blast and Shockwave Detectors are running on the DSP of the custom sensor board board. The TOA data from the board is sent through the I2C interface to the mote. The Acoustic Event Encoder assembles a packet containing the TOA data and passes it to the Routing service.

In addition to transporting the packets to the base station through multiple hops, the Message Routing service also performs implicit time synchronization as described in the next section. The Base Station runs the Sensor Fusion algorithm utilizing known sensor positions and displays the results on the User Interface.

2.1 The fusion of time

The countersniper system does not use a continuous time synchronization service providing a virtual global clock. From an application standpoint it is desirable that the system operates in a stealthy manner; it should not use any radio communication when not necessary. This also saves power, a precious resource in wireless sensor networks. But how can we correlate multiple observations in the absence of a global clock?

Our implicit time synchronization approach is based on the Elapsed Time of Arrival (ETA) algorithm [10]. The ETA algorithm works by storing the time of all events using the local clock of the node. Local-time conversions occur whenever messages are transmitted from one node to another. This is enabled by our highly accurate time stamping service that is part of the MAC layer of the radio stack of the mote [11]. At transmission time the elapsed time since the event occurrence is computed and included as a separate field in the message. On receipt, a node computes the event time by subtracting the elapsed time from the receiving node's local time.

The Routing Integrated Time Synchronization (RITS) protocol is an extension of ETA over multiple hops by integrating it into the Directed Flood Routing Framework (DFRF) [12]. After detecting an event, the sensor node provides RITS with the local time of the event occurrence and a data packet. In the countersniper system this is a single bit differentiating between muzzle blast and shockwave. RITS, configured for convergecast, sends the packet to the base station along a multi-hop path while maintaining the event time using ETA. At each hop it converts the event time expressed in the local time of the sender to that of the receiver. When the message arrives at the base station the detection time of the acoustic event is provided to the sensor fusion algorithm using the *base station clock* in a transparent manner. Different observations of the same event then can be accurately correlated.

We observed that the maximum pairwise error of RIPS in a 10-hop 60-node setup of MICA2 motes was well under 100 microseconds. As the sound travels approximately 3 cm-s in this time, it has a negligible effect on the overall localization accuracy.

3 Muzzle blast fusion

The muzzle blast sensor fusion algorithm uses TOA measurements as input data. The algorithm is very robust against small or large measurement errors and it can handle multiple simultaneous sources as well. The fusion algorithm is a search on a 3-dimensional surface defined by a consistency function. The global maximum of the surface defines the solution, which is typically close to the true source location. Multiple sources are shown as multiple local maxima on the surface. Note that since erroneous input data is permitted, it is always possible to create an artificial scenario where enough bad measurements create a false optimum. It is very unlikely, though, that in reality such ill-posed cases occur, even in the presence of many large erroneous measurements, as shown by experimental data [6].

3.1 Consistency function

For the sake of clarity let's suppose that there is only one characteristic point of the signal (e.g. start of the muzzle blast of a shot) and thus each sensor assigns one TOA measurement for each acoustic event emitted by the source. The number of sensors is *N* and thus, the input of the fusion algorithm is the set of time measurements *t1, t2, ...,* t_N, along with the corresponding known sensor positions (x_i, y_i, z_i), $i = 1,\ldots,N$.

Since the ith sensor detects the TOA of the acoustic event at position (x_i, y_i, z_i), the time of emission can be calculated as

$$t^i(x,y,z) = t_i - \frac{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}}{v_s}, \quad (1)$$

where v_s is the speed of sound and $t^i(x,y,z)$ is the emission time estimated by sensor i, provided the source is located at (x,y,z). If (x,y,z) corresponds to the true source location and the measurements are accurate then every sensor will estimate the very same $t^i(x,y,z)$ value. If the source location is not (x,y,z) then the time estimates of the sensors will not be the same, though there are (x,y,z) locations where some sensors will estimate the same $t^i(x,y,z)$ value. If the measurements are not accurate then the sensor estimates $t^i(x,y,z)$ are not exactly the same, even if (x,y,z) is the true location, but will be in a close proximity of the true source time. Measurements with large errors will be outliers.

