Passive acoustic sound source tracking in 3D using distributed microphone arrays

Passive Acoustic Sound Source Tracking in 3D Using Distributed Microphone Arrays

Erik Verreycken*‡, Walter Daems*‡, Jan Steckel*‡
erik.verreycken@uantwerpen.be
*FTI CoSys Lab, University of Antwerp, Belgium
‡Flanders Make Strategic Research Centre, Belgium

***Abstract*—In this paper we describe a system for passive acoustic sound source localization in 3D using a distributed microphone array. By distributing the microphones into small-scale and large-scale arrays we can exploit array processing algorithms that use near-field and far-field properties of the sound source with respect to the array. The microphones on the small-scale device are sampled using a single, simultaneously sampling analog to digital converter which results in time synchronization between these microphone channels up to a small fraction of the signal's phase. The large-scale array consists of multiple small-scale devices and can be deployed over a much larger capture volume. We create the distributed microphone arrays in a manner that is scale-free with regard to the amount of microphones in the system, the capture volume, the type of sound and the type of sensors/microphones. In this paper we describe the hardware topology, the localization algorithms and the obtained results.**

Keywords—passive acoustic localization; microphone array; scale-free; wireless sensor network; bio-acoustic localization

I. Introduction

Passive acoustic localization has many benefits in biological research as opposed to active tracking methods which typically require capturing the animal for attaching some kind of marker or device which may be very disturbing to the animal. A passive approach to animal localization by their vocalizations can be done using a microphone array [1]–[4]. By observing the difference in arrival time of the subject's vocalization, an estimate about the position of the sound source can be made [5]–[7]. In our distributed microphone array we make the distinction between large- and small-scale arrays. A small-scale array is defined as a single device containing a limited amount of microphones spaced relatively close together (in the order of centimeters). Due the microphones being spaced closely together, most sound sources will originate in the far-field of that small scale array. Therefore, due to the planar nature of the impinging sound waves this type of array can only provide information about the direction of arrival [8], [9]. A large-scale array consists of multiple smaller devices encompassing a larger area thus placing the sound source in the near-field of the array aperture. From a sound source in the near-field we can find the 3D position by making use of the spherically-shaped wavefronts [10]. Acoustic localization using microphone arrays can usually be divided in two main areas: (1) acoustic imaging using small-scale arrays within a single capture device where the result is often an acoustic overlay of sound intensity over an image or video [11], [12] (2) 2D or 3D localization using large-scale arrays [13], [14] e.g. by placing microphones along the runway on an airport, the location of a plane on the runway can be detected.

Small-scale nodes can be manufactured as a single device with exact specifications such as microphone locations on a Printed Circuit Board (PCB) using a single Analog to Digital Converter (ADC). Due to the fact that a synchronous sampling ADC can be used, there are less uncertainties in the time differences between the signals of the microphones which in turn reduces the uncertainty in the estimated quantity of the device. This is Angle of Arrival (AoA) in the case of a small-scale node. Large-scale arrays suffer from time synchronization errors [15], [16] and higher uncertainties in the microphone locations as they consist of multiple nodes and can be deployed over much larger areas. By constructing a large-scale array consisting of small-scale devices we can create a distributed Wireless Sensor Network (WSN) that is scale-free in terms of capture area, amount of microphones, amount of nodes and type of sensor. E.g. different microphones using different sample rates or even different types of sensors such as 3D camera's can be combined due to the use of a probabilistic localization algorithm and the usage of a synchronization mechanism.

The goal of this paper is to describe a complete system for bio-acoustic localization using a distributed microphone array resulting in a scale-free WSN. In section II we describe in more detail how the small- and large-scale arrays can be constructed and what considerations there may be for the topology with special consideration for microphone positions and data synchronization between the microphones. We show that it is possible to robustly time-synchronize nodes in a WSN. We also describe how the synchronization method allows for various data sources to be synchronized into a single context. In section III we explain our localization algorithms for both the small-scale array where we can identify the direction or angle of the incoming sound, and the large-scale array where we can identify the sound source location in 3D. We also indicate measures for reducing the computational complexity of the localization algorithm. Next, in section IV we show the results of our system from both lab experiments where we tracked the location of an ultrasonic transducer in a lab environment following a trajectory similar to that of a free-flying trawling bat. Finally, in section V we give our conclusions to this work and indicate future research.

