Acoustic Source Localization in a Heterogeneous Network

Tingyu Thomas Lin

Abstract

I present a potential design of a heterogeneous acoustic sensing network comprised of the Acoustic Embedded Networked Sensing Box (ENSBox) and the Berkeley Mica2 platforms. I also present one potential method of leveraging the presence of Mica2 motes to improve acoustic source localization. More specifically, I present a method of rejecting grating lobes, which may be present in direction-of-arrival (DOA) estimates created by ENSBoxes, using information available from the additional Mica2 motes.

1. Introduction

The ENSBox platform is a self-calibrating distributed acoustic sensing platform. It can accurately localize each node inside the ENSBox network with an average 2D positional error of 5cm and an average orientation error of 1.5 degrees (1). It has the facilities to localize other acoustic sources, and has been successfully deployed to detect and localize marmot calls (2). It also has time synchronization and wireless network services to facilitate acoustic localization.

Though the ENSBox platform as-is can already perform acoustic source localization, using additional motes can augment the acoustic sensing capabilities of the platform. While trying to localize acoustic sources, the ENSBox platform frequently encounters ambiguity in the recorded data. One solution to this is to place more ENSBoxes in the field. These added nodes provide additional information for the network, reducing potential ambiguity. However, deploying many ENSBox nodes is prohibitively costly, both in the sense that each ENSBox is monetarily costly and in the sense that to reduce ambiguity the complete functionality of an ENSBox is not completely needed.

Therefore, it would be beneficial if it was possible to deploy cheaper nodes, namely Mica2 motes. Mica2 motes are much more cost effective to deploy in large numbers. Mica2 motes are heavily resource constrained, as well as having relatively low resolution in all measurements they make when compared to the ENSBoxes (Sallai et. al. (3) have implemented an acoustic sensing platform on a Mica2 mote network that achieved acoustic localization accuracy with a standard error of 10+ cm, at least a magnitude greater than what the ENSBox can provide). But while not being able provide acoustic localization to the degree of accuracy that the ENSBox provides, Mica2 motes equipped with a microphone and speaker (a sensor board with these two pieces of hardware is available from the manufacture of Mica2 motes) can act as event detectors for the ENSBoxes. To further motivate the use of Mica2 motes with ENSBoxes, recorded information at the Mica2 motes can be offloaded onto the more capable ENSBoxes for computation.

There are probably many improvements in acoustic localization that can be made to the ENSBox platform now that Mica2 motes can be utilized as event detectors. However, I will only be focusing on one, which I call "lobe rejection." See section 3 for details.

2. Integrating Mica2 Motes into the ENSBox Platform

There are three things needed to integrate Mica2 Motes into the ENSBox platform:

- Wireless Communication between the two networks
- Time synchronization in the mote network
- Localizing Mica2 Motes

2.1. Wireless Communication

Communication between motes and between motes and ENSBoxes is necessary for the motes to exchange information to the ENSBoxes. Built into the ENSBoxes are mote radios that operate at the same frequency as Mica2 motes. Also, the ENSBox provides libraries that greatly simplify operating the radio.

The Mica2 motes themselves do provide a MAC layer protocol, so for now both in simulations and in the bits of actual implementation that I have broadcast packets and assume that everything is single-hop and has no collisions. However, I have not looked into nor designed any particular routing protocol to facilitate communication within the mote network as well as in between the two networks. A mote network is required to perform time synchronization (see section 2.2), so it is likely the ad hoc network needed by the time synchronization protocol FTSP will also be used to route data packets through the mote network.

2.2. Time Synchronization

Time synchronization is critical to acoustic source localization. When a mote detects an event, it must be able to know when it detects the event relative to other motes' event detection times. Given that the difference in times of detection at different motes will be incredibly small due to the speed of sound, the accuracy of time synchronization must be in the order of microseconds.

The time synchronization mechanism that the ENSBox fundamentally uses is the Reference Broadcast Synchronization (RBS) (4). In RBS, a radio broadcast is made and all nodes in range record the time the packet was received. The difference then between each of the receive nodes' reception time is the clock conversion between them. Using this mechanism, the ENSBox platform achieves time synchronization on the order of tens of microseconds across multiple hops (1). However, RBS is very "chatty," requiring quite a few message exchanges between nodes. This is not an issue in ENSBox networks, where there are only 6 to 10 ENSBoxes, but for large mote networks the amount of radio traffic needed is prohibitively large (5).