The concept is illustrated in Figure 2, where the sensors are placed on a grid in a plane. For simplicity, assume that the grid size is 1 and the speed of sound is 1. The sensors $S_1, S_2, \ldots, S_6$ measure TOA values *t*1, *t*2, …, *t*6, as shown in the figure. Note that measurement noise is present for all sensors and S6 reported an erroneous value. For positions P1 and P2 the estimated emission times for each sensor are shown in the timelines. For P1, which is the true source location, five sensors estimate emission times around 10, which is the true value, and the faulty sensor estimated a different value.

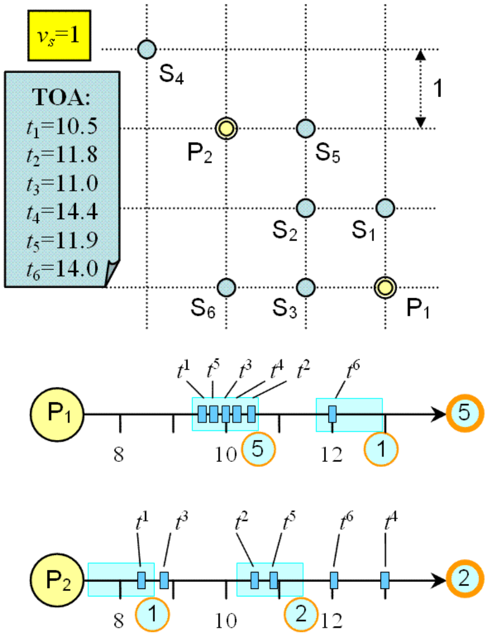

Figure 2. Illustration of the consistency function

For P2 the estimated emission times are scattered through the timeline, no definite cluster can be found. Although in the presence of small measurement noise the values $t^i(\mathrm{P}_1)$, i= 1,2,…5, are not the same, but are close to each other and thus are considered consistent measurements, while $t^6(\mathrm{P}_1)$ is still far from the group, and thus t_6 is not consistent with any of them. In general, a set of measurement is considered to be consistent with an event time *t* if the absolute difference between *t* and any elements in the set is smaller than the predefined positive uncertainty value $\tau/2$.

In the example of Figure 2, a sliding window of length $\tau = 1$ is moved along the timeline of P1. Around the value of 10, five consistent estimates are in the window, while at other places the sliding window contains less consistent estimates. Thus, the maximum number of consistent estimates for P1 is five, while for P2 this number is only 2. Formally, if the inequity

$$\left|t^i(x,y,z) - t\right| \le \tau/2 \qquad (2)$$

holds for a time value t and for sensors $i = i_1, i_2, \ldots, i_{K(x,y,z,t,\tau)}$, where $K(x,y,z,t,\tau)$ is the number of consistent measurements for position (x,y,z) and time t with uncertainty τ, then the consistency function is defined as

$$C_\tau(x,y,z) = \max_t K(x,y,z,t,\tau). \qquad (3)$$

The optimal source location estimate is at the maximum of the consistency function:

$$(x_s, y_s, z_s) = \underset{(x,y,z)}{\arg\max}\, C_\tau(x,y,z) \cdot \qquad (4)$$

The corresponding $\hat{t}_s(x,y,z)$ emission time estimate for position (x,y,z) is

$$\hat{t}_s(x,y,z) = \underset{t}{\arg\max}\, K(x,y,z,t,\tau) \cdot \qquad (5)$$

In the proximity of the true location of the source the value of the consistency function is high even for small uncertainty values, while further away from the true location the consistency value decreases for the same uncertainty value. In ideal circumstances, when all of the active sensors provide correct measurements, and the location and time synchronization errors of every sensor are zero then the value of the consistency function is:

$$C_\tau(x_s, y_s, z_s) = N \qquad (6)$$

for any $\tau \ge 0$. In the presence of localization error and time synchronization error equity (6) holds for

$$\tau \ge 2(\Delta/v_s + \mathrm{T}), \qquad (7)$$

where Δ and T are the maximum sensor position error and the maximum time synchronization error, respectively. In the presence of faulty sensor readings the value of the consistency function is less than N and equals to the number of good measurements.

