DiVA: Distributed Voronoi-based Acoustic Source Localization with Wireless Sensor Networks

Xueshu Zheng*, Shuailing Yang*, Naigao Jin*, Lei Wang*, Mathew L. Wymore†, and Daji Qiao†

*Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province
School of Software, Dalian University of Technology, China
†Department of Electrical and Computer Engineering, Iowa State University, USA

Abstract—This paper presents DiVA, a novel hybrid range-free and range-based acoustic source localization scheme that uses an ad-hoc network of microphone sensor nodes to produce an accurate estimate of the source's location in the presence of various real-world challenges. DiVA uses range-free pairwise comparisons of sound detection timestamps between *local Voronoi neighbors* to identify the node closest to the acoustic source, which then estimates the source's location using a constrained range-based method. Through simulation and experimental evaluations, DiVA is shown to be accurate and highly robust, making it practical for real-world applications.

I. INTRODUCTION

The continued prevalence of gun violence in public areas has shown a need for more advanced methods for dealing with such scenarios. An ad-hoc wireless acoustic sensor network pre-deployed in an at-risk area could use a robust localization scheme to provide authorities with critical information. Such a system could provide shooter localization [1], [2] or, along with a camera system, advanced surveillance [3]. With in-network acoustic classification [4], such a system could also be used for applications such as detailed wildlife monitoring [5].

Assuming that sensor nodes have known geographical positions, simple range-based localization methods, such as time of arrival (TOA) and time difference of arrival (TDOA), would work well for an ideal version of these scenarios. However, these methods can be inaccurate in the presence of the real-world challenges inherent in acoustic localization. These challenges include the imprecise nature of acoustic timestamping, the imperfect clock synchronization between sensor nodes, and the inaccuracies of outside systems, such as GPS, in determining the positions of sensor nodes. Additionally, more advanced schemes are needed to meet the challenges inherent in low-power, mutlihop wireless sensor networks.

This paper presents DiVA, a distributed Voronoi-based acoustic source localization scheme that addresses these issues. DiVA is a unique hybrid of range-free and range-based methods, making it accurate, robust to real-world conditions, quick to converge, and lightweight in terms of communication and computational complexity. More specifically, DiVA uses range-free pairwise comparisons of sound detection timestamps of neighboring nodes, a process that is tolerant to various sources of error, to produce a coarse estimate of the target's location. This coarse estimate is then refined using TDOA and constrained optimization, reliably producing a high-accuracy estimate even in the presence of harsh real-world conditions. The remainder of this paper presents related work, the baseline approach of DiVA, practical considerations and extensions for DiVA, a simulation study, and an experimental evaluation of DiVA.

II. RELATED WORK

Acoustic source localization systems can be classified based on the acoustic feature used for localization, including TOA [6]–[8], TDOA [2], [9]–[11], angle of arrival (AOA) [12], [13], and acoustic energy [14]–[18]. Range-free methods also exist [19]–[24]. Both TOA- and TDOA-based methods require accurate time synchronization between sensor nodes, and are highly sensitive to the relative positions and the position errors of the sensor nodes used for localization. AOA-based methods usually require additional hardware support, such as multiple microphones on each sensor node. Energy-based methods do not require time synchronization, but may suffer from the energy-distance mismatch problem: louder does not necessarily imply closer. Range-free methods mainly utilize proximity information and rely on collaborations between sensor nodes to produce an estimate of the source location. These methods have high robustness and low system cost, but suffer from decreased accuracy as the distance between nodes increases. DiVA is a unique hybrid solution that combines the robustness of range-free methods with the accuracy of TDOA.

Localization schemes can also be classified by system architecture. Most research focuses on methods for processing sensor data in a centralized architecture, where all nodes report readings to a base station. These systems can be highly accurate, but do not address practical issues for sensor networks, such as scalability. For instance, transferring all data to a base station can be time-consuming and lead to high communication overhead.

A limited number of distributed or decentralized acoustic source localization schemes have been proposed. The energy-based DIG [15] uses a cycling mechanism to arrive at a final estimate. C-DRASL [18], also energy-based, uses a distributed consensus algorithm. Both methods require a number of iterations, and therefore time and communication overhead, to converge. Lightning [24] uses the difference between acoustic and RF signal propagation to quickly identify the node closest to the acoustic source. Lightning's hard real-time requirement and broadcast-based communications may not be robust under

harsh conditions. The DSL family of schemes [22], [23], uses a broadcast-based election process to identify the node closest to the acoustic source, which then establishes a voting grid centered on itself. Each nearby node uses timestamp comparisons with the other nearby nodes to estimate the source's location and cast votes in the voting grid. The DSL family relies heavily on one-hop broadcast communication, limiting its scalability. Chen et al. [17] proposed a clustered scheme that uses a Voronoi diagram to determine which cluster head is the closest to the source, but the scheme requires a backbone of higher-capability cluster head nodes.

In contrast, DiVA uses a hybrid architecture in which nodes first collaborate using a distributed, Voronoi-based, range-free method to identify the node closest to the acoustic source. This node then has enough information to act as a local base station. It first uses a range-free method to determine a small region that contains the source, then solves a constrained optimization problem on TDOA estimates from a select group of neighboring nodes to obtain a precise estimate of the source's location. With this architecture, DiVA achieves high accuracy and robustness with low complexity and no assumptions about topology or deployment conditions, making it practical for real-world deployments.

III. BASELINE APPROACH

DiVA's key idea is to divide the area covered by the microphone nodes into *local Voronoi cells* and use pairwise comparisons between neighbors to traverse the diagram to the acoustic source. This is accomplished in four phases. In the initialization phase, nodes determine their *local Voronoi neighbors* (LVNs), defined in the following subsection. Then, when a sound is heard, nodes collaborate with their LVNs to identify the node closest to the source of the sound. This node then uses information from its LVNs to narrow down the location to a target subregion. Finally, TDOA estimates from the surrounding nodes are processed with a constrained optimization procedure to produce a final location estimate within the target subregion. In this hybrid globally-distributed and locally-centralized architecture, nodes communicate only with their LVNs.

The baseline approach described in this section assumes a 2D network of nodes with known positions (provided by an outside system), and a single acoustic source that emits a single beep. The extension of DiVA to 3D space, by replacing 2D Voronoi cells with 3D Voronoi spaces, is fairly straightforward; however, the details are omitted due to space limitations. Analysis of practical issues is presented in Section IV, along with an extension for localizing a series of beeps.