A better alternative is the Flooding Time Synchronization Protocol (FTSP), proposed by Elson et. al. (5), and is the time synchronization mechanism I propose to be used. FTSP was designed and implemented on the Mica2 platform, making it an ideal candidate for time synchronization application

here. In FTSP, one node is elected the root, and its clock is considered to be the reference for global time. The root then sends out a packet with the timestamp of the global time of transmission. At the receiving nodes' ends, the nodes compare the local reception time with the global transmission time. This difference determines the clock differences between motes. Elson et. al. implemented FTSP on a 60-node network and achieved an average per-hop error of about 3 microseconds and a maximum error of 14 microseconds. When translating these errors into error of how far sound traveled between motes, the average error is about .1cm and the maximum error is about .5cm.

I have not implemented this, only simulated the errors in time sync for simulations done in matlab.

Tight time synchronization between the ENSBoxes and Mica2 motes is not actually necessary for the lobe rejection mechanism. However, it is conceivable that the FTSP be expanded to include the ENSBoxes in the time synchronization. The mote radio on an ENSBox is actually a Mica2 mote tethered to the ENSBox. The tethered mote runs a special program that passes any packets it receives up to the ENSBox as well as sends any packets the ENSBox passes down. For not-so-tight time synchronization, the root mote in FTSP can send its reference global time to the ENSBox. There will be a delay as the ENSBox's tethered mote receives, processes and passes up the packet. It is not tested what this delay will be. For tighter time synchronization, the program running on the tethered mote can be modified to perform FTSP time synchronization without communicating up to the ENSBox just for these packets. This option has not been explored beyond this thought.

2.3. Localizing Mica2 Motes

Event detection requires knowing where the event occurred. Therefore, we need to be able to give positions to each of the Mica2 motes. To achieve this, the Mica2 motes are equipped with a speaker that the motes can chirp from. The ENSBoxes possess the ability to localize acoustic sources, so leveraging this ability the ENSBoxes can localize on any given mote's chirp. The approach to localizing sound sources in the field is in a large part the same method that Ali et. al. approached localizing marmots calls in (2), but there is one key difference that localizing Mica2 motes has. To be more specific, it is the ability to control the properties of the signal (e.g. frequency and length of call). This will have impacts in the accuracy of the source localization.

In (2), Ali et. al. propose three possible solutions for localizing acoustic sources; differential signal amplitudes, time difference of arrivals (TDOA) and comparison of direction of arrival (DOA) estimates. The method of differential signal amplitudes was rejected because environmental factors, such as obstructions and sound reflections, will affect the signal amplitudes, making them unreliable. The second method was also rejected on the basis that TDOAs are hard to determine. In their case, the animal calls that they are trying to track do not lend themselves easily to determining where the start of a sound is, especially in the presence of background noise. Therefore, they elected to use DOA estimates to determine the location of marmots.

My application is localizing motes, not marmot calls, but I face the same three options, and I reject the first 2 solutions for the same reason. I have one additional option that they did not have

however, and it is the time of flight estimate. Since the time that a mote chooses to chirp at is controllable, especially with tight time synchronization, it is possible to determine the time that it took for the sound wave to travel from the mote's speaker to the ENSBoxes' microphone array. This method suffers from the same flaw that the TDOA method has where it is difficult to determine when a sound started. Though the exact signal pattern emitted from a mote can be controlled, this still does not mitigate the problem of background noise interfering with determining the start of a sound.

2.3.1. DOA-Based Localization

DIAGRAM/PICTURE OF THE FOUR MIC ARRAY

To determine the direction of arrival, the ENSBox has a four-microphone sub-array. For simplicity, I am restricting the case to 2D localization (that is, I assume the source and the array are in the same plane), though the ENSBox is capable of 3D localization. From the top view of the ENSBox, the array forms a square with sides of XXX cm (8,12,15?). Software developed for (2) running on the ENSBox allows for continuous recording and detection of events specific frequency ranges, using their Constant False-Alarm Rate (CFAR) algorithm. After an event is detected (that is, when the ENSBox's array detects a sound in a specific frequency range), the Approximate Maximum Likelihood (AML) algorithm is ran on the relevant sound recordings for the duration of that event. What the AML algorithm produces is a bearing estimate from the ENSBox where the sound most likely came from.