An important property of the consistency function is its monotony with respect to the uncertainty:

$$C_{\tau_1}(x,y,z) \le C_{\tau_2}(x,y,z), \text{ if } \tau_1 < \tau_2. \qquad (8)$$

Since no analytical solution is known to find the maximum of the consistency function, a numerical search algorithm is necessary. After partitioning the guarded space into small cubes, with edge length of

τ, it is theoretically possible to find the global maximum by an exhaustive search. However, the required number of operations is too high for a large guarded area to provide an acceptably low latency time. Based on property (8), a search mechanism was introduced in [6]: it utilizes an iterative partitioning algorithm using Generalized Bisection method based on interval arithmetic [9]. The uncertainty value of the search is set to $\tau = 2(\Delta/v_s + T)$, using a priori information about the accuracy of the time synchronization and the sensor localization. The iterative solution provides approximately eight orders of magnitude faster solution than the simple exhaustive search, providing approximately one second latency time in practical tests [6].

3.2 Evaluation

The performance of the fusion algorithm was tested as part of the countersniper system [6]. The input data was provided by acoustic sensor nodes measuring the TOA of muzzle blast events. In the tests 60 sensors were deployed in an approximately 100m x 50m built-in area. From 20 reference shooter positions 171 shots were fired using both live and blank ammunition. As shown in Figure 3, the system accuracy was remarkably good: the average 2D and 3D errors were 0.6m and 1.3m, respectively.

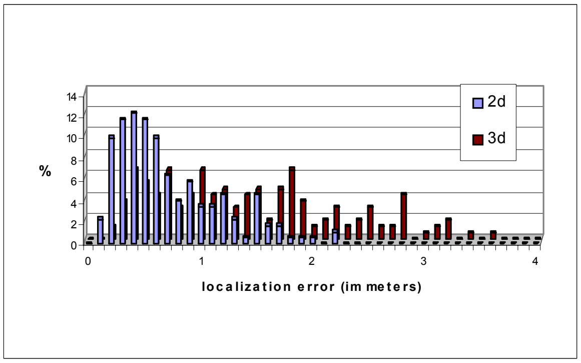

Figure 3. Histogram of the localization error

The accuracy of the fusion depends on the applicable uncertainty value τ, which is dominated by the sensor localization error. The sensitivity of the fusion algorithm to this source of inaccuracies was tested by simulation. The initial individual sensor localization errors during the test were not known exactly, but the average was approximated as 0.3m. In the simulation experiment, additional random errors were added to the sensor positions, generating 1000 experiments. The results are shown in Figure 4. As expected, the increasing sensor position error caused the localization error to grow approximately in a linear fashion in the range 0-2m of sensor position error.

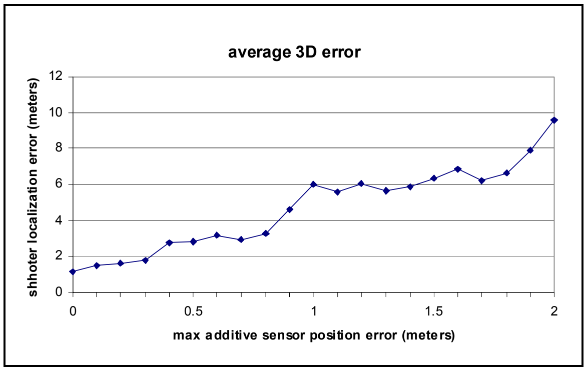

Figure 4. Sensitivity to sensor localization error

We performed similar experiments for measuring the effects of time synchronization error. The time synchronization error in the original test system was negligible (< 0.1ms). An additional 1-2ms of synchronization error caused insignificant increase of the localization error as shown in [6]. [6] also presents analysis of necessary sensor density and a comparison of the presented fusion algorithm data to the traditional TDOA-based approach.

The system was also tested with multiple concurrent gunshots. In different experiments up to four different shooters fired shots from different positions in rapid succession. In one test a machine gun was also used in full automatic mode. The system accurately localized all shooters in most such tests. The only exceptions were missed shots due to the fact that the numerous detections saturated the bandwidth of the radios. In addition to increasing the latency of the system, in some cases not enough detections were delivered to the base station for accurate localization due to increased packet loss.