A. Phase I - Initialization

Nodes must first determine their LVNs using local neighbor information and a Voronoi diagram. LVNs and the necessary related concepts are defined as follows.

Voronoi neighbor: Applied to a sensor network, a *Voronoi diagram* [25], [26] divides a region into cells, with one cell per node, such that each point in the region belongs to the cell of the node closest to that point. A *Voronoi neighbor* (VN) of a node A is any node whose cell shares a border with A's cell in the Voronoi diagram of the deployment.

Communication neighbor: Any node that has a wireless link with a node A is a *communication neighbor* (CN) of A. For simplicity of analysis, all wireless links are assumed to be symmetric.

Local Voronoi neighbor: Any node that is both a VN and a CN of a node A is a *local Voronoi neighbor* (LVN) of A. Specifically, if V_A is the set of A's VNs, C_A is the set of A's CNs, and Φ_A is the set of A's LVNs, then $\Phi_A = V_A \cap C_A$. An example is shown in Fig. 1.

In DiVA, nodes interact only with their LVNs. LVNs are used instead of VNs due to the practical limitation of communication range. In an irregular deployment, a node may not be able to communicate directly with some VNs; for instance, in Fig. 1a, nodes A and B are outside each other's communication range. Consequently, the local Voronoi cells of A and B overlap. In DiVA, an acoustic source in such an overlapping area, such as ▲ in Fig. 1, will generate results at both nodes, and both results will be reported to the user for post-processing, as discussed in Phase IV.
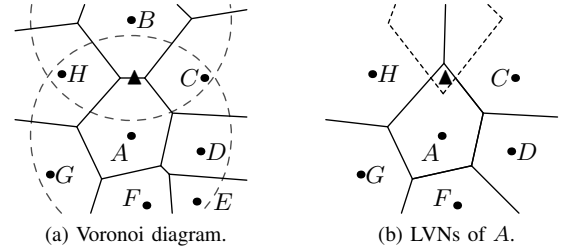

Fig. 1. LVN example. The dashed circles in (a) show the communication ranges of A and B. The dotted lines in (b) show the local Voronoi cell of B.

When a node enters Phase I, it broadcasts a request to discover its CNs and their positions. With this information, the node can determine its set of LVNs by computing its local Voronoi diagram. In order to maintain optimal operation of DiVA, nodes must keep their sets of LVNs updated. This can be achieved with periodic handshake messages.

B. Phase II - Target Cell Identification

1) Overview: With its LVNs determined, a node is ready to participate in localization. In Phase II, nodes collaborate to identify the node closest to the target acoustic source (▲), a process shown in Fig. 2. By definition, this node's Voronoi cell contains ▲.

In Phase II, each node that hears a sound matching the target acoustic signature, called a *beep*, designates itself the *leader* of its own *localization instance*, a unique execution of DiVA's localization process. This leader then probes its LVNs in a circular pattern, looking for an LVN closer to ▲, determined by comparing timestamps for the beep. If the leader finds an LVN that is closer to ▲, it transfers leadership to that LVN. Successive iterations of this process are shown in Figs. 2a–2c. In this way, leadership is passed among the nodes until the

node closest to ▲ becomes the leader. This node probes all of its LVNs and finds none closer to ▲, as shown in Fig. 2d, so this node's local Voronoi cell contains ▲.
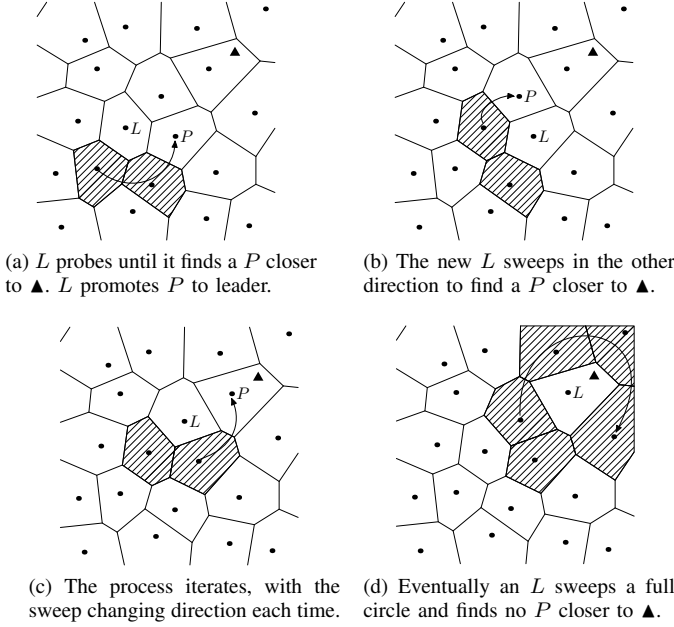

(a) L probes until it finds a P closer to ▲. L promotes P to leader.

(b) The new L sweeps in the other direction to find a P closer to ▲.

(c) The process iterates, with the sweep changing direction each time.

(d) Eventually an L sweeps a full circle and finds no P closer to ▲.
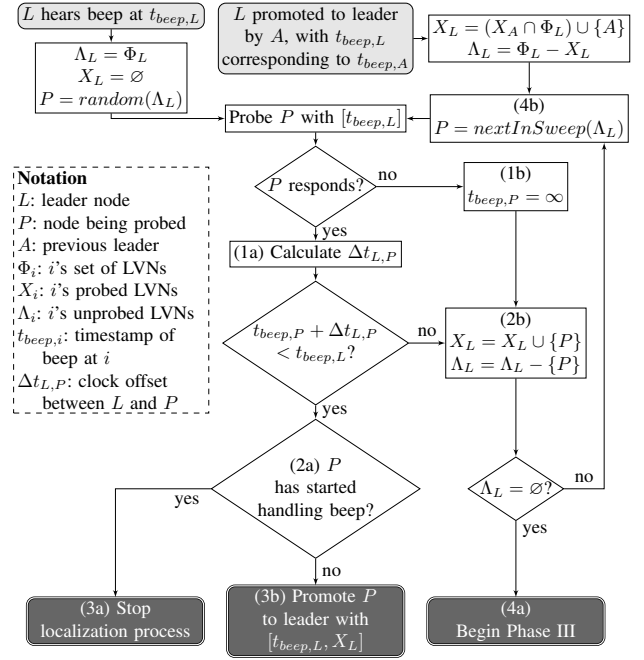
Fig. 2. Example of Phase II - Target Cell Identification. ▲ is the target acoustic source. The arrows indicate probing sweeps, and hatched cells indicate nodes that L has probed or nodes that L knows have already been probed.