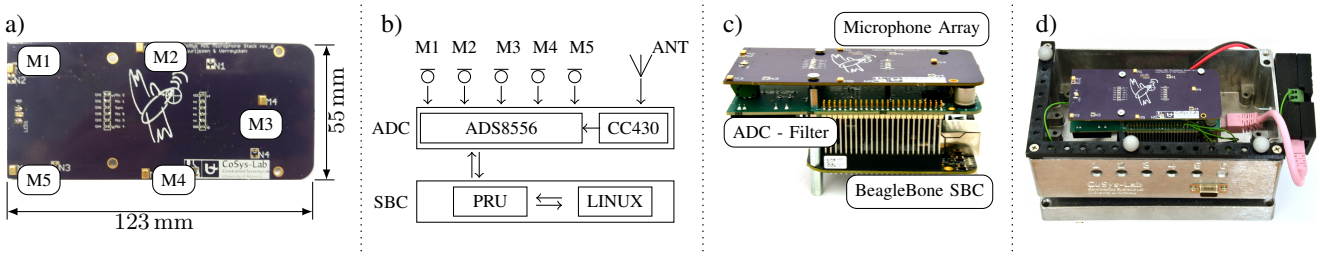

Fig. 1. **Panel a):** The small-scale microphone array consisting of five MEMS-microphones soldered in an elliptical pattern on a PCB. The component count on the top layer of the PCB is kept to a minimum in an attempt to keep the acoustic profile of the board close to a flat surface. **Panel b):** Overview of the hardware stack. At the top level there are five microphones on the PCB. The microphones are sampled by a single simultaneously sampling ADC. An optional antenna and CC430 module can be used for wireless synchronization between nodes in a sixth channel. Low level interfacing with the ADC is done using a PRU to ensure real-time consistency in the data. Higher level interfacing such as network interfacing for data offloading is done using a BeagleBone Black Single-Board Computer (SBC). **Panel c):** Complete hardware stack of a single small-scale node showing the three layers of the node. At the bottom (black) there is a BeagleBone Black SBC with Programmable Real-time Unit (PRU), On the middle layer (green) there is a PCB with the 6-channel ADC and a hardware band-pass filter, on top (purple) the microphone array can be found. **Panel d):** The node in an enclosure to protect it during transport. The enclosure allows for easy mounting on a tripod or similar mounting system.

II. Acoustic Localization - System Topology

We have created hardware and software that allow the construction of microphone arrays for 3D acoustic localization. At the small-scale level we consider a single node consisting of five closely spaced microphones, as can be seen in figure 1. As the microphones are spaced closely together, any detected sound will be in the far-field which means we can only extract AoA information. We provide more insights in the underlying reasoning for this in section III-B. By distributing multiple small-scale arrays over a larger capture area we create a large-scale array where the sound is in the near-field of the array, see figure 2. Using the large-scale array we can use a probabilistic Time Difference of Arrival (TDoA) approach to locate the sound source in 3D.

Considering that TDoA localization relies on detecting offsets in the arrival times of a single sound at different nodes or microphones, there are two key factors that determine the accuracy and precision of our localization algorithms:

1) Degree of synchronization: As the absolute emission time of the sound is not known, we can only use relative differences in the arrival times of the sound for our localization algorithms. We also have to know the relative (or absolute) start and stop times of the each of the recordings which requires our nodes to have a synchronized timing reference. Any synchronization errors will appear to the system as a timing offset in the relative arrival times, hence resulting in possibly severe errors on the final source position estimates. More information on synchronization can be found in section II-C.

2) Microphone position: Considering that TDoA works by comparing arrival times based on the location of the microphones, we can surely state that inaccuracies in the microphone locations will lead to more uncertainties in the localization algorithm. This can be explained due to the fact that an erroneous microphone position results in an inconsistent measurement which in turn will lead to a lower, more spread out probability distribution for the source position. More information on the probabilistic source position estimation algorithm can be found in section III.

A. Small-Scale Arrays

A single node contains a top level PCB with five closely spaced microphones. The inter-microphone distance lies between $4\,\mathrm{cm}$ and $11\,\mathrm{cm}$, as can be seen in figure 1a. These microphones are soldered to a predestined pad on the PCB which means their location is known to a very high degree of accuracy. The microphones are sampled using a single simultaneously sampling ADC to ensure a very high degree of time-synchronization between the microphones. As shown in figure 1b an optional antenna can be connected for wireless synchronization between different nodes. The synchronization channel is then recorded as a sixth channel by the ADC [17]. More information on the synchronization method for the sensor array can be found in section II-C.

Low-level interfacing with the ADC must be done at a precise rate using a real-time processor to ensure data is captured at a correct sample rate. In our system, we selected a BeagleBone Black (BBB) Single Board Computer (SBC) as this device uses a Sitara AM335x processor. This processor has a real-time co-processor, or Programmable Real-time Unit (PRU), which is a Reduced Instruction Set Computer (RISC) processor running at $200\,\mathrm{MHz}$ capable of single instruction IO and Direct Memory Access (DMA), programmed in assembly language. The PRU executes commands at a fixed rate without interrupts which guarantees real-time code execution at a sufficient data rate to deal with the relatively high data throughput:

$$285\,\mathrm{kHz} \cdot 16\,\mathrm{bit} \cdot 6\,\mathrm{channels} \approx 3.3\,\mathrm{MiBps}$$

Higher level interfacing, such as data transfer over User Datagram Protocol (UDP), is handled by a small SBC running GNU/Linux. Some processors such as the Sitara AM335x combine a PRU and a general purpose processor running GNU/Linux in a single chip solution. In the case of the Sitara processor both PRU and general purpose processor share access to the same memory which can be accessed through DMA.

The PRU of the BBB controls a six-channel simultaneously sampling ADC from Texas Instruments (ADS8556) capable of

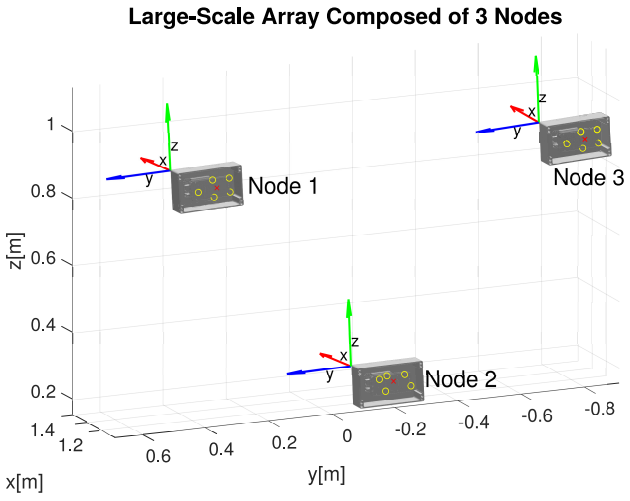

Fig. 2. Example of a large-scale array composed of three nodes of nearly identical orientation. The relative 3D rotation for each node with regard to the global coordinate system is indicated as a 3-axis coordinate system. Each node has five microphones indicated with a yellow *o*. The red *X* indicates the center of the microphones or the virtual microphone. Node 1 and node 3 have the same microphone layout, node 2 differs slightly. This layout could be used for the localization of a bat flying towards the nodes. The locations and orientations of the nodes have been gathered using the Qualisys system, see also section IV.

sampling up to 630 kHz. Data is fed to the ADC by a third order Sallen-Key low-pass sampling filter, the ADC and filter reside on a separate board as can be seen on figure 1c. On top of the ADC board, a PCB with the microphone array is mounted.