4 Shockwave fusion

The muzzle blast fusion algorithm works very well when the acoustic source is located within the sensor field and there are at least 8-10 line-of-sight measurements. Once the shooter is located outside of the field the accuracy starts to decrease. One reason is that the angle of the sensor field from the shooter ("field of view") is getting smaller and hence, individual measurement errors have larger effect on the result. The other reason is that as the distance to the shooter increases, less and less sensors are able to detect the muzzle blast at all. Once the shooter is beyond 30-50 meters, muzzle blast alone is not enough to make accurate localization with our system.

Notice that the shockwave does not have this problem since it is generated by the projectile and not the gun. The sensor network is presumably deployed in and around the protected area. As long as the bullet flies in this area, there will be plenty of detections. For these reasons we have developed a shockwave fusion algorithm that is used for long range shots. In practical terms, when the number of muzzle blast detections are not enough (i.e. < 8) for accurate localization, the shockwave fusion outlined below is applied. The two different fusion approaches could be executed in parallel as well.

The shockwave caused by a supersonic projectile is the result of the air being greatly compressed at the tip of the bullet as it slices through the air. A broadening wave of the compressed air trails out diagonally from the tip and all the sides of the bullet as it moves forward, creating a conical waveform. See Figure 5. The shockwave front travels at the speed of sound while the bullet travels faster. The angle of the cone depends on the ratio of the speed of the bullet and the speed of sound:

$$\alpha = \arcsin\left(\frac{v_{sound}}{v_{bullet}}\right)$$

Note that this angle is constantly changing as the bullet decelerates and, in fact, the shockwave is not a cone. However, we assume constant bullet speed to simplify the necessary calculations. As it turns out, the overall accuracy of the system is hardly affected.

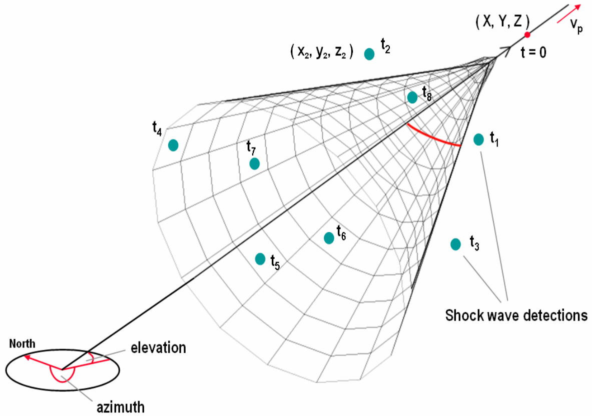

Figure 5. Shockwave front with TOA detections

Sensors can identify the shockwave based on its unique waveform and the high energy level. If enough sensors make detections and their positions and shockwave time of arrivals are known, the bullet trajectory can be computed. However, the same arguments about noisy and erroneous measurements and multiple simultaneous shots that were made in the previous section are equally valid here. Thus, we have applied a numerical approach for this fusion problem as well.

The consistency function based solution could be generalized and applied here. In fact, the muzzle blast and shockwave measurements could be considered at the same time forming a single consistency function. However, the number of dimensions would increase significantly. In addition to the three coordinates of the shooter position (x, y, z) and the shot time t, we have the two angles for azimuth and elevation and the speed of the projectile (see Figure 5). The search in this 7-dimesional space is currently computationally infeasible. Instead of the consistency function-based method, we have applied a genetic algorithm.

The real-time fusion algorithm receives shockwave measurements from the sensor network. First, the algorithm splits the received measurements into smaller groups representing single shots or possibly almost concurrent multiple shots. Then a genetic algorithm searches for the trajectories in each group that match the detections the best. Once the best trajectories have been found, a range estimation algorithm computes the estimated ranges using muzzle blast measurements if they are available.

4.1 Slicing

The slicing algorithm is responsible for splitting the incoming measurements into relatively small, processable groups. The online algorithm has three major states: idle, collecting and computing. When a new measurement is received, the system goes into the collecting state and the collection of the measurements begins. It stops collecting and goes into the processing state after a fixed amount of time elapsed form the receiving of the first message, or a fixed amount of time elapsed without receiving any more messages.

In the computing state, the first step is to split the shockwave measurements into smaller groups. Let D be the maximum of pairwise distances between sensors in the field. If there are two shockwave measurements which have a time difference larger than D / v_{sound}, then the two detections cannot be the result of a single shockwave, because the wave front travels at the speed of sound and the bullet is even faster.