Since the node closest to ▲ should hear the beep and initiate probing first, leadership passes should rarely be necessary in practice, but this well-defined process is still needed to be robust to uncertainty in the timing of distributed activities.

2) Details: The details of the Phase II process are outlined in the flowcharts in Fig. 3. Fig. 3a shows the actions of the leader node during this phase. A node can become the leader L of a localization instance by hearing a beep and starting the localization process. L records the time of the beep as $t_{beep,L}$. L populates Λ_L, its set of unprobed LVNs, with all members of Φ_L, its set of LVNs. L also empties X_L, the set of probed LVNs. As LVNs are probed, they are removed from Λ_L and added to X_L. To begin probing, L chooses a random node from Λ_L as the first P, the node being probed.

A node L can also become the leader by being promoted by some previous leader node A. In this case, L uses information passed from A to initialize $X_L = (X_A \cap \Phi_L) \cup \{A\}$ and $\Lambda_L = \Phi_L - X_L$. The first P chosen by L is then determined by a sweeping process, called *nextInSweep* in Fig. 3a, that uses L's knowledge of the positions of Φ_L to determine the next $P \in \Lambda_L$ in a circular sweep around L. The direction of the sweep changes when leadership is passed, sweeping away from the already-probed common LVNs of A, to minimize the number of probed nodes. The *nextInSweep* process is also used to find the next P if the previous P was not closer to ▲.

Regardless of how L became the leader, once it has chosen the next P, it begins probing. To probe P, L unicasts P a probe message containing $t_{beep,L}$. As shown in Fig. 3b, if P

(a) Flowchart for a leader node L.
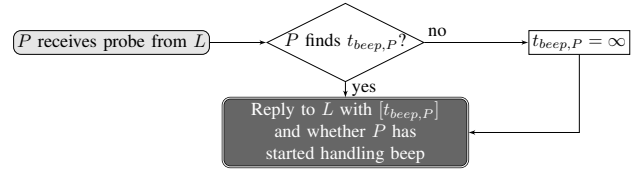
(b) Flowchart for a probed node P.

Fig. 3. Flowcharts for Phase II - Target Cell Identification. Lightly shaded shapes indicate entry points and darkly shaded shapes indicate exit points.

has a corresponding $t_{beep,P}$, it is sent back to L in a reply message. If not, $t_{beep,P} = \infty$ is reported instead. P also informs L of whether or not it has started handling the beep.

If L receives a reply from P, it calculates the clock offset between L and P, $\Delta t_{L,P}$, as shown in Case 1a in Fig. 3a. $\Delta t_{L,P}$ is discussed in the following subsection. L compares its beep timestamp with P's beep timestamp corrected with $\Delta t_{L,P}$. If $t_{beep,P} + \Delta t_{L,P} < t_{beep,L}$ (Case 2a), then P is closer to ▲ than L. If P has not already started handling the beep (Case 3b), L sends a promotion packet to P that contains $t_{beep,L}$ and X_L. On the other hand, if P is closer and has already begun handling the beep (Case 3a), L stops acting as a leader, ending this instance of the localization process. Most localization instances are expected to end in this manner.

If P is farther away from ▲ than L (Case 2b), then L moves P from Λ_L to X_L. If Λ_L is not yet empty (Case 4b), L uses *nextInSweep* to determine the next $P \in \Lambda_L$. L then probes the new P. However, if Λ_L is empty after removing P (Case 4a), then L must be the closest node to ▲. Therefore, the target cell has been identified, so L proceeds to Phase III.

A final case is that L may not receive a reply from P after probing it (Case 1b). This could happen if channel conditions have changed, or if P has failed. In this case, after a timeout,

L may retry the probe a number of times up to a retry limit. If the probe continues to fail, L assumes $t_{beep,P} = \infty$, updates the probing sets, and chooses the next P.

3) Clock Offset Correction: Since each node in the network maintains its own clock, an offset $\Delta t_{L,P}$ may exist between the clocks of nodes L and P. Beep timestamps must be corrected for this clock offset, as in Fig. 3a, before they can be compared. Since only the current clock offset is needed, clock drift is not a factor. To find this clock offset with minimal overhead, DiVA integrates a simple and well-known pairwise local synchronization method [27] into the probing process. This method uses the four timestamps shown in Fig. 4: $t_{probe,L}$, $t_{probe,P}$, $t_{reply,P}$, and $t_{reply,L}$.
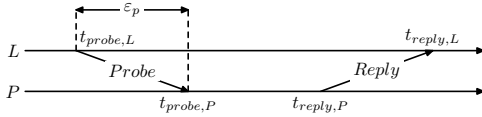

Fig. 4. Timestamps used for calculating the clock offset between L and P.

With these timestamps, $\Delta t_{L,P}$ can be calculated as follows:

$$\Delta t_{L,P} = \frac{(t_{probe,L} - t_{probe,P}) - (t_{reply,P} - t_{reply,L})}{2}. \quad (1)$$

Using this method, in a sensor network, $\Delta t_{L,P}$ can be determined with an error of around 44 μs or less [27]. As previously noted, $\Delta t_{L,P}$ is used for correcting beep timestamps, as shown in the flowchart in Fig. 3a. $\Delta t_{L,P}$ is also used when localizing a series of beeps, discussed in Section IV-B.
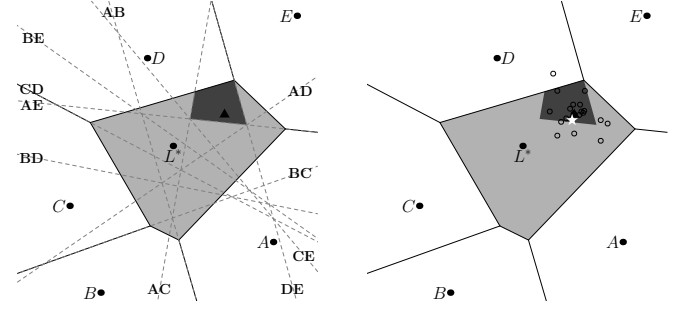
C. Phase III - Target Subregion Identification

Let L^* be the node that is closest to ▲, as determined in Phase II. In Phase III, shown in Fig. 5a, L^* uses pairwise timestamp comparisons to narrow down the location of ▲ to a *target subregion*, denoted by S, of its local Voronoi cell. First, L^* cuts its cell into subregions by drawing a perpendicular bisector between each pair of its LVNs. Then, for each cut, L^* determines which side of the cut contains ▲ by comparing t_{beep} for the corresponding LVNs. L^* discards all subregions on the side of the cut without ▲. After cutting and discarding for each pair of LVNs, only the subregion S containing ▲ remains. If this target subregion is open on any side, then it is bounded by the sensing range of L^*'s acoustic sensor on that side. The target subregion is used to constrain the final target location estimation in Phase IV.