B. Large-Scale Arrays

By distributing multiple small-scale nodes over a larger area we can expand the capture volume of our array. This is what we designate as a large-scale array, see figure 2 for an illustration of a large-scale array constructed using three small-scale nodes. By varying the configuration in terms of amount of small-scale nodes, their position and their orientation we can control the size of the capture volume and the accuracy within that volume. As the array can span an entire room or part of a forest, the acoustic source can be considered to be in the near-field of the large-scale array and as a result, we can locate the acoustic source in 3D given that the large-scale array is 1) spread over a large enough area to place the sound source in the near field, and 2) the large-scale area has enough dimensional diversity in the node locations to eliminate any ambiguities in the sound source location. The system described here is scale-free with regard to the amount of nodes that can simultaneously be used for localization. As such, the large-scale array is also scale-free in terms of the size of the capture volume and the accuracy withing that capture volume. For example, when tracking the foraging routes of Proboscis monkeys, the necessary capture volume of the system may well exceed a couple of square kilometers while precision within 2 m may be acceptable as this leaves ample room for identifying the location of the animal to a single tree. On the other hand, when analyzing the flightpath of a bat approaching prey as in [1], [4], the system must be able to locate the animal much more accurately, however the capture area can be kept much smaller because such experiments are typically done in a flight cage or above a small portion of a lake. From these examples we can make the following requirement for the system: the large-scale array must be able to support any amount of nodes in any size of capture volume.

As the TDoA algorithm depends on the combination of data from multiple nodes, the calculations must be done offline on a central processing point such as a laptop. The large-scale network requires a medium for sending commands to the nodes and transporting the recorded data from the nodes to the processing point. For large-scale networks that are restricted in size (e.g. networks that are confined to a single room), this medium could be WiFi or a cabled network connection. Most SBCs provide interfaces for connecting to standard WiFi or ethernet. For larger setups, where a single access-point is beyond the wireless range of a standard WiFi connection, or where a wiring all nodes to a central hub or switch is not feasible, a multi-hop protocol is needed. Many references about the creation of multi-hop wireless networks for the intended purpose of creating a WSN can be found in the literature [10], [18] as well as solutions for a wired multi-hop protocol. The wireless solution can have it's merits if the whole setup is made wireless and battery powered, however we will restrict ourselves to wired experiments in the near future so we created a Power over Ethernet (PoE) solution where the power, synchronization and network data are transported over a single cable. A small secondary device attached to the node is able to split the signal from the cable and feed them to the node. This device could have an internal ethernet switch to allow daisy chaining of the devices thus supporting multi-hop communication.

C. Synchronization

In our large-scale microphone array, microphones are not simultaneously sampled. As a result, timing offsets will occur in the recorded data. Indeed as we use non real-time network protocols such as UDP over a wired, or even wireless network to perform synchronization, small time differences will exist in the capture start and stop times. Network timing protocols such as Network Time Protocol (NTP) can achieve timing offsets below 1 ms under ideal circumstances [16]. These timing errors may be even larger for multi-hop networks. As we use a timing sensitive TDoA algorithm for localization, it is quintessential that we are able to accurately synchronize our recordings. Indeed, an error of 1 ms results in a distance of 34 cm, given that the speed of sound in air is 343 m/s. The actual error of our TDoA algorithm will change significantly with the changed TDoA measurement due to the fact that the sound is located on a hyperbolic surface which is the solution of the standard TDoA approach.

Synchronization of recorded data is usually done by adding a time stamp to the recorded data. Accuracy of the time stamp is not trivial as this requires a high degree of synchronization

between the nodes. Different methods for synchronizing clocks in a WSN can be found in the literature [19]. Some methods such as NTP, Mock's algorithm [20] or Delay Measurement Time Synchronization (DMTS) [21], rely on a master node for synchronization and are therefore less robust against the failure of the master node. Furthermore synchronization errors increase with the amount of hops in the case of a multihop WSN. A more robust approach would be to utilize a peer-to-peer synchronization method where all nodes contribute equally. These methods are more robust against the failure of any single node and do not suffer from multi-hop delays. Peer-to-peer synchronization allows for a more flexible and more scale-free deployment of nodes in the WSN. Examples of masterless synchronization can be found in the peer-to-peer version of NTP or a separate protocol such as Reference-Broadcast Synchronization (RBS) [22], or Time-diffusion Synchronization Protocol (TDP) [23]. Furthermore, the time stamp needs to be added very carefully in the metadata of the capture and include information such as pre- and post-trigger data. This information can easily be lost if the data is truncated or otherwise manipulated.

Other methods of synchronization such as the pulse per second (PPS) output from a GPS receiver [24] may be used for accurate slot-synchronization. However, in a distributed system such as described here there is a need for time- or absolute-synchronization as slot synchronization alone may still yield ambiguities in the post processing of the data. Furthermore, the usage of GPS may not be feasible in an indoor setting.