The shockwave measurements are sorted by time and split into groups where there are time gaps larger than D / v_{sound}.

4.2 Trajectory search

Given n shockwave measurements:

$$s_1(x_1, y_1, z_1, t_1), s_2(x_2, y_2, z_2, t_2), \ldots, s_n(x_n, y_n, z_n, t_n)$$

where x_i, y_i, z_i are the coordinates of a sensor and t_i is the corresponding TOA of the shockwave, we are looking for m trajectories

$$tr_1(X_1, Y_1, Z_1, \alpha_1, \beta_1, v_1), \ldots, tr_m(X_m, Y_m, Z_m, \alpha_m, \beta_m, v_m)$$

that can generate the given measurements, where X_j, Y_j, Z_j is a point in space where the trajectory goes through, α_j is the azimuth, β_j is the elevation and v_j is the speed of the projectile. The X_j, Y_j, Z_j point of the trajectory is the point where the bullet was at time t_0 where

$$t_0 = \frac{1}{n} \sum_{i=1}^{n} t_i \, .$$

This is a somewhat arbitrary choice for a single point on the trajectory that is close to the sensor field.

For a given $tr_j(X_j, Y_j, Z_j, \alpha_j, \beta_j, v_j)$ trajectory and a $s_i(x_i, y_i, z_i, t_i)$ shockwave measurement, the theoretical time of arrival of the shockwave at the sensor (t'_i) can be calculated using simple Euclidean geometry.

From the measured and the theoretical time of arrivals a simple least squares error function of the trajectory is calculated. A genetic algorithm (GA) has been used to find the trajectory with the smallest error. Here is the brief description of the GA applied:

1. Generate an initial population of k random trajectories.
2. Select individuals from the population randomly.
3. Evaluate each individual in the selected subset using the error function (described later).
4. Sort the subset according to error.
5. Remove the worst 20% of the individuals in the subset. Generate new individuals by selecting random parents from the best 20% and applying genetic operators on the parents.
6. Go to 2.

Typical parameters used were k = 5000 and l = 500. A general problem with GA is that it can get stuck in a local minimum when the whole population becomes homogeneous containing the same exact individuals. Different heuristics have been proposed to avoid this problem. We use the tournament selection technique. We select a smaller subpopulation randomly at each step; this gives a chance for individuals with worse error value to breed also and makes the homogenization slower.

What happens if there are multiple shots or echoes? In this case we cannot use all the measurements, we need to identify those that are good and belong to the same shot. Therefore, we have extended the representation of a trajectory with a subset of the measurement indexes. This way the GA not only searches for the trajectory, but for a set of measurements as well, where the trajectory has the smallest error. Once the best trajectory is found, the corresponding events are removed from the group and the search is started again on the rest. It is repeated until the number of events becomes too small to be able to define a new solution. A trajectory has 6 parameters, therefore, at least 6 measurements are needed to find the trajectory.

The representation of a possible solution is the following: $sol(x, y, z, \alpha, \beta, v, S)$ where $x, y, z, \alpha, \beta, v$ is the trajectory parameters and S is a subset of numbers from 1...n (shockwave measurement indexes).

The following error function has been used to evaluate a given $sol(x, y, z, \alpha, \beta, v, S)$ solution:

$$error = \begin{array}{ll} \dfrac{1}{\left| i \in S \vee \left| t_i - t' \right| < T \right|} \sqrt{\sum_{i \in S \vee \left| t_i - t' \right| < T} (t_i - t'_i)^2} & |S| >= K \\ \infty & |S| < K \end{array}$$

If we have fewer than *K* measurements selected then the error is infinite. Otherwise, we sum the squared difference of all the measurements selected in the subset and the ones that has smaller difference than *T*. This second extension part is needed because otherwise the GA would just decrease the number of measurements in the set to *K* because leaving a measurement out always decreases the error. We set *K* to 7 after experimentation with field data. If we choose a small *T*, not all the measurements of the same shot will be selected. If we set it to too large, measurements from other shots can be included, increasing the error and distorting the result. If we know the localization error of the sensors and the typical shockwave detection error, we can estimate the largest measurement error in one shot. We have to set *T* to that value.