Determining the AML bearing estimate relies on the geometry of the sub-array as well as the signal that is received. When a sound wave passes over the sub-array, the wave will arrive at each individual microphone at different times. Therefore, the wave recorded at each microphone will have a phase shift. The phase shift is a function of the direction that the sound arrived from. This phase shift can be determined, and from which the direction of arrival can be determined. This is actually done not by determining a single direction value, but a polar plot of likelihood values (which I call the AML likelihood vector). However, there is a limitation on how high a frequency can be. Quoting (2):

> "In order to measure the phase of an incoming signal by comparison from two points in space, those two points must lie in the same half-wave. Energy in frequencies with wavelengths shorter than 2x the sensor spacing will be aliased into lower frequencies. This implies that for a sensor spacing of D and signal propagation speed V_s, the maximum frequency detectable without aliasing is $F_c = V_s/(2D)$."

SHOW THE SIMULATED BEAM PATTERNS FOR A CASE OF ALIASING AND A CASE OF NOT ALIASING. I believe 1khz call, 15cm/8cm array will/won't alias. Same with a fixed 8cm array, a 4khz/1khz call. Both at 45 degrees orientation.

As a consequence, as the frequency of the observed sound increases, the expectation is that the likelihood polar plot will start aliasing. Also, as the sensor spacing becomes larger, the lower the threshold for aliasing is, producing more aliasing at lower frequencies. In either case, as the signal starts aliasing, grating lobes (aka false lobes, the ones that aren't pointing in the right direction) will start to appear. As aliasing gets worse, the likelihood value of the grating lobes approaches the value of the true lobe.

A solution to this is to simply have the array size be small; this will prevent any aliasing. However, a trade off for this is that the lobes become wider. This yields more ambiguity in the general region that the sound could have come from. In (2) a balance was struck between the two with 8 cm spacing.

SHOW PIC OF SMALL ARRAY SIZE (2 cm?) AT 1 kHz CALL

2.3.2. Localizing Motes with DOA-Based Localization

Regardless of actual array geometry, what is important for localizing motes is the critical frequency. At the critical frequency is where we will get the least aliasing with the narrowest lobe possible. Since the frequency of call is controllable with the motes, the critical frequency should produce the least ambiguous AML likelihood vector.

2.3.3. Pseudo-Likelihood Maps

After a particular event, each ENSBox will have an AML likelihood vector. These vectors can then be fused together to produce a pseudo-likelihood map. The location of the sound source and therefore the mote is most likely at the intersection of the DOAs.

(Note: I do not have the pseudo-likelihood map code, beyond the aml_fuse_plot.c that mike made. I need to convert that into mat lab and try it) More details in the paper (2).

3. Lobe Rejection

The ability to determine which lobes are false will significantly reduce the ambiguity in AML likelihood vectors. This problem can be lessened by making the array geometry smaller, but not without the trade-off of having lower resolution lobes (i.e. wider lobes). Also, changing array geometry can only be done before the ENSBoxes are made; after manufacturing, the array size is fixed. But even with small distances between microphones in the array, localizing really high frequencies, e.g. 6000+ kHz, will still produce a multitude of false lobes.

3.1. Current Lobe Rejection Method

The current method of lobe rejection, done by (2), is with the pseudo-likelihood maps. If there are few (2, maybe 3) false lobes at each ENSBox, the chances of these lobes intersecting each other at a single point is relatively small. This works for most cases, but becomes increasingly unsuitable as the number of side lobes increase.

SHOW PIC OF 10000 KHz call, it's got like 16 or so lobes of comparable size. Very cool.