In [17] we describe a biologically inspired, distributed synchronization mechanism using pulse-coupled oscillators for clock synchronization in a WSN. By embedding the synchronization data in the recorded data, we can robustly add timing information to a recording. More details on such a synchronization mechanism can be found in [25] where we embed a 1-bit signal with pseudo-randomly chosen duty cycles in the recorded data. As the random signal exhibits a very narrow autocorrelation function we can use it to find timing offsets between multiple streams of recorded data. As shown in figure 3, we can use the cross-correlation product of two synchronization channels, a sharp peak will then identify the timing offset between the recordings. Selecting fragments of a stream does not hamper the cross-correlation based synchronization.

Each of our small-scale nodes can sample up to six channels simultaneously. One of the channels is sacrificed for the recording of the synchronization channel, leaving five channels for acoustic data. After the capture, recordings are transferred to a central processing point (e.g. a laptop) where the synchronization channel is used to find timing offsets between all recordings. This method also enables synchronization of other sources with the acoustic data. An infrared LED on the PCB, connected to the same synchronization channel, will emit a visual signal with the same pseudo-random structure as the recorded data. By extracting the pseudo-random signal from a camera recording, we can correlate the acoustic recordings with a high-speed camera. Using formula 1 from [25] we can find an optimal operating point for the random signal:

$$K = \frac{2 \cdot L \cdot T_S}{P_{min}(\frac{P_{max}}{P_{min}} + 1)} \tag{1}$$

with K the estimated number of transitions in each capture, K should be kept above 10 for a strong auto-correlation function. With L being the smallest fragment length in number of samples that can still be used for synchronization, T_S representing the largest sample period in seconds, this corresponds to the lowest sample rate in the system, with P_{min} the minimal period and P_{max} the maximal period of the random sequence. In our setup, we record acoustic data at $285\,\mathrm{kHz}$, the camera records at $300\,\mathrm{Hz}$, this gives the following values for (1)

$$\begin{aligned} f_{min} &= 300\,\mathrm{Hz} \\ T_s &= \frac{1}{f_{min}} = 3.33\,\mathrm{ms} \\ P_{min} &= 2T_s = 6.67\,\mathrm{ms} \end{aligned} \tag{2}$$

Considering our minimal sample time to be $0.7\,\mathrm{s}$ thus

$$L = f_{min} \cdot 0.7\,\mathrm{s} = 210\,\mathrm{samples}$$

we choose

$$P_{max} = 20 \cdot P_{min}$$

to keep the operating point K above 10, which results in a strong autocorrelation function as is shown in figure 3a and figure 3b. Indeed, when applying the coefficients calculated in formula 2, the autocorrelation function shown in figure 3a displays a sharp peak at $t = 0$. In figure 3b we show the cross-correlation product between a small-scale node, where the synchronization is sampled directly by the ADC at $285\,\mathrm{kHz}$ and a high speed camera recording of an LED on one of the PCBs recorded at $300\,\mathrm{Hz}$. From figure 3b it is clear we can find the timing offsets in the recordings using our synchronization method.

III. Probabilistic Localization Algorithms

The goal of sound source localization is to retrieve the position $\vec{P}_s$ of a sound source:

$$\vec{P}_s = [x_s, y_s, z_s]^T$$

We use a probabilistic algorithm for sound source localization as a probabilistic method retains information on the likelihood of this and other positions rather than a deterministic algorithm such as least squares approximated solution which will only give us a single best fitted position. Data analysis may present edge cases where a single best fitted result is not the desired result e.g. multiple clusters of higher likelihood or an even spread of almost equally likely positions. A deterministic algorithm does not quantify the uncertainty of the solution, something that the probabilistic techniques do include naturally. Furthermore, the probabilistic formulation allows elegant extension to recursive Bayesian tracking algorithms such as an extended Kalman filter (EKF) or a particle filter.

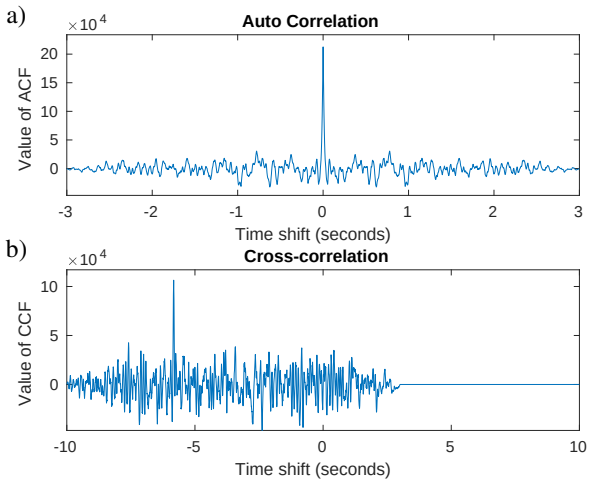

Fig. 3. The cross-correlation product of two embedded synchronization channels will show a sharp peak which can used to identify timing offsets in the recorded data so that we may synchronize the data in post processing; **Panel a):** Autocorrelation peak of the synchronization channel from a single small-scale node sampled at 285 kHz which shows a strong peak at $t = 0$ s. This means the synchronization channel has no timing offset when compared to itself. **Panel b):** Cross correlation product between two nodes where the synchronization channel is sampled at two different sampling frequencies, 285 kHz and 300 Hz. Data sampled at a lower sampling frequency is re-sampled to a higher sampling frequency. This plot shows a timing offset of $t \approx -5.8$ s which is to be expected due to a timing offset while starting a capture.