4.3 Range estimation

Once we know the trajectory and we have a few muzzle blast measurements, we can estimate the range to the source. Again, special care must be taken because of potential multipath measurements and multiple shots.

The trajectory gives us not only a line in space but also a timeline of the bullet, because we know the trajectory, the speed of the projectile and a point in both space (X, Y, Z) and time (t_0) of the bullet's location. As the position of the muzzle blast is also on the trajectory, we just have to correlate the trajectory and the muzzle blast TOA data.

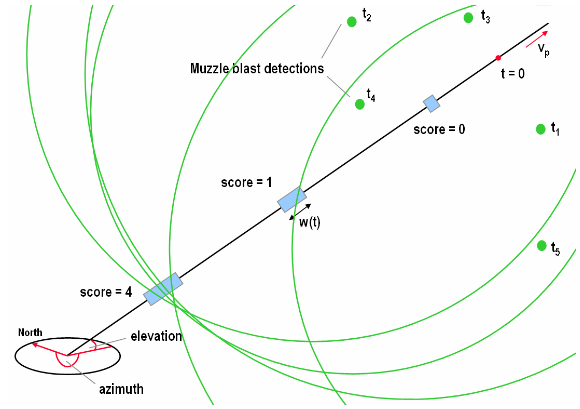

Figure 6. Range estimation utilizing muzzle blast TOA data

To compensate for erroneous detections and multiple sources, we just simply count the number of muzzle blast measurements that support a given point in space and time on the trajectory. We slide a small window backwards on the trajectory which gives us the possible space regions. The width of the window is determined by the estimated detection errors. A typical number is 1 meter. The time corresponding to the window location can be easily calculated form the speed of the bullet. A muzzle blast event is consistent with the current window if

$$d_{\min} < (t_m - t_w) v_{sound} < d_{\max}$$

where d_{min} (d_{max}) is the minimum (maximum) distance of the sensor from the window on the trajectory, t_m is the TOA measurement and t_w is the time corresponding to the current window location.

The window with the maximum number of consistent muzzle blast detections gives us the estimated origin of the shot and, hence, the range.

4.4 Evaluation

To test the accuracy of the shockwave fusion algorithm, we have conducted field experiments in a US Army facility. The sensor network of 60 motes covered an 80x80m area. The motes were placed on surveyed positions with an estimated accuracy of 20 cm. About half of the motes were on the ground, the rest were placed at elevations up to 6m. Shooter positions were also surveyed. They were located up to 100m from the edge of the sensor field. Two locations were in the basket of a cherry picker at approximately 13m height. Various targets were placed inside and outside the sensor field opposite from the shooter positions. 12 shots were fired over the middle of the network shooting approximately 100 meters from the edge of the sensor field, so there where sensors on each side of the trajectory. The average azimuth error was 0.66 degrees, the average elevation error was 0.61 degrees, and the average range error was 2.56 meters. Another 11 shots were fired from the same distance near the edge of the network, so there were no or only a few sensors on one side of the trajectory. The average error of these two combined shot groups increased to 1.41 degrees in azimuth, 1.11 degrees in elevation and 6.04 meters in range. The calculation of a single shot trajectory took about 3-4 seconds on a 3GHz PC.

Multiple simultaneous shots have also been tested with mixed results. The typical test involved two shots only. About half the time the system correctly localized both trajectories. There were cases, however, when three trajectories were found. Two of these were typically very close to each other and one of the true trajectories. This error can happen when there is more than double the number of detections needed for localizing a single trajectory. In such a case, the error value corresponding to a subset of the detections may be smaller than the one involving all the detections for one trajectory. The error function and the genetic algorithm need to be adjusted to avoid this situation.

Finally, the problem we refer to as "trajectory inversion" was also observed numerous times. Consider the two trajectories shown in Figure 7. One side of the generated shockwaves is almost the same. If we have sensors such that only this side is being measured, the wrong trajectory can be identified. In fact, the only way to differentiate the two possible sources of the half shockwave front is to measure its curvature caused by the deceleration of the projectile. However, that would mandate highly accurate sensor placement and very precise time synchronization. This can only be achieved by a microphone array with fixed and known microphone separation.

The area of correctly localizing multiple concurrent long range shots based on shockwave measurements needs further research. Unfortunately, we only had the opportunity to conduct a single field test. Our project ended in December 2004 and we have not worked on improving the presented techniques since.