D. Phase IV - Constrained Target Location Estimation

In Phase IV, shown in Fig. 5b, L^* uses a range-based method to produce a final estimate of the target's location. The target subregion S identified in Phase III is trusted to contain ▲, so S is used as a constraint to reduce the search area. The following is proposed as a simple range-based method suitable for computing on a sensor node, but other range-based methods may be applicable as well.

First, L^* calculates TDOA estimates of the location of ▲ using all combinations of three nodes in $L^* \cup \Phi^*$, where Φ^*

(a) Phase III. The dashed lines are cuts, labeled with the pair of LVNs that generate that cut.

(b) Phase IV. The TDOA estimates are marked ○ and $\hat{t}$ is marked as ☆.

Fig. 5. Examples of Phase III - Target Subregion Identification and Phase IV - Constrained Target Location Estimation. L^*'s local Voronoi cell is lightly shaded and contains ▲. The identified target subregion S is darkly shaded.

is the set of L^*'s LVNs. The set of TDOA location estimates is denoted by T, and the total number of location estimates $|T|$ is:

$$|T| = \binom{|\Phi^*| + 1}{3} = \frac{(|\Phi^*| + 1)(|\Phi^*|)(|\Phi^*| - 1)}{6}. \quad (2)$$

L^* then performs a simple Grubbs' test [28] on these estimates, with a significance level of $\alpha = 0.05$, to remove the outliers from T. Let T' denote the set of TDOA estimates remaining after the Grubbs' test. The final location estimate $\hat{t}$ is the point in S that minimizes the mean squared error (MSE) for $t_i = (x_i, y_i) \in T'$, as follows:

$$\hat{t} = (x^*, y^*) = \underset{(\hat{x},\hat{y}) \in S}{\operatorname{argmin}} \frac{1}{|T'|} \sum_{t_i \in T'} \left[(x_i - \hat{x})^2 + (y_i - \hat{y})^2 \right]. \quad (3)$$

Finally, L^* reports $\hat{t}$ and its MSE to the user. Due to the possible overlap of local Voronoi cells discussed in Section III-A, and the possibility of *false local minimums* in the presence of *time error*, as discussed in Section IV-A3, multiple nodes may reach Phase IV, resulting in multiple estimates for the same beep. The user can then select the $\hat{t}$ with the smallest MSE as the final estimate.

If L^* only has one LVN, no TDOA estimates are possible. In this case, L^* reports the center of gravity of the target subregion as $\hat{t}$, and reports a high associated MSE so that estimates made with more information are given higher priority.

E. Complexity Analysis

The complexity analysis of DiVA is divided into three parts: network-wide communication, Phase II convergence time, and single-node computation in Phases III and IV. Communication in the network occurs primarily during Phase II, when nodes are probed and leadership is passed. Every probe and pass crosses one edge in the Voronoi diagram, and by design, edges are not recrossed in a single localization instance. It is well known that the maximum number of edges in a Voronoi diagram with n nodes is $3n-6$. Therefore, if n_{detect} is the number

of sensors in the network that can detect the beep, the network-wide communication complexity from probing is $O(n_{detect})$. Since communication generally consumes the most energy in a sensor network, and the amount of communication depends on the localization scheme, this also serves as a metric for DiVA's energy use. The LVN maintenance process also leads to communication overhead, depending on the rate of change in the network topology. This overhead is minimal for a static network, and can be further reduced by piggybacking LVN maintenance information on other traffic.

Assuming that the node L^* closest to ▲ hears the beep and begins localization first, the convergence time of Phase II depends only on the number of LVNs of L^*, which is no more than k, the number of L^*'s communication neighbors. Since each LVN is probed once, the convergence time is $O(k)$.

Finally, the nodal computational complexity is dominated by Phases III and IV, when L^* computes perpendicular bisecting cuts, TDOA estimates, outliers, and MSEs to estimate the location of ▲. A cut is computed for every pair of LVNs, so the number of cuts computed is no more than $\binom{k}{2} = \frac{k(k-1)}{2}$. The number of TDOA estimates is $|T|$, obtained in Eq. (2), with $|\Phi_L| \leq k$, yielding $O(k^3)$. The complexity of outlier detection and the MSE estimates also depends on $|T|$. Since these operations are performed sequentially, the overall computational complexity for L^* is $O(k^3)$. This is a loose bound, because the number of LVNs is expected to be small and may be much smaller than k. Nodes not associated with a target cell do not perform significant computations.

IV. PRACTICAL CONSIDERATIONS & EXTENSIONS

A. Robustness

DiVA is designed to be highly robust to node failure, packet loss, microphone position error, beep measurement error, clock synchronization error, and obstacles.

1) Node Failure: Node failure, including power loss and hardware or sensor malfunction, could occur either at a node expected to start the localization process, or at a node expected to be part of the target cell identification process in Phase II. DiVA is robust to both cases. The first case has a minimal effect on DiVA because all nodes that hear a beep initiate a localization instance, so DiVA does not depend on one particular node. However, most of these localization instances are expected to terminate early in Phase II, when they encounter a node closer to ▲ that has already handled the beep. Therefore, the overhead from this redundancy is small, while the gain in robustness is large.

For the second case, DiVA's Phase II is capable of detouring around a node P that has failed, missed the beep, or is reporting excessively large timestamps, such as during a sensor malfunction. In these cases, a leader L may not promote P to leader, even though P may be closer to ▲ than L. However, if a second route exists, the transfer of leadership will detour around the malfunctioning node. The detour can cause extra iterations before the target cell is found, but the correct target cell will still be identified.

2) Packet Loss: DiVA is also robust to packet loss. The leader controls the unicast probing process, so a probe may be retried if a response is not received. This contrasts with schemes based on broadcast messages, such as the DSL family [22], [23], in which the process of sharing data is undirected and the amount of data that should be received is uncertain. This also contrasts with centralized solutions, for which the process of gathering data from all nodes becomes more burdensome under poor channel conditions.

3) Input Error: One of DiVA's design goals is robustness to various sources of error not in control of the scheme, called *input error*. Such error includes the following:

Microphone position error: Inaccuracy in the geographical coordinates provided by an external system or node localization service.