A. Peak Detection

The first step in either localization algorithm, AoA or TDoA, is determining the relative arrival times of the sounds. When identifying a sound with known properties we use a matched filter on the recorded data so that we can observe a sharp peak at timing intervals whenever the sound of interest is present in the recording. Another frequently used approach for determining timing differences in different microphone recordings relies on the generalized cross-correlation phase transform (GCC-PHAT) [26], [27]. Using GCC-PHAT takes into account phase information of the frequency spectrum and may yield lesser results for recordings with a bad signal to noise ratio [7]. Some filtering may be required based on properties such as room acoustics, signal to noise ratio or spectral properties of the sound to better identify peaks and eliminate false positives. Special care for linking sounds across the recordings needs to be taken when multiple sounds are present in a single recording due to the fact that not all sounds are detected by all microphones, which will definitely be the case when tracking a directive sound source such as a bat. By placing constraints on the maximum relative time difference based on properties of the capture area and previous sound locations we can better separate discrete sound emission events in our recordings and identify which microphones may have *missed* the sound.

From the detected peaks we can construct an arrival time vector $\vec{A}$ of the relative arrival times for each sound position relative to microphone 1. Note that microphone 1 is an arbitrarily chosen microphone which is used as a reference point. As no exact time of emission is known, it is safe to subtract d_1 from all detected peak times.

$$\vec{A} = [d_1 - d_1, d_1 - d_2, d_1 - d_3, \ \ldots, \ d_j - d_1]^T$$

Where j is the number of microphones. Hereafter we will refer to the relative arrival time $\vec{A}(j)$ when we mention to the arrival time d_j.

B. Small-Scale - Angle of Arrival

At the small-scale level, microphones are spaced a few centimetres apart and lie in a flat plane due to being placed on the surface of a PCB. Due to this configuration, nearly all incoming sound sources will be in the far-field for this array. As a first step we define a local measurement vector $\vec{M}_L$ with the difference in arrival times from all microphones of the node. To keep the formulas below compact, we restricted ourselves to 3 microphones:

$$\vec{M}_L = [d_1 - d_2, d_1 - d_3, d_2 - d_3]^T$$

with d_j the relative arrival time for microphone j. Using the location of each microphone, we can calculate an expected local measurement vector $\vec{M}_L, calc(\vec{P}_s)$ for all possible points $\vec{P}_s$. This gives us a distance vector $\vec{D}(\vec{P}_s)$ as a function of $\vec{P}_s$.

$$\vec{D}(\vec{P}_s) = calc(\vec{P}_s) \tag{3}$$

Sampling the points $\vec{P}_s$ for the distance vector $\vec{D}(\vec{P}_s)$ from a Cartesian grid is less optimal due to the far-field nature of the sound, indeed when we express the position in a spherical coordinate system we obtain more natural results. A spherical grid is better suited to the shape of the upcoming probability distributions.

$$\vec{P}_s = [\theta_s, \varphi_s, r_s]^T$$

With θ_s the azimuth of the sound source position, φ_s the elevation of the sound source position and r_s the range to the sound source, or the distance from the microphone to point $\vec{P}_s$. Sampling the grid at equidistant intervals in a 3D volume would be wasteful of computational resources as this will yield multiple points with equal azimuth and elevation that only differ in range. Sampling the grid as a 2D projection of equal azimuth and elevation would also be wasteful of computational resources as the equator length becomes shorter for directions near the poles. Therefore, we use a unit sphere partitioning algorithm to divide N points equally on the frontal hemisphere, avoiding the oversampling of the polar regions [28]. The radius of the unit sphere is much larger than the inter-microphone distance, thus all points of the distance vector reside in the far-field of the node. By only using points on a single hemisphere we can greatly reduce the amount of points $\vec{P}_s$ in the calculation thus reducing the computational complexity. Using distance vector $\vec{D}$, we can calculate the local likelihood $\mathcal{L}_L$ of the measurement $\vec{M}_{L,i}$ for a given point $\vec{P}_s$ for each microphone pair i as:

$$\mathcal{L}_L(\vec{M_{L,i}}|\vec{P}_s) = \exp\left[-(\vec{M}_{L,i} - \vec{D})^T \cdot \Sigma_L^{-1} \cdot (\vec{M}_{L,i} - \vec{D}) \right]$$