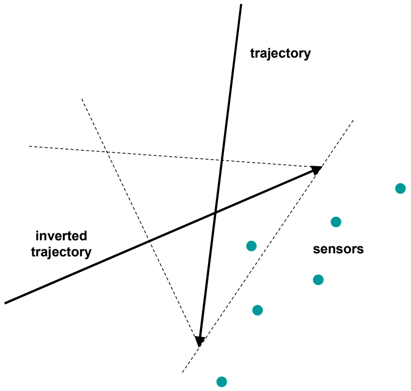

Figure 7. Trajectory inversion

However, to test the sensitivity of the fusion algorithm to sensor location error, we have conducted an additional laboratory experiment using the real TOA data corresponding to the 12 shots fired in the middle of the network described previously. We added additional errors to the sensor location data used during the field test. For each shot and for each coordinate (x, y, z) of each sensor position, a random error was added using a uniform distribution between 0 and d_{max}.

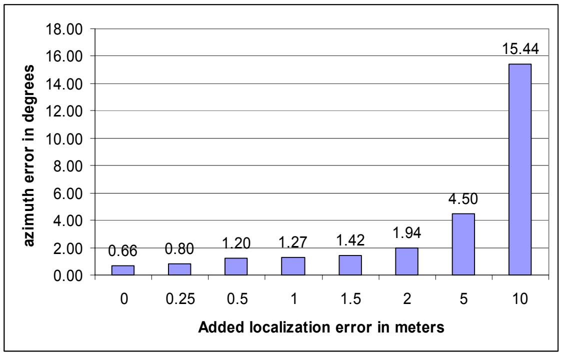

Figure 8. Sensitivity to sensor location error

The results for azimuth accuracy are summarized in Figure 8. For each maximum additional sensor location error, each shot was tested ten times using different sensor locations. Therefore, each bar in the figure represents 120 experiments. It is somewhat counterintuitive that an additional 1 meter sensor location error only increases the azimuth error to 1.27 degrees. We expect, however, that the effect on multiple shot localization would be more significant. Nevertheless, the apparent insensitivity of the data fusion to sensor location error is encouraging for the future development of the system.

5 Conclusions

The performance of the presented sensor network-based countersniper system is on par with existing centralized systems in regular settings. However, it outperforms other systems in urban environments because it can mitigate multipath effects and resolve multiple concurrent shots. These attributes constitute the primary novel contribution of the presented work. These capabilities are due to system's widely distributed sensing and above all, the novel sensor fusion approach. To our knowledge, the muzzle blast fusion technique is unique in its ability to reject echoes and discriminate multiple simultaneous sources.

Furthermore, the shockwave fusion approach can also successfully eliminate echoes; while the multiple shot resolution capability of the first prototype is marginal. However, the accuracy of better than 1 degree in both azimuth and elevation for single shots is also unmatched by other systems.

Acknowledgement

We would like to gratefully acknowledge the sponsorship of the DARPA/IXO NEST program. We thank the anonymous reviewers for their constructive criticism.

References

[1] Chen, J. C., et al. 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., et al. Coherent acoustic array processing and localization on wireless sensor networks. *Proceedings of the IEEE*, vol. 91, Aug. 2003, pp. 1154—1162.

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

[5] Mahajan, A. et al. 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, Feb. 2001, pp. 91-94

[6] Simon, G., et al.: Sensor Network-based Countersniper System. *In Proc of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 04),* November 2004.

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

[8] 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

[9] 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.

[10] Kusy, B. et al. Elapsed Time on Arrival: A simple, versatile, and scalable primitive for canonical time synchronization services, accepted for publication in *Int. J. of Ad Hoc and Ubiquitous Computing*, 2005

[11] Maroti M., Kusy B., Simon G., Ledeczi A.: The Flooding Time Synchronization Protocol, in *Proc of ACM Second International Conference on Embedded Networked Sensor Systems (SenSys 04)*, November 2004.

[12] Maroti M.: Directed Flood-Routing Framework for Wireless Sensor Networks, In Proc. of *Middleware 2004*, October, 2004

[13] Wang, H. et al.: Target Classification and Localization in Habitat Monitoring, *In Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003),* April 2003