Beep measurement error: Inaccuracy in the recorded time of the beep, t_{beep}. This may be caused by hardware delays or the indeterministic nature of audio feature extraction.

Clock synchronization error: Inaccuracy in the calculated clock offset between two nodes L and P, $\Delta t_{L,P}$. This may be caused by hardware delays or the indeterministic nature of message timestamping and channel access.

Position error does not affect the range-free process of Phase II, so the network can still find L^*, the node closest to ▲, in the presence of position error. The position error of L^* and its LVNs $\phi \in \Phi^*$ may skew the target subregion in Phase III, and this position error also affects the range-based estimation in Phase IV. However, since DiVA constrains the final estimate to the local Voronoi cell of L^*, the effect of position error is also constrained to this area.

The other two input error types, beep measurement error and clock synchronization error, are collectively referred to as *time error*. Time error affects t_{beep} and the pairwise comparisons of t_{beep} in Phases II and III. Time error also affects the TDOA estimates in Phase IV. Therefore, time error can have more complex effects than microphone position error.

Specifically, time error affects Phase II if it causes incorrect results in pairwise t_{beep} comparisons, making the wrong node appear to be closer to ▲. This possibility is examined in detail using the following three cases:

1) Time error at L^*, the node closest to ▲. As shown in Fig. 6a, if L^* reports a small t_{beep}, then L^* is still identified correctly in Phase II. But if L^* reports a t_{beep}
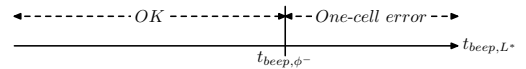
(a) Time error at L^*. $t_{beep,\phi-}$ is the smallest beep timestamp of all $\phi \in \Phi^*$. If $t_{beep,L^*} \leq t_{beep,\phi-}$, then the effect of error is mitigated. If $t_{beep,L^*} > t_{beep,\phi-}$, a one-cell error is produced.
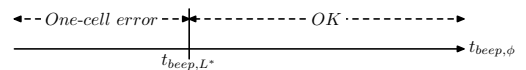
(b) Time error at any $\phi \in \Phi^*$. If $t_{beep,\phi} > t_{beep,L^*}$, then the effects are minimal. If $t_{beep,\phi} \leq t_{beep,L^*}$, a one-cell error is produced, similar to (a).

Fig. 6. Analysis of time error.

larger than any of its LVNs, L^* will pass leadership to that LVN, and Phase II will falsely identify that node as L^*, a one-cell error. Any localization instance for this beep will encounter the same problem and produce the same result.

2) Time error at any $\phi \in \Phi^*$, the set of L^*'s LVNs. As shown in Fig. 6b, this is essentially the dual of the previous case. If ϕ reports a large t_{beep}, then leadership will still be passed to and kept by L^* in Phase II, yielding correct results. But if ϕ reports a t_{beep} smaller than L^*'s, then ϕ will maintain leadership. Phase II will identify ϕ as the node closest to ▲, resulting in a one-cell error. Any localization instance for this beep will encounter the same problem and produce the same result.

3) Time error at a node A at least two cells away from ▲. Excessive time error at A can cause either A or one of its LVNs, depending on the sign of the error, to appear to be the node closest to ▲. Such a node is called a *false local minimum*. A false local minimum executes Phases III and IV, producing an inaccurate final location estimate. However, due to the constrained nature of Phase IV, this estimate will have a high associated MSE. A localization instance that does not involve A in Phase II can still correctly identify L^*, and in most cases, L^*'s estimate will have a much lower MSE than A's estimate. Therefore, the false local minimum can be detected and ignored.

In summary, position error and reasonable time error cause a minimal amount of output error. Excessive time error can cause a one-cell output error, which can be managed by the user. Thus, DiVA's design mitigates the effects of input error, as has been observed in both simulation and experiments.

4) Obstacles: DiVA is also robust to physical obstacles and various acoustic effects. An obstacle may cause reflection or scattering of the acoustic waves, but these effects do not interfere with DiVA's operation because the reflected and scattered signals arrive after the original signal, and generally with lower amplitude, so a node can distinguish the original signal and still accurately timestamp the beep. An obstacle may also cause acoustic diffraction, in which the sound waves propagate in a polygonal line. If a node only hears the diffracted beep, it may appear farther away from ▲ than it actually is. This can be thought of as a special case of measurement error, for which the previous discussion on time error applies. Finally, if a node misses the beep due to an obstacle, the neighboring nodes act as though the node has failed, and the discussion on node failure applies.

B. Localization of a Series of Beeps

In order for DiVA to localize the source of a series of beeps, collaborating nodes must be able to identify and communicate which specific beep is being localized. To aid in this, a *beep search window* is defined. A beep search window is the time interval in which P searches for beep b corresponding to $t_{beep,L}(b)$ received from L in a probe. A beep search window is defined by its center point and its width. To find the center point $t_{win}(b)$, P simply corrects $t_{beep,L}(b)$ for $\Delta t_{L,P}$, the clock offset between L and P, as follows:

$$t_{win}(b) = t_{beep,L}(b) - \Delta t_{L,P}. \tag{4}$$

Since time error can cause the beep timestamp to vary between nodes, the window width should be as large as possible to account for as much error as possible. If the beeps occur no more frequently than a known interval I_{min}, this value can be used as the window width. P then searches for $t_{beep,P}(b)$ within $I_{min}/2$ of either side of $t_{win}(b)$.

This method of determining the beep search window assumes that the difference in beep timestamps between L and P caused by the acoustic propagation delay is less than $I_{min}/2$. This assumption imposes a new requirement for how a node's LVNs are determined: any LVN P of L must be no more than a maximum distance $|\overline{LP}|_{max}$ from L, calculated as follows:

$$|\overline{LP}|_{max} = \frac{I_{min} v_s}{2}. \tag{5}$$

If $|\overline{LP}|_{max}$ is greater than the communication range of L, then this LVN requirement has no effect. But if I_{min} is very small, this requirement could be restrictive. Therefore, when designing a deployment to localize a particular source of a series of beeps, Eq. (5) should be used to verify that the deployment is dense enough for the given I_{min}.

V. SIMULATION STUDY

MATLAB simulations were used to evaluate the effects of input error, sensor density, communication range, and deployment irregularities on DiVA's performance. The simulations compared DiVA to DSL [22], [23] implemented with a dynamically-sized voting grid area and an 8×8 resolution. The simulations also implemented simple TOA and TDOA to provide a baseline comparison. Finally, a version of DiVA without Phase IV, called *DiVA Basic*, was also implemented. DiVA Basic uses the center of mass of the subregion identified in Phase III as the final location estimate.