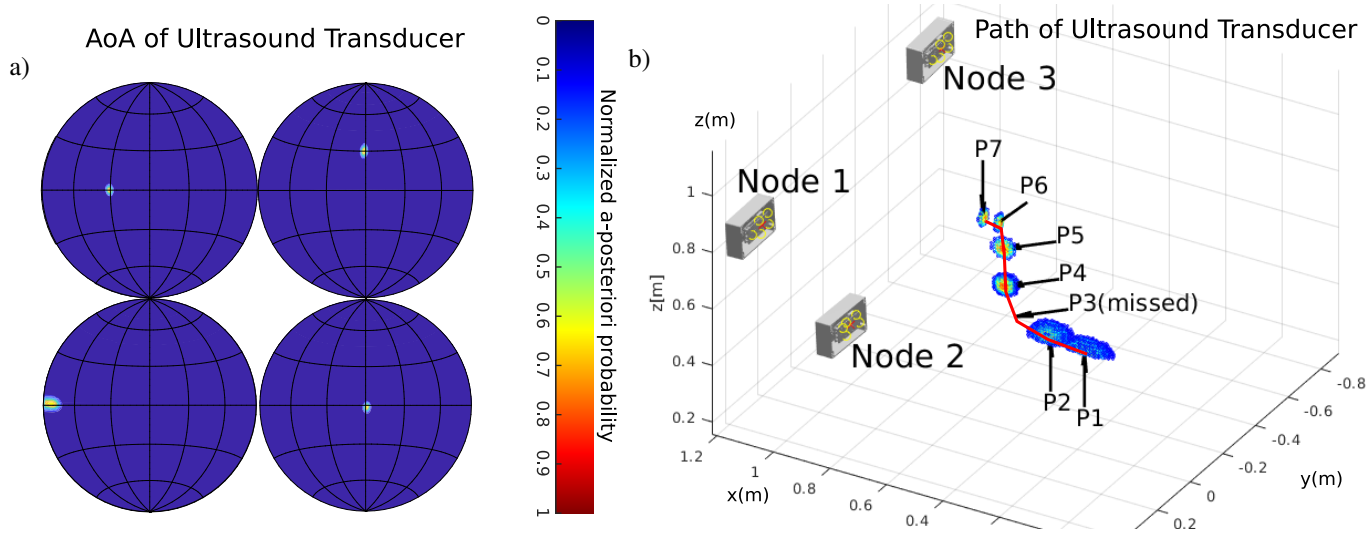

Fig. 4. **Panel a):** Shows the AoA from measurements on a single node. A sound source was placed at (θ, φ) relative to the node, upper left: (30,0), upper right: (0,30), lower left: (90, 0), lower right: (0,0). **Panel b):** In the lab we simulated a bat flight by moving an ultrasonic transducer free-handed in a manner that resembles the flighpath of a bat approaching prey. The ultrasonic emitter imitates a bat call i.e. a frequency sweep from 80 kHz to 20 kHz at an interval of 100 ms. Each discrete emission event is recognized by the large-scale array and the probabilistic likelihood is plotted. The obtained result shows an unimodal distribution arround the true source position in the form of an orb. To keep the figures clean, we only plotted points above a likelihood threshold.

with Σ_L the expected covariance matrix of the measurement errors. The combined local likelihood $\mathcal{L}_L$ for the entire small-scale array can then be found by multiplying the likelihoods for the individual microphone pairs together:

$$\mathcal{L}_L(\vec{M}_L|\vec{P}_s) = \prod_{i=1}^{n} \mathcal{L}_{L,i}(\vec{M}_L|\vec{P}_s)$$

with i the individual microphone pairs. The posterior distribution probability $\mathcal{P}$ for each position $\vec{P}_s$ given the measurement $\vec{M}_L$ can be found using Bayes' theorem:

$$\mathcal{P}(\vec{P}_s|\vec{M}_L) = \frac{\mathcal{L}_L(\vec{M}_L|\vec{P}_s) \cdot \mathcal{P}(\vec{P}_s)}{\mathcal{P}(\vec{M}_L)}$$

with $\mathcal{P}(\vec{P}_s)$ the prior distribution for position $\vec{P}_s$ and $\mathcal{P}(\vec{M}_L)$ the marginal of the posterior distribution. An estimate about the sound source position can now be made as

$$\widetilde{P}_L = \text{argmax } \mathcal{P}(\vec{P}_s|\vec{M}_L) \tag{4}$$

The result of (4) for a single node will be a point with Cartesian coordinates. However due to all $\vec{P}_s$ being sampled on a hemisphere with unique azimuth and elevation, we usually express (4) only in azimuth and elevation, or AoA.

C. Large-Scale - Time-Difference of Arrival

At the large-scale level we have microphones from multiple nodes distributed over a much larger area. As a result, sound recordings are made by multiple devices and the recordings are started and stopped by a non-realtime network protocol. The first step for the TDoA algorithm is to offload the recordings to a central processing point where the recordings can be synchronized. As described in section II-C we use an embedded synchronization signal to synchronize the recordings post-capture by calculating timing offsets in the recording and padding the recordings to match their timings. As a next step we select one or more microphones from each node to use in the TDoA localization algorithm. The trade-off made by choosing fewer microphones from each node for localization may only marginally decrease the performance of the localization algorithm while requiring far less computation power thus improving the time required for obtaining a result which may be desirable if the results of the localization algorithm are used in a feedback loop or if the calculations are done on an embedded SBC which typically has less than average computing power. We start by constructing the global measurement vector $\vec{M}_G$ for the large-scale array as:

$$\vec{M}_G = [d_1 - d_2, d_1 - d_3, d_2 - d_3]^T$$

where d_i can be any microphone of any small-scale node. In the notations, we restricted ourselves to 3 microphones for readability. As a next step, a grid of possible source locations is constructed by taking into account information such as the node locations, the capture volume, room dimensions or known properties of the sound source path/location. An equidistant spaced grid in the area of interest is constructed for $\vec{P}_s$ so that we can calculate a distance vector $\vec{D}$ for the points in the grid and the microphones of interest. Much consideration is needed when constructing $\vec{P}_s$ to ensure the actual position of the sound is within the bounds of $\vec{P}_s$ otherwise the most likely position will be meaningless. Furthermore, under-sampling $\vec{P}_s$, or choosing too few points, may lead to inaccurate results while over sampling can drastically increase the required compute time, a good strategy for choosing points of interest is needed.

Similarly to what we have done in (3) for the local measurement, we construct a global distance vector $\vec{D}_G$ as a function

of $\vec{P}_s$ where we use the global grid so that we can calculate the global likelihood $\mathcal{L}_G$ for measurement $\vec{M}_G$ given $\vec{P}_s$. For individual microphone pairs or individual clusters of microphones, the individual global likelihood $\mathcal{L}_{G,i}$ can be found using:

$$\mathcal{L}_G(\vec{M}_{G,i}|\vec{P}_s) = \exp\left[-(\vec{M}_{G,i}-\vec{D}_G)^T \cdot \Sigma_G^{-1} \cdot (\vec{M}_{G,i}-\vec{D}_G)\right]$$

with i the individual microphone pair or cluster. During the construction of Σ_G^{-1} it is important to note that the error variance between microphones from the same small-scale node are much smaller the the variance in the error between microphones from different nodes of the large-scale array. Furthermore, it is expected that covariances are required only for the small-scale microphone pairs and not for large-scale pairs. This is something that can be measured in future experiments. For now, we use a diagonal matrix with different values for the small-scale pairs and large-scale pairs for Σ_G^{-1}.