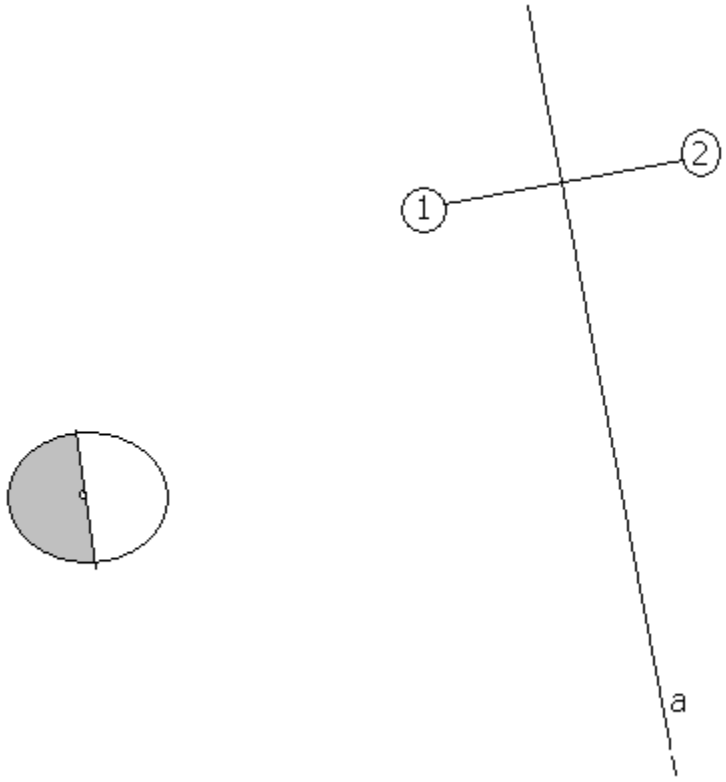
3.2. Proposed Lobe Rejection Mechanism

I propose an alternative mechanism for rejecting false lobes by utilizing the information available from the Mica2 motes.

Let there be a field where several ENSBoxes and many motes are in. The initial self-calibration of the system is performed, so all ENSBoxes and motes know their positions. A sound (say, a bird's call) goes off somewhere in the field. Due to the tight time synchronization, the motes will have a known

order in which they detected the sound in. Also, each ENSBox will have an AML likelihood vector with many false lobes.

The order in which motes detected an event is sufficient in eliminating certain lobes.



In this figure, the numbered circles, 1 and 2, are motes. The larger circle on the left is an ENSBox. Line a is perpendicular to the line that goes from 1 to 2. If mote 2 detected the event before mote 1, then the sound source could not have been to the left of line a. From the ENSBox's perspective, this translates to that the bearing of the sound could not have come from the shaded half of the ENSBox. This then is sufficient to reject half of the ENSBox's bearing estimate possibilities and any lobes in that region. This can be iterated between all mote pairs (there are an order of n^2 pairs, where n is the number of motes). This should eliminate most of the false lobes.

There are several cases where we should not reject any portion of the ENSBox's bearing estimates. The first one is that of mote 1 was the first to detect the event, we should not reject anything, as the sound origin can be anywhere around the ENSBox. Second is if the difference in detection time between the two motes is very close. If the difference is less than the maximum error of time synchronization, then there is a possibility that the current ordering of one before the other is a product of errors. Also, there is the issue of time to detect that an event has taken place. There may be a non-deterministic delay in the processing time needed at a mote to recognize the event. Therefore, in

addition to the error in time synchronization, a bit more time padding needs to be added to allow for time differences in processing time.

Also, in this example I reject 180 degrees at a time. In reality, we cannot be as certain. There are positional errors in the motes and ENSBoxes, as well as orientation errors in the ENSBoxes. Therefore, we may be rejecting more than we mean to. Instead of rejecting the 180 degrees, a smaller portion should be rejected.

3.3. Consequence of Lobe Rejection

If a significant number of false lobes can be eliminated, this implies that the sensor spacing in the array can be made larger to provide better resolution. The cost of this better resolution is more aliasing, but the proposed lobe rejection mechanism should negate this cost.

(No work done to prove this yet)

4. Approach to Simulation

In the works. I'm still in the middle of simulating the lobe elimination.

5. Simulation Results

In the works.

6. Conclusion and Future Work

In the works.

Acknowledgements

Thanks to everyone who helped.

Works Cited

1. **Girod, Lewis, et al.** The Design and Implementation of a Self-Calibrating Distributed Acoustic Sensing Platform. November 1-3, 2006.

2. **Ali, Andreas M., et al.** An Empirical Study of Collaborative Acoustic Source Localization. *IPSN'07.* April 25-27, 2007.

3. **Sallai, Janos, et al.** Acoustic Ranging in Resource-Constrained Sensor Networks. *ICWN '04.* June 2004.

4. **Elson, Jeremy, Girod, Lewis and Estrin, Deborah.** Fine-Grained Network Time Synchronization using Reference Broadcasts. *OSDI 2002.* December 2002.

5. **Maroti, Miklos, et al.** The Flooding Time Synchronization Protocol. *SenSys '04.* November 3-5, 2004.