Except those noted otherwise, each simulation randomly distributed 1000 microphone nodes in a 100×100 m area. Each data point is the average of results from 1000 different random acoustic source locations. The geographical distance between the actual acoustic source location and the estimate given by a localization method is referred to as the *output error*. Input error was simulated by randomly choosing a value between zero and the specified maximum amount, with position error taking a random geographic direction and beep measurement error randomly being either positive or negative. The average distance from a node to its nearest neighbor, d, is used to normalize both input and output error. Using a well-known formula, d is calculated as $d = (2\sqrt{1000/(100 \times 100)})^{-1} \approx 1.58$ m. The speed of sound, v_s, was taken to be 340 m/s. The clock synchronization error for all schemes was taken to be 44 μs [27].

A. Effect of Input Error

The four schemes were compared in terms of output error versus the amount of position error and the amount of beep measurement error, with a communication range of $6d$. The results for the two types of input error are nearly identical. The averaged results for position error, with zero beep measurement error, are shown in Fig. 7a, and the averaged results for beep measurement error, with zero position error, are shown in Fig. 7c. Since the results of TOA and TDOA depend on which nodes' measurements are used, the average 40th percentile (denoted as $K = 0.4$) and 80th percentile ($K = 0.8$) are shown in these figures.

As expected, DiVA performs the best, matching TOA's accuracy at low levels of input error and decreasing slower than either TOA or TDOA as input error increases. DiVA Basic also performs well, converging with DiVA at higher levels of input error, when the range-based estimates are not as accurate. DSL performs worse than the other schemes at low input error, since it is a range-free method, though it is consistently outperformed by the range-free DiVA Basic.

A CDF of the output error when the position error is $0.2d$ is shown in Fig. 7b, and the corresponding CDF for measurement error is shown in Fig. 7d. Nearly all of DiVA's output errors are less than d. DiVA Basic continues to show the strength of DiVA's range-free components, performing better than DSL. Also, DiVA Basic intersects with TOA at the 90th percentile, validating DiVA's hybrid approach. Range-based methods work well at low levels of input error, and range-free methods work well at high levels of input error, but the hybrid method performs the best at all levels of input error.
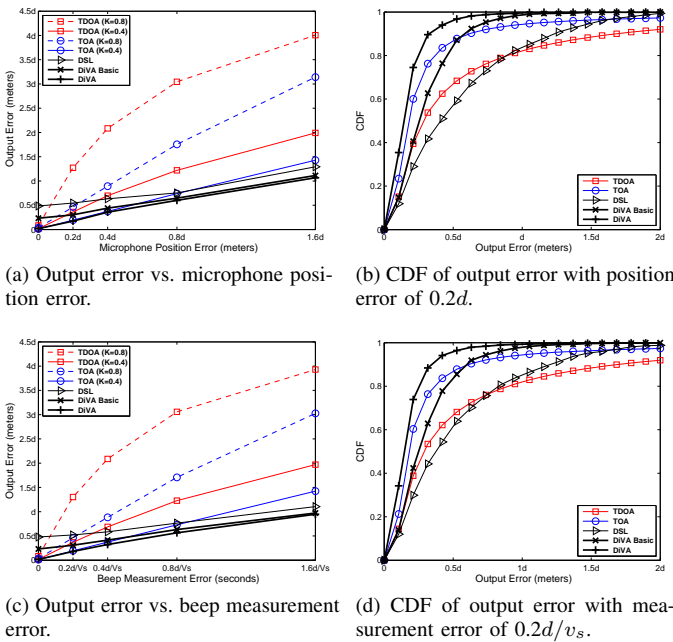

(a) Output error vs. microphone position error.

(b) CDF of output error with position error of $0.2d$.

(c) Output error vs. beep measurement error.

(d) CDF of output error with measurement error of $0.2d/v_s$.

Fig. 7. The effects of input error, with $d \approx 1.58$ m and $d/v_s \approx 4.65$ ms.

B. Effect of Sensor Density

Simulations were conducted to find the relationship between sensor density and output error for the distributed schemes. These simulations randomly distributed n microphone nodes in a 100×100 m area, with $n = 500$, 1000, and 2000. For each case, $d = (2\sqrt{n/(100 \times 100)})^{-1}$. Fig. 8 shows both normalized results, when the amount of input error varies with d, and absolute results, when the amount of input error stays constant.

From Fig. 8a, if input error changes proportionally with d, normalized output error remains essentially the same. In other words, the output error scales with d. But when the input error remains constant, as in Fig. 8b, the output error increases as the number of sensors decreases. This trend is intuitive: a denser network yields better results because the nodes are closer together. However, the slope of DiVA's output error in Fig. 8b is very small, especially compared to DSL and DiVA Basic. This shows that DiVA is highly scalable and can perform well in much sparser networks than the other solutions, leading to lower cost.
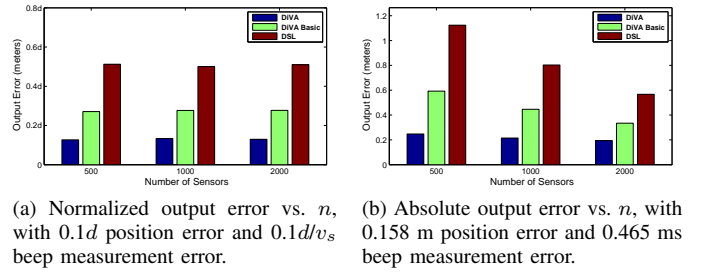

(a) Normalized output error vs. n, with $0.1d$ position error and $0.1d/v_s$ beep measurement error.

(b) Absolute output error vs. n, with 0.158 m position error and 0.465 ms beep measurement error.

Fig. 8. The effects of sensor density.

C. Effect of Communication Range

Simulations were used to test the effect of communication range, and therefore average node degree, on DiVA and DSL with $0.1d$ position error and $0.1d/v_s$ measurement error. Fig. 9a shows the resulting output error and the number of nodes involved in the final localization step, including the closest node L^*, the LVNs of L^* for DiVA (and DiVA Basic), and the nodes within a one-hop range of L^* for DSL. DiVA performs better than DSL, but all schemes suffer in accuracy as the communication range decreases. DiVA reaches a saturation point in terms of output error with a communication range of around $5d$. After this point, the number of nodes involved for DiVA stays constant, due to its use of LVNs, suggesting that DiVA is highly scalable. However, the number of nodes involved for DSL continues to grow. In short, DiVA achieves better accuracy while utilizing fewer nodes.