The global likelihood $\mathcal{L}_G$ can be found by multiplying the individual likelihoods together:

$$\mathcal{L}_G(\vec{M}_G|\vec{P}_s) = \prod_{i=1}^{n} \mathcal{L}_G(\vec{M_{G,i}}|\vec{P}_s)$$

which can be used in a Bayesian formulation to find the probability $\mathcal{P}$ for a certain position P_s:

$$\mathcal{P}(\vec{P}_s|\vec{M}_G) = \frac{\mathcal{L}(\vec{M}_G|\vec{P}_s)\cdot\mathcal{P}(\vec{P}_s)}{\mathcal{P}(\vec{M}_G)} \tag{5}$$

Finally, the sound source position can be estimated as:

$$\widetilde{P}_G = \operatorname{argmax}\ \mathcal{P}(\vec{P}_s|\vec{M}_G) \tag{6}$$

By calculating the posterior distribution as described in (5) we are able to find a unimodal peak around the true source position that takes the form of an ellipsoid, as can be seen in figure 4b, which is true if the measurement $\vec{M}$ only contains unbiased Gaussian measurement errors. A possible optimization can be performed by running the TDoA algorithm again on a finer grid delimited by the ellipsoid.

Equation (6) can also be maximized using a non-linear optimization function which reduces the need for evaluation on a grid. Furthermore, the tracking of an animal becomes more straightforward as we can use a recursive Bayesian filter such as an EKF or particle filter where we can also use the motion model for the source-of-interest.

IV. Experimental Results

To verify the validity of our approach, we constructed three small-scale nodes, see figure 1c for the hardware stack and figure 1d for the finished device. Using an ultrasound transducer (SensComp 7000 series) we can play a bat call that we recorded from an *M. Daubentonii* bat i.e. a frequency sweep from $60\,\mathrm{kHz}$ to $40\,\mathrm{kHz}$. The large-scale array consists of three small-scale nodes mounted on tripods for easy reconfiguration of the node location and orientation. In the lab we also have access to a Qualisys optical motion capture system which uses infrared light to triangulate small retro-reflective markers within $0.5\,\mathrm{mm}$. Using the Qualisys system we can measure the location and orientation of the nodes and provide ground-truth verification for the location of the sound source. The sound source was placed on a robot-arm for easy replication of measurements, as well as used free-handed to imitate the flight-path of a bat approaching his prey. As shown in [1], [4], [29], some species of bats exhibit trawling behavior when approaching prey.

1) Angle of Arrival: Using a robot-arm (STRobotics R17) we were able to place the emitter at exact locations with regard to the the nodes. We moved the transducer to a couple of points with known azimuth and elevation in front of the node. The results, as can be seen in figure 4a show that we were able to estimate the AoA of the sound using a single node to within 2 degrees with an average error of 1 degree.

2) 3D localization: Using a large-scale array of three nodes of equal orientation, we made a free-handed motion mimicking a trajectory of a trawling bat (e.g. M. Daubentonii) toward the large-scale array. The emitter transmitted an ultrasonic pulse mimicking the hunting calls of M. Daubentonii continuously with a repetition rate of $5\,\mathrm{Hz}$. We collected 10 echolocation calls of which 7 were eligible for path reconstruction. The missing calls can be explained by the directivity of the transducer, causing some microphones not to be able to pick up the emitted signal. We repeated this experiment 10 times and we show a typical example of a reconstructed path in figure 4b. Around the estimated maximum a-posteriori positions we indicated the a-posteriori probability function using colored circles. As expected from array localization theory, the posterior distribution of the location estimate becomes more narrow as the source gets closer to the microphone array. This is caused by the fact that the source gets more in to the near-field of the microphone array.

V. Conclusions and Future Work

In this paper we have shown that a low-cost hardware architecture can be used to construct a flexible, scale-free WSN capable of localizing an bio-acoustic source. We have described a topology that can be used for the creation of distributed microphone arrays where the microphones are distributed in large-scale and small-scale arrays. We described some considerations that are relevant when creating such a topology. We also created the required hardware for constructing a low-cost microphone array that can be used to create a WSN able to locate a sound. In our lab experiments we are able to locate a sound source to within 2 degrees using the AoA method for the small-scale arrays and within a couple of centimeters using the TDoA algorithm for the large-scale array. In our future work we will focus on tracking wild echolocating bats in relevant biological settings and use motion tracking systems to validate the localization performance of our acoustic sensor. Furthermore we will expand on the tracking algorithm by implementing recursive Bayesian filters such as particle filters or non-causal approaches as this type of behavioral data is typically processed in batches. Finally

we will demonstrate the scale-free aspect of our approach by jointly tracking different animal species that interact in a certain ecological niche such as Trachops cirrhosus hunting small vocalizing frogs.

ACKNOWLEDGEMENTS

Erik Verreycken is funded by a grant from the special research fund of the University of Antwerp.

REFERENCES

[1] N. B. Kothari, M. Wohlgemuth, and C. F. Moss, "Dynamic representation of 3D auditory space in the midbrain of the free-flying echolocating bat," *The Journal of the Acoustical Society of America*, vol. 142, no. 4, pp. 2691–2691, oct 2017.

[2] F. De Mey, F. Schillebeeckx, D. Vanderelst, A. Boen, and H. Peremans, "Modelling simultaneous echo waveform reconstruction and localization in bats," *Biosystems*, vol. 100, no. 2, pp. 94–100, may 2010.

[3] F. Guarato, L. Jakobsen, D. Vanderelst, A. Surlykke, and J. Hallam, "A method for estimating the orientation of a directional sound source from source directivity and multi-microphone recordings: Principles and application," *The Journal of the Acoustical Society of America*, vol. 129, no. 2, pp. 1046–1058, feb 2011.

[4] L. Jakobsen and A. Surlykke, "Vespertilionid bats control the width of their biosonar sound beam dynamically during prey pursuit," *Proceedings of the National Academy of Sciences*, vol. 107, no. 31, pp. 13 930–13 935, aug 2010.

[5] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, jul 1996.

[6] Y. Han, Y. Shen, X.-P. Zhang, M. Z. Win, and H. Meng, "Performance Limits and Geometric Properties of Array Localization," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 1054–1075, feb 2016.

[7] S. Argentieri, P. Danès, and P. Souères, "A survey on sound source localization in robotics: From binaural to array processing methods," *Computer Speech & Language*, vol. 34, no. 1, pp. 87–112, nov 2015.

[8] S. Thrun, "Affine structure from sound," *IN NIPS*, vol. 18, pp. 1353—-1360, 2005.

[9] J.-A. Luo, X.-P. Zhang, Z. Wang, and X.-P. Lai, "On the Accuracy of Passive Source Localization Using Acoustic Sensor Array Networks," *IEEE Sensors Journal*, vol. 17, no. 6, pp. 1795–1809, mar 2017.

[10] G. Simon et al, "Sensor network-based countersniper system," in *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*. New York, New York, USA: ACM Press, 2004, p. 1.

[11] J. Kang, Y. Kim, and D. Lee, "Visualiation of door-slam sound by using beamforming," in *6th Berlin Beamforming Conference (BEBEC16)*, 2016.

[12] J. C. Rivey, G.-Y. Lee, J. Yang, Y. Kim, and S. Kim, "Acoustic Emission Beamforming for Detecting and Localizing Damage in Composite Materials," in *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, jan 2016.

[13] F. Hoflinger, J. Hoppe, R. Zhang, A. Ens, L. Reindl, J. Wendeberg, and C. Schindelhauer, "Acoustic indoor-localization system for smart phones," in *2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14)*. IEEE, feb 2014, pp. 1–4.

[14] H. Balakrishnan, R. Baliga, D. Curtis, M. Goraczko, A. Miu, N. B. Priyantha, A. Smith, K. Steele, S. Teller, and K. Wang, "Lessons from developing and deploying the cricket indoor location system," *Preprint*, 2003.

[15] N. M. Freris, S. R. Graham, and P. R. Kumar, "Fundamental Limits on Synchronizing Clocks Over Networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1352–1364, jun 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5605654/

[16] D. Veitch, J. Ridoux, and S. Korada, "Robust Synchronization of Absolute and Difference Clocks Over Networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 417–430, apr 2009. [Online]. Available: http://ieeexplore.ieee.org/document/4569868/

[17] E. Verreycken, D. Laurijssen, W. Daems, and J. Steckel, "Firefly based distributed synchronization in Wireless Sensor Networks for passive acoustic localization," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, oct 2016.

[18] C. Funai, C. Tapparello, and W. B. Heinzelman, "Supporting Multi-hop Device-to-Device Networks Through WiFi Direct Multi-group Networking," *CoRR*, vol. abs/1601.0, dec 2016.

[19] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y.-C. Wu, "Clock synchronization in wireless sensor networks: an overview." *Sensors (Basel, Switzerland)*, vol. 9, no. 1, pp. 56–85, 2009.

[20] M. Mock, R. Frings, E. Nett, and S. Trikaliotis, "Continuous clock synchronization in wireless real-time applications," in *Proceedings 19th IEEE Symposium on Reliable Distributed Systems SRDS-2000*. IEEE Comput. Soc, pp. 125–132.

[21] S. Ping, "Delay measurement time synchronization for wireless sensor networks," tech," *REP., INTEL RESEARCH*, 2003.

[22] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, p. 147, dec 2002.

[23] W. Su and I. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 384–397, apr 2005.

[24] L. Gasparini, O. Zadedyurina, G. Fontana, D. Macii, A. Boni, and Y. Ofek, "A Digital Circuit for Jitter Reduction of GPS-disciplined 1-pps Synchronization Signals," in *2007 IEEE International Workshop on Advanced Methods for Uncertainty Estimation in Measurement*. IEEE, jul 2007, pp. 84–88.

[25] D. Laurijssen, E. Verreycken, I. Geipel, W. Daems, H. Peremans, and J. Steckel, "Low-cost synchronization of high-speed audio and video recordings in bio-acoustic experiments." *The Journal of experimental biology*, vol. 221, no. Pt 4, p. jeb173724, feb 2018.

[26] B. Mungamuru and P. Aarabi, "Enhanced Sound Localization," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 3, pp. 1526–1540, jun 2004.

[27] Q. H. Wang, T. Ivanov, and P. Aarabi, "Acoustic robot navigation using distributed microphone arrays," *Information Fusion*, vol. 5, no. 2, pp. 131–140, jun 2004.

[28] P. Leopardi, "A partition of the unit sphere into regions of equal area and small diameter," *Electronic Transactions on Numerical Analysis*, vol. 25, no. 12, pp. 309–327, 2006.

[29] T. P. Gonzalez-Terrazas, J. C. Koblitz, T. H. Fleming, R. A. Medellín, E. K. V. Kalko, H.-U. Schnitzler, and M. Tschapka, "How Nectar-Feeding Bats Localize their Food: Echolocation Behavior of Leptonycteris yerbabuenae Approaching Cactus Flowers," *PLOS ONE*, vol. 11, no. 9, sep 2016.