D. Effect of Deployment Irregularities

The effect of deployment irregularities was tested by removing nodes from the deployment to create circular voids with a 10 m radius. The output error versus the number of voids, with a communication range of $6d$ and the same input error as before, is shown in Fig. 9b. The value of d
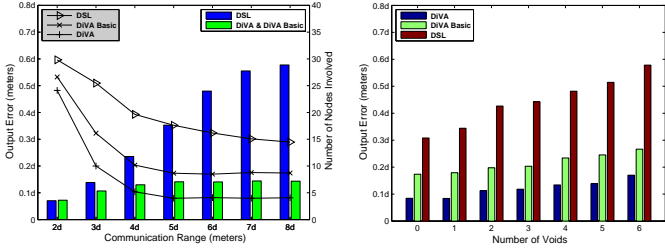
(a) Effects of microphone node communication range.

(b) Output error vs. number of voids in the deployment.

Fig. 9. The effects of communication range and deployment irregularities for $0.1d$ position error plus $0.1d/v_s$ beep measurement error, with $d \approx 1.58$ m.

remains constant across the number of voids for consistency. Both DiVA and DiVA Basic are more accurate than DSL, and both also show a slower growth rate in output error as the number of voids increases. These results support the conclusion that DiVA remains functional in the presence of deployment irregularities.

VI. EXPERIMENTAL EVALUATION

DiVA was implemented on custom wireless microphone nodes as shown in Fig. 10a. The nodes were based on the Texas Instruments CC2530 chip, which integrates an IEEE 802.15.4 wireless radio with an 8051 microcontroller and 8 kB of RAM. The software modules of DiVA are shown in Fig. 10b and were implemented in the C language. The audio detection module also used an audio chip that generates an interrupt in the microcontroller when the amplitude of the microphone signal reaches a threshold, set by adjusting the gain on the audio chip. This was sufficient for proof-of-concept testing, and both the hardware and software of the audio detection module could be modified or replaced to meet different application requirements.

(a) Wireless microphone node.
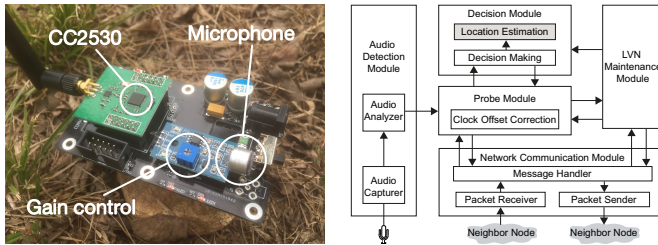
(b) DiVA software modules.

Fig. 10. DiVA implementation platform.

Two experiments are reported here: a laboratory experiment in an open indoor room, and a more challenging experiment in an outdoor area. A cap gun was used as the acoustic source. Any measurement or position error in the system occurred naturally. With simple benchmarking experiments, discrepancies in beep timestamps due to time error were estimated to be within the range of ± 200 μs.

A. Laboratory Experiment

The purpose of the laboratory experiment was to test the performance of DiVA in an ideal environment. The deployment consisted of 32 microphone nodes in an 18×12 m area of an indoor gymnasium, yielding $d \approx 1.30$ m. There were no obstacles or voids in the deployment. Positions for the nodes and acoustic sources were determined using measuring tapes. Beeps were generated at 187 locations in the area, and the results are shown in Fig. 11a. Fig. 11b shows a CDF of the error in terms of d.
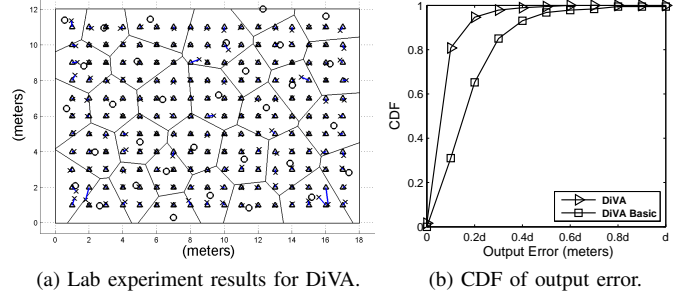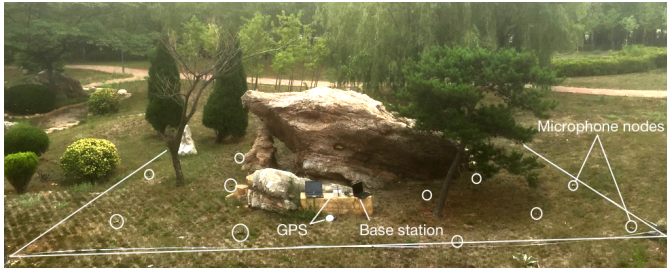

(a) Lab experiment results for DiVA.

(b) CDF of output error.

Fig. 11. Results of the lab experiment, with $d \approx 1.30$ m. In (a), node positions are marked as $\circ$, acoustic source locations are marked as $\triangle$, and estimates are marked as $\times$. A blue line connects each estimate to the corresponding $\triangle$.

As expected, the estimates are distributed around, but generally close to, the actual locations. The larger errors are found near the edge of the deployment, such as in the lower right and left corners. The CDF shows a similar shape and better results than the simulation CDFs shown in Figs. 7b and 7d, with an error for DiVA of $0.1d$ or less for around 80% of all estimates. This suggests that the lab environment generated little input error, as desired. The conclusions from the laboratory experiment are that DiVA performs well in ideal conditions, and that the implementation of DiVA is sound.
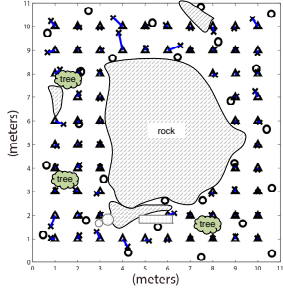
B. Outdoor Experiment

To test DiVA's performance in a more challenging environment, an experiment was performed in an 11×11 m outdoor area that features trees, vegetation, and large rocks, as shown in Fig. 12a and marked in Fig. 12b. Positions for the nodes were determined using a high-accuracy ComNav Technology GPS system, and positions for the acoustic sources were determined using measuring tapes, introducing the possibility of position error between the two methods. The obstacles in the area increased the probability of beep measurement error. Uncontrolled background noises, such as a nearby road and the buzz of cicadas, were also present. The deployment consisted of 26 microphone nodes, and a total of 68 acoustic source positions were each tested twice, with the averaged results shown in Fig. 12b. A CDF is shown in Fig. 12c.

Overall, DiVA performs well. Less accurate results are found in the areas with few LVNs, such as the narrow area above the large rock in Fig. 12b. Still, the results suggest that DiVA can work well in challenging environments, making it suitable for real-world applications. To further demonstrate this, a field trial in a larger area, with a specific monitoring goal, is considered for future work.

(a) Outdoor experiment area.

(b) Outdoor experiment results.
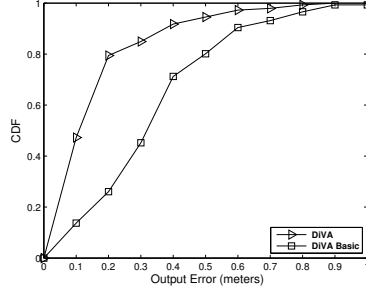
(c) CDF of output error.

Fig. 12. Outdoor experiment. In (b), node positions are marked as ○, acoustic source locations are marked as △, and estimates are marked as ×. A blue line connects each estimate to the corresponding source location.

VII. CONCLUSIONS AND FUTURE WORK

This paper presents DiVA, a new distributed acoustic source localization algorithm for wireless sensor networks that is robust and suitable for use in real-world environments. DiVA uses range-free pairwise comparisons of the sound detection timestamps of neighboring microphone nodes to traverse the Voronoi diagram and find the node closest to the acoustic source, which then estimates the source's location using a constrained range-based method. Simulation and experimental results demonstrate the accuracy and robustness of DiVA's hybrid range-free and range-based approach when facing practical challenges such as position and time error, obstacles, and deployment irregularities. DiVA was also shown to be scalable to different sensor densities and communication ranges. Future work includes a large-scale field trial and extensions for handling multiple sources and underwater scenarios.

ACKNOWLEDGEMENT

REFERENCES

[1] J. Sallai, Á. Lédeczi, and P. Völgyesi, "Acoustic shooter localization with a minimal number of single-channel wireless sensor nodes," in *ACM SenSys*, 2011.

[2] Á. Lédeczi, A. Nádas, P. Völgyesi, G. Balogh, B. Kusy, J. Sallai, G. Pap, S. Dóra, K. Molnár, M. Maróti *et al.*, "Countersniper system for urban warfare," *ACM Transactions on Sensor Networks*, vol. 1, no. 2, pp. 153–177, 2005.

[3] K. Na, Y. Kim, and H. Cha, "Acoustic sensor network-based parking lot surveillance system," in *EWSN*, 2009.

[4] B. Wei, M. Yang, Y. Shen, R. Rana, C. T. Chou, and W. Hu, "Real-time classification via sparse representation in acoustic sensor networks," in *ACM SenSys*, 2013.

[5] M. Allen, L. Girod, R. Newton, S. Madden, D. T. Blumstein, and D. Estrin, "Voxnet: An interactive, rapidly-deployable acoustic monitoring platform," in *IEEE IPSN*, 2008.

[6] E. Xu, Z. Ding, and S. Dasgupta, "Source localization in wireless sensor networks from signal time-of-arrival measurements," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2887–2897, 2011.

[7] I. Enosh and A. J. Weiss, "Outlier identification for TOA-based source localization in the presence of noise," *Elsevier Journal on Signal Processing*, vol. 102, pp. 85–95, 2014.

[8] S. Venkateswaran and U. Madhow, "Localizing multiple events using times of arrival: A parallelized, hierarchical approach to the association problem," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5464–5477, 2012.

[9] T. Ajdler, I. Kozintsev, R. Lienhart, and M. Vetterli, "Acoustic source localization in distributed sensor networks," in *The Asilomar Conference on Signals, Systems, and Computers*, 2004.

[10] J. Zhang, T. Yan, J. A. Stankovi, and S. H. Son, "Thunder: towards practical, zero cost acoustic localization for outdoor wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 15–28, 2007.

[11] K. Yang, G. Wang, and Z.-Q. Luo, "Efficient convex relaxation methods for robust target localization by a sensor network using time differences of arrivals," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2775–2784, 2009.

[12] H. Liu and E. Milios, "Acoustic positioning using multiple microphone arrays," *The Journal of the Acoustical Society of America*, vol. 117, no. 5, pp. 2772–2782, 2005.

[13] A. N. Bishop, B. D. Anderson, B. Fidan, P. N. Pathirana, and G. Mao, "Bearing-only localization using geometrically constrained optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 1, pp. 308–320, 2009.

[14] D. Li and Y. H. Hu, "Energy-based collaborative source localization using acoustic microsensor array," *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 321–337, 2003.

[15] M. G. Rabbat and R. D. Nowak, "Decentralized source localization and tracking," in *IEEE ICASSP*, 2004.

[16] D. Blatt and A. O. Hero, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3614–3619, 2006.

[17] W.-P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 258–271, 2004.

[18] Y. Liu, Y. H. Hu, and Q. Pan, "Distributed, robust acoustic source localization in a wireless sensor network," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4350–4359, 2012.

[19] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *ACM MobiCom*, 2003.

[20] Y. Shang, W. Ruml, Y. Zhang, and M. P. Fromherz, "Localization from mere connectivity," in *ACM MobiHoc*, 2003.

[21] K. Vu and R. Zheng, "Geometric algorithms for target localization and tracking under location uncertainties in wireless sensor networks," in *IEEE INFOCOM*, 2012.

[22] Y. Kim, J. Ahn, and H. Cha, "Locating acoustic events based on large-scale sensor networks," *Sensors*, vol. 9, no. 12, pp. 9925–9944, 2009.

[23] J. Ahn and H. Cha, "A sensor network-based multiple acoustic source localization algorithm," in *ACM Symposium on Applied Computing*, 2012.

[24] Q. Wang, R. Zheng, A. Tirumala, X. Liu, and L. Sha, "Lightning: a hard real-time, fast, and lightweight low-end wireless sensor election protocol for acoustic event localization," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 570–584, 2008.

[25] F. Aurenhammer, "Voronoi diagrams–a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

[26] B. A. Bash and P. J. Desnoyers, "Exact distributed voronoi cell computation in sensor networks," in *IEEE IPSN*, 2007.

[27] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *ACM SenSys*, 2003.

[28] F. E. Grubbs, "Sample criteria for testing outlying observations," *The Annals of Mathematical Statistics*, pp. 27–58, 